Blistering Fast SQL Access to Hadoop using IBM BigInsights 3.0 with Big SQL 3.0

SQL-over-Hadoop implementations are ready to execute OLAP and complex query workloads at a fraction of the cost of traditional RDBMSs -- if you pick the right technology. Get beyond the hype and explore how IBM's BigInsights 3.0 with Big SQL 3.0, Cloudera's CDH 5.1.2 with Impala 1.4.1 and Hortonworks HDP 2.1 with Hive 0.13 executing on Tez stack up when running the Hadoop Decision Support benchmark. This benchmark typifies the kind of work associated with augmenting a relational data warehouse with a Hadoop-based platform and was modeled on the Transaction Processing Performance Council Decision Support (TPC-DS) benchmark and audited by a TPC-DS certified auditor.

Key Findings

- Query support matters. All 99 TPC-DS queries could be expressed in Big SQL without the need to resort to extensive rewrites. Indeed, it took our SQL professionals less than one hour to port the whole workload to Big SQL. Due to limited SQL support, we were only able to express 88 of the 99 queries in Impala. For Hive, only 90 of the 99 queries could be successfully validated. The ports to both Hive and Impala took several weeks and SQL limitations forced us to extensively re-write many queries. See the "Query generation, re-write and validation" section for details.
- <u>Scale matters.</u> Big SQL was the only implementation capable of executing all 99 queries at 10TB in both single-user and multi-user runs. By contrast, Hive executed only 70% of the workload at 10TB and Impala executed only 73%. Big SQL was also the only one able to execute the full workload at 30TB. See "Query Execution" section for details.
- <u>Throughput matters.</u> When the workload was restricted to the 46 queries readily supported by all 3 implementations, *Big SQL achieved 3.6 times more throughput than Impala and 5.4 times more throughput than Hive*.

Recommendations

Information management leaders should:

- Know your needs. Consider the types of queries you might want to run initially, as well as those you may need to run in the future. How complex are your queries? What language constructs do you expect to use? How skilled is your staff in query rewrite technology?
- Look beyond the hype. When vendors proffer performance claims, ask the tough questions: What workloads did they use? To what degree do these workloads comply with industry standards? What hardware configuration was used? Who audited the benchmark?
- **Kick the tires.** If SQL support and/or performance are critical, there's no substitute for performing your own in-house comparative tests using your own workload. But if you don't have the skills, time, or resources to do so, examining comparative, audited benchmark results can be a good alternative.

Contents

Blistering Fast SQL Access to Hadoop using	1
IBM BigInsights 3.0 with Big SQL 3.0	1
Contents	2
Preface	4
Introduction	6
Methodology	6
Cluster configuration & tuning	8
Database schema	9
Query generation, re-write and validation	9
Data load	11
Query execution	11
10TB scale factor using a common query set	12
10TB scale factor using all 99 queries	17
30TB scale factor using all 99 queries	19
Summary	21
Appendix A: Workload times:	24
Appendix B: Cluster topology and hardware configuration	25
Appendix C: Tuning	26
C.1 Big SQL Configuration	26
C.2: Impala Configuration	29
C.3: Hive Configuration	40
C.4 OS Storage Configuration:	42
Appendix D: Database Schema	45
D.1 Big SQL	45
D.2 Impala	52
D.3 Hive 0.13	59

A	Appendix E: Query Text	65
	E.1 Big SQL Queries:	65
	E.2 Impala Queries:	86
	E.3 Hive 0.13 Queries:	106
ļ	Appendix F: Load & Analayze Scripts:	126
	F.1 Big SQL Load & Analyze scripts:	126
	F.2 Impala Load & Analyze scripts:	146
	F.3 Hive0.13 Load & Analyze scripts:	149
Å	Appendix G: Attestation Letter:	155

Preface

The information contained in this document is distributed on an AS IS basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is the customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

Performance data contained in this document were determined in various controlled laboratory environments and are for reference purposes only. Customers should not adapt these performance numbers to their own environments and are for reference purposes only. Customers should not adapt these performance numbers to their own environments as system performance standards. The results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used.

This publication was produced in the United States. IBM may not offer the products, services, or features discussed in this document in other countries, and the information is subject to change without notice. Consult your local IBM representative for information on products and services available in your area.

© Copyright International Business Machines Corporation 2014 All rights reserved.

Permission is hereby granted to reproduce this document in whole or in part, provided the copyright notice as printed above is set forth in full text on the title page of each item reproduced.

U.S. Government Users - Documentation related to restricted rights: Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Trademarks

IBM, the IBM logo, System x and System Storage are trademarks or registered trademarks of International Business Machines Corporation.

The following terms used in this publication are trademarks of other companies as follows: TPC Benchmark and TPC-DS are trademarks of Transaction Processing Performance Council; Intel and Xeon are trademarks or registered trademarks of Intel Corporation. Other company, product, or service names, which may be denoted by two asterisks (**), may be trademarks or service marks of others.

Cloudera, the Cloudera logo, Cloudera Impala are trademarks of Cloudera.

Hortonworks, the Hortonworks logo and other Hortonworks trademarks are trademarks of Hortonworks Inc. in the United States and other countries.

Apache, Apache Hadoop, Hadoop, Apache Hive, Hive, and Apache Tez are either registered trademarks or trademarks of the **Apache Software Foundation** in the United States and/or other countries.

Notes

- ¹ GHz and MHz only measures microprocessor internal clock speed, not application performance. Many factors affect application performance.
- ² When referring to hard disk capacity, GB, or gigabyte, means one thousand million bytes. Total user-accessible capacity may be less.

About the authors

Simon Harris: Simon is the Big SQL performance lead working in the IBM BigInsights development team. He has 20 years of experience working in information management including MPP RDBMS, federated database technology, tooling and big data. Simon now specializes in SQL over Hadoop technologies.

Abhayan Sundararajan: Abhayan is a Performance Analyst on IBM BigInsights with a focus on Big SQL. He has also held a variety of roles within the IBM DB2 team, including functional verification test and a brief foray into development before joining the performance team to work on DB2 BLU.

Edward G. (Gus) Branish: Gus is a Client Technical Professional on the Information and Analytics Group Team in the IBM Competitive Project Office. Gus has 20 years of experience on performance benchmarks of information management software. He has a special interest in query optimization in parallel processing environments.

Kenneth Chen: Ken is an Executive IT Specialist and Consulting Software Product Design Professional from the IBM Big SQL/DB2 Technical Alliance organization at the IBM Toronto Lab, specializing in business partner application enablement - application design, development, and performance. His areas of focus included distributed database platforms, as well as IBM DB2 Content Manager.

Special thanks to the following people for their contribution to the benchmark and content:

Berni Schiefer – Distinguish Engineer, Information Management Performance and Benchmarks, DB2 LUW, Big Data, MDM, Optim Data Studio Performance Tools; Adriana Zubiri – Program Director, Big Data Development; John Poelman – BigInsights Performance; Mike Ahern – BigInsights Performance; Mi Shum – Senior Performance Manager, Big Data; Cindy Saracco - Solution Architect, IM technologies - Big Data; Avrilia Floratou – IBM Research; Fatma Özcan – IBM Research; Glen Sheffield – Big Data Competitive Analyst; Gord Sissons – BigInsights Product Marketing Manager; Matthew Emmerton - DB2 OLTP Performance and Solutions Development; Stewart Tate – Senior Technical Staff Member, Information Management Performance Benchmarks and Solutions; Jo A Ramos - Executive Solutions Architect - Big Data and Analytics.

Introduction

Performance benchmarks are an integral part of software and systems development, as they can evaluate systems performance in an objective way. They have also become highly visible components of the exciting world of marketing SQL over Hadoop solutions.

Good benchmarks reflect, in a practical way, an abstraction of the essential elements of real customer workloads. In a typical Hadoop system this includes the task of loading the files onto HDFS as well as having both individual and concurrent users submitting SQL queries to the system.

IBM has constructed and used the Hadoop Decision Support (Hadoop-DS) benchmark, which was modelled on the industry standard TPC-DS¹ benchmark and validated by a TPC certified auditor. This benchmark typifies the kind of work associated with augmenting an existing relational data warehouse with a Hadoop-based platform. While adapting the workload for the nature of a Hadoop system we worked to ensure the essential attributes of both typical customer requirements and the benchmark were maintained.

In order to calibrate the Big SQL results we identified the common working subset of the full TPC-DS query set that could be measured and compared across three SQL over Hadoop implementations – IBM's BigInsights 3.0 with Big SQL 3.0 (hereafter referred to as Big SQL), Cloudera's CDH 5.1.2 with Impala 1.4.1 (hereafter referred to as Impala) and Hortonworks HDP 2.1 with Hive 0.13 executing on Tez (hereafter referred to as Hive 0.13) – these were the latest versions of the products available in September 2014. In a number of cases sophisticated SQL construction skills were required in order to translate the official reference SQL into a syntax executable on the target systems. The resulting subset of queries reflects a broad range of SQL functionality and query complexity. While we were surprised at just how difficult it was to get both Hive and Impala to run many of the queries, particularly in the multi-user concurrency test, we are proud to have clarified the validated TPC-DS query portfolio known to work on both Impala and Hive.

In this paper we will provide an overview of our methodology, details on query generation, selection and validation. We will also describe the system under test from a hardware and software perspective, and detail the process for creating the tables, populating the Hadoop files and gathering statistics. We also summarize and analyze the results of the performance measurements.

Methodology

The aim of this benchmark was to provide an accurate and fair comparison across all three SQL over Hadoop distributions in a scenario common to many organizations adopting the technology today. The most common scenario we see involves off-loading subsets of workloads from the traditional relational data warehouse to SQL over Hadoop solutions (a process commonly referred to as warehouse augmentation). For this reason our Hadoop-DS workload was modelled on the Transaction Processing Performance Council Benchmark DS (TPC-DS)². The TPC-DS benchmark uses relational database management systems (RDBMSs) to model a decision support system that examines large volumes of data and gives answers to real-world business questions by executing queries of various types (such as ad-hoc, reporting, OLAP and data mining type queries). It is,

¹ TPC Benchmark and TPC-DS are trademarks of the Transaction Processing Performance Council (TPC).

² The Hadoop-DS benchmark is based on the latest revision of the TPC-DS specification. The specification can be found at http://www.tpc.org/tpcds/default.asp

therefore, an ideal fit to mimic the experience of an organization porting parts of their workload from a traditional warehouse housed on an RDBMS to a SQL over Hadoop technology. As highlighted in IBM's "Benchmarking SQL-on-Hadoop Systems: TPC or not TPC?" Research paper, SQL over Hadoop solutions are in the "wild west" of benchmarking. Some vendors may use the data generators and queries of existing TPC benchmarks, but cherry pick the parts of the benchmark most likely to highlight their own strengths and fail to adhere to the specified rules – thus making comparison between results meaningless.

To make this a complete and fair comparison, we did not cherry-pick the parts of the TPC-DS workload that would highlight Big SQL's strengths. Instead, we included **all** parts of the TPC-DS workload that are appropriate for SQL over Hadoop solutions: data loading, single user performance and multi-user performance. Since TPC-DS is a benchmark designed for relational database engines, some aspects of the benchmark are not applicable to SQL over Hadoop solutions. Broadly speaking, those are the "Data Maintenance" and "Data Persistence" sections of the benchmark. Consequently these sections were omitted from our Hadoop-DS workload. The TPC-DS benchmark also defines restrictions related to real-life situations – such as preventing the vendor from changing the queries to include additional predicates based on a customized partitioning schema, employing query specific tuning mechanisms(such as optimizer hints), making configuration changes between the single and multi-user, tests etc.

To ensure the fairest possible comparison, we endeavoured to stay within the bounds of these restrictions for the Hadoop-DS workload and conducted the comparison with candor and due diligence. To validate our candor, we retained the services of Infosizing⁴, an established and respected benchmark auditing firm with multiple TPC certified auditors, including one with TPC-DS certification, to review and audit all our benchmarking results. It is important to note that these are not official TPC-DS benchmark results since aspects of the standard benchmark that do not apply to SQL over Hadoop solutions were not implemented. However, the independent review of the environment and results by a certified auditor shows IBM commitment to openness and fair play in this arena. All deviations from the TPC-DS standard benchmark, for all products, are noted in the attached auditor's attestation letter in Appendix G. In addition, all the information required to reproduce the environment and the Hadoop-DS workload for all three vendors is published in the various Appendices of this document – thus allowing any vendor or third party the ability to execute the benchmark and verify the results independently.

There are 99 queries in the official TPC-DS benchmark along with a toolkit to generate the queries in various SQL dialects. Since there are many variations of SQL dialects, the specification also allows the sponsor to make pre-defined minor modifications to the queries so they can be successfully compiled and executed. The team attempted to port all 99 queries to all 3 platforms. For us, close to

³ "Benchmarking SQL-on-Hadoop Systems: TPC or not TPC?" http://researcher.ibm.com/researcher/files/us-aflorat/BenchmarkingSQL-on-Hadoop.pdf

⁴ Infosizing: www.infosizing.com/

a third of the queries ported to Impala needed extensive re-writes⁵, while for Hive, close to a quarter of the queries required extensive re-writes. These extensive re-writes were necessary to avoid limitations in the SQL support for these vendors. Some were considered re-writes that are commonly employed by SQL programmers, but others were much more involved. The team assigned one SQL programmer to port the queries to Hive, one to port the queries to Impala and one to port the queries to Big SQL – each programmer had many years of SQL experience. As a consequence of the limited SQL dialect available in Hive and Impala, it took approximately four weeks of effort to port the queries to these products. Since Big SQL supports a rich set of SQL syntax, many queries worked *out of the box* and consequently the porting effort was less than 1 hour.

Although SQL dialect support was not the focus of this project, the teams experiences porting the queries show how important rich SQL support is for an SQL over Hadoop solution. This is even more important in an enterprise when the queries are being automatically generated by Business Intelligence tools such as Microstrategy or Cognos which restrict the ability to manually re-write the queries they generate. Parallels can certainly be drawn between the team's experience of porting the queries from TPC-DS (an RDBMS workload) to Hadoop-DS (an SQL over Hadoop workload) and organizations attempting to move their own workloads away from their data warehouses to SQL over Hadoop. Unless an organization wants to spend considerable time and effort porting their queries, they should seriously evaluate the SQL capabilities and compliance of SQL over Hadoop vendors.

Several queries that had been successfully ported to Hive and Impala and ran well in some circumstances, failed when executed on larger data sets or with multiple concurrent streams . The most common error for both vendors was running out of memory. The team spent several more weeks analysing the failed Hive and Impala queries and tuning the environment to get as many queries working as possible. All queries were successful in Big SQL without the need for additional tuning.

To provide the fairest apples to apples comparison the Hadoop-DS benchmark needed to use the same set of queries across all vendors – therefore the team executed a number of *query validation runs* to check which queries could successfully execute both the single and multi-user runs at the chosen scale factor. Any query which failed on either Hive or Impala was removed from the set of queries to be tested in the final performance run. In this way, we were left with a common sub-set of queries that all vendors could execute at the target scale factor. This query set was used to execute both the single-user and multi-user performance tests.

Cluster configuration & tuning

The benchmark was conducted on three identical 17 node clusters, each node being an IBM x3650 M4 BD server. A complete specification of the hardware used can be found in

⁵ For Impala, we followed the guidelines in "Porting SQL from Other Database Systems to Impala" in the "Impala SQL Language Reference" http://www.cloudera.com/content/cloudera/en/documentation/cloudera-impala/v1/latest/Installing-and-Using-Impala/ciiu_porting.html?scroll=porting_statements_unique_1

Appendix B: Cluster topology and hardware configuration.

In our performance evaluation, a number of tuning best practices were adopted across all products. As with all benchmarks, the tuning process was iterative and continually refined across all three distributions as queries and workloads were executed. The clusters all started with the same set of OS and kernel tunings common to Hadoop clusters. The Hive cluster was tuned further using the recommendations found on Hortonworks latest benchmark (at

http://hortonworks.com/blog/benchmarking-apache-hive-13-enterprise-hadoop/). The latest Impala benchmarks do not provide configurational information, so the team used the best practices from the Cloudera documentation. Big SQL was configured according to IBM's best practices. The detailed tuning and configuration used for each product can be found in Appendix C.

Tuning features that benefited performance of the overall workload were employed wherever reasonable. Of course, the team had a finite amount of time to complete the benchmark and it was not possible to evaluate all features and tuning knobs for all products. We focused on those features we thought would give the biggest payback in terms of performance, based on our understanding of each of the products.

Big SQL was configured to use unique and differentiating optimization features such as Informational Constraints and Statistical Views. Both these features provide additional information to the Big SQL optimizer allowing for better access strategy cost estimates and consequently, improved performance.

The TCP-DS specification recognizes the importance of constraints and statistics in the query optimization process. It is our belief, however, that some of the limitations imposed by the TPC-DS specification are less applicable to the read-only SQL over Hadoop environment than to the RDBMS environment for which they are targeted. Although these features may not meet all aspects of the TPC-DS specification, they are commonly used by customers in other IBM data management products, and therefore we strongly believe they provide value to our customers. Our statistics gathering commands, constraint definitions and statistical view definitions are disclosed in Appendix F.

In order to simulate a production system the Hadoop-DS benchmark was executed without individual query tuning and without configuration changes between the single and multi-user runs.

Database schema

The Hadoop-DS schema mirrors the TPC-DS schema and consists of 7 large fact tables and 17 smaller dimension tables. All products implemented the full 24 table schema.

Full schema definitions for each product can be found in Appendix D.

Query generation, re-write and validation

The team followed the documented process to generate the queries from the templates provided in the TPC-DS specification. Those queries which failed to compile or experienced run-time errors went

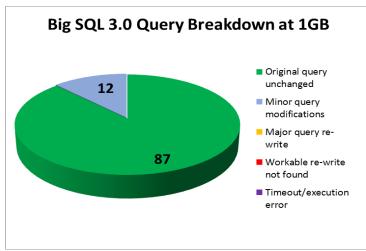
onto the second phase which was to apply a set of minor query modifications in the spirit of what is allowed by the TPC-DS specification. Queries which still would not compile, execute successfully or provided incorrect results went onto a third phase in which more extensive and complex query rewrites were attempted. All queries were validated using the 1GB qualification database against pre-

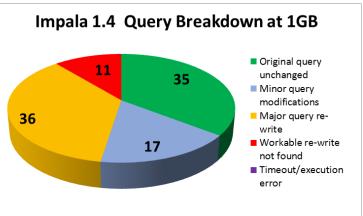
defined result sets provided by the TPC-DS specification – to help verify that the modified queries still answered the business questions being posed.

The team dedicated significant time and effort to get as many queries as possible working. Chart 1 summarizes the breakdown of minor query modifications and major rewrites against the 1GB qualification database for each yendor.

A product's ability to execute SQL queries generated directly from the unchanged templates, or slightly altered with only minor query modifications is an excellent indicator of its support of the SQL language.

Following a substantial porting effort a valid re-write could not be found for 11 Impala queries - Impala 1.4.1





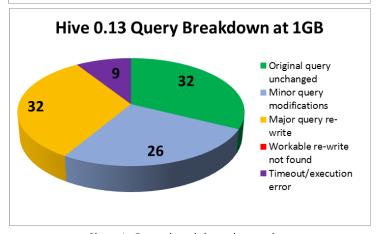


Chart 1: Query breakdown by vendor

does not support SQL windowing functions. All 99 queries were ported to Hive syntax, but 9 had execution errors or timed-out (after 2 hours) on the 1GB qualification database.

For Big SQL, 87 of the 99 queries worked directly from the original query. The other 12 queries required only simple and minor modifications (mainly type casts) and took less than an hour to complete.

One of the recommendations from this experience is that organizations considering SQL over Hadoop solutions should carefully evaluate the level of SQL support in order to avoid a similar painful experience.

In summary, after the porting exercise there were 83 queries working across all three products against the 1GB qualification database.

Data load

The team started with a database scaled at 10TB with a plan to scale it up to 30TB later. This decision was motivated by the fact that Cloudera had published a TPC-DS like benchmark at 15TB, and HortonWorks similarly at 30TB – both using a smaller query subset that accessed just a fraction of the full set of TPC-DS tables

The data load phase of the workload includes everything required to ingest the data into the database and prepare the cluster to execute the single and multi-user runs. Briefly, this phase consists of hardware and software set-up, data generation and copy to HDFS, cluster tuning, table creation, ingest of the data into the storage format of choice, gathering of statistics and creation of any data structures to aid performance and scalability.

For each product the optimal storage format was chosen - for Hive 0.13 this is ORC with ZLib compression enabled. For Impala and Big SQL, the Parquet format was used with Snappy compression.

T 1 4			1 . 1			•	4070	•	1 .
Table 1	Shows	the	database	INGEST	times	t∩r	101R	ot raw	uata.
IUDICI	3110443		aatabase	1115036	unics	101	1010	OI I G VV	aata.

	Big SQL 3.0	Impala 1.4.1	Hive 0.13
Storage format	Parquet	Parquet	ORC
Compression	Snappy	Snappy	ZLib
Data ingest time (mins)	185	107	663

Table 1: Database load times for Hadoop-DS @10TB.

Query execution

Hadoop-DS uses the "Hadoop-DS Qph" metric to report query performance. The Hadoop-DS Qph metric is the effective query throughput, measured as the number of queries executed over a period of time. A primary factor in the Hadoop-DS metric is the scale factor (SF) -- size of data set -- which is used to scale the actual performance numbers. This means that results have a metric scaled to the database size which helps differentiate large databases from small database (since performance is typically a factor of database size).

A Hadoop-DS Qph metric is calculated for each of the single and multi-user runs using the following formula:

Hadoop-DS Qph @ SF = ((SF/100) * Q * S) / T

Where:

- SF is the scale factor used in GB (10,000 in our benchmark).
 SF is divided by 100 in order to normalize the results using 100GB as the baseline.
- Q is the total number of queries successfully executed
- S is the number of streams (1 for the single user run)
- T is the duration of the run measured in hours (with a resolution up to one second)

Hadoop-DS Qph metrics are reported at a specific scale factor. For example 'Hadoop-DS Qph@10TB' represents the effective throughput of the SQL over Hadoop solution against a 10TB database.

10TB scale factor using a common query set

The same 83 queries that successfully executed on all 3 products on the 1GB qualification database were taken and executed on a 10TB database. Both Hive and Impala struggled to execute all 83 common queries in the initial single user validation runs at 10TB. The most common error received came from running out of memory, followed by queries timing out (after 2 hours). There is a well-known restriction in Impala 1.4.1 that the internal state of the query has to fit into the aggregate memory on the cluster. However, to experience similar errors using Hive 0.13 on Tez was a surprising find. Hive also experienced *Null pointer* exceptions when particular (valid) predicates were substituted into some queries. Several more days were spent tuning Hive and Impala in order to increase the number of queries successfully completing at 10TB. Meanwhile, Big SQL was able to complete all 99 queries at 10TB without additional tuning.

Initially on the 4-stream concurrent run, there were many more out of memory failures in both Hive and Impala. For Impala, it appears the "coordinator node" for each Impala query requires more memory and CPU cycles than the other nodes that process the query⁶. Using the default configuration (of a single Impala coordinator) therefore led to additional out of memory errors when executing multiple concurrent streams. The Impala driver scripts were therefore modified to issue queries using round-robin scheduling, so that each connection used a different coordinator node. This load-balancing technique lets the Impala nodes share the additional coordinator work, rather than concentrating it on a single machine. In addition it was also necessary to configure *Dynamic Resource Pools* within the Cloudera cluster to manage resource allocations and prevent queries from failing. Full details can be found in Appendix C.

For Hive, the YARN memory configuration settings are particularly critical. The number of containers was increased by reducing the container size, as each YARN task is assigned a container. The Java heap (particularly the *mapreduce.reduce.java.opts* property) was increased to address many of the *out of Heap* issues in the multi-stream run. These changes were implemented to override default configurations in the *mapred-site*, *yarn-site*, *tez-site*, and *hive-site* configuration files. It was also noted that at times of high CPU usage some jobs in the multi-stream run timed-out before executing in the Tez session. Consequently the Tez time out property was set to a very large value to prevent

⁶ http://www.cloudera.com/content/cloudera/en/documentation/cloudera-impala/v1/latest/Installing-and-Using-Impala/ciiu_concepts.html

jobs from timing-out and to allow the job to wait for the resource to be free to start execution. In order to provide the most available memory to the environment, 85% of total memory (of 128GB) was made available for configuration. In addition, HBase and Oozie were stopped as they were not required to run the benchmark. A number of other best practices across Hive and Tez were also adopted – see Appendix C for full details. It took the team a few weeks to resolve all these issues on Hive and Impala.

The benefits of Big SQL's rich SQL support and robust execution engine clearly shines through at this point and it is worth emphasizing the fact that Big SQL is the only product capable of executing all 99 queries at 10TB in both single and multi-stream validations runs. Chart 2 highlights the fact that Impala can only successfully execute 73% of the original workload, and Hive 70%.

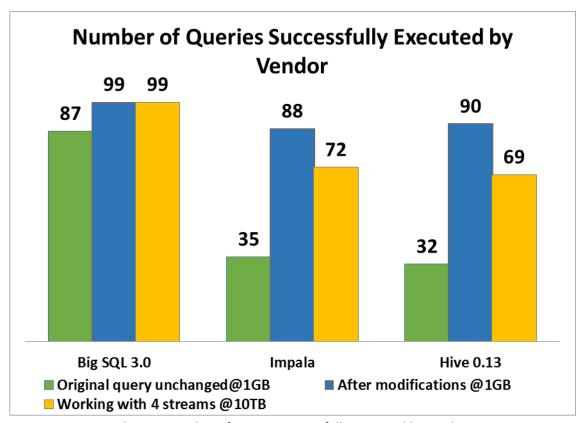


Chart 2: Number of queries successfully executed by product

Chart 2 also highlights how Impala and Hive struggle to scale. 18% of Impala queries that worked at 1GB failed to work at 10TB, and 23% of Hive queries that worked at 1GB failed to work at 10TB.

There were 46 common queries across the 72 Impala and 69 Hive queries working after the single and multi-user validation runs at 10TB. These were the queries used during the performance runs.

Having identified the common set of 46 working queries the team was now in a position to execute the performance run and thus provide a true apples to apples comparison across the three distributions.

In compliance with the TPC-DS specification, a single user performance run was executed, followed by a multi-user run⁷. In both runs, each stream executed the set of 46 common queries in the permutation defined by the specification. Due to the challenges discussed earlier of configuring multiple concurrent streams in Hive and Impala, it was decided to limit the multi-user test to only 4 streams. In our internal performance tests, Big SQL has successfully executed the same workload with 12 concurrent query streams.

Charts 3 and 4 summarise the elapsed times and effective query throughput of the single stream performance run on the 10TB dataset:

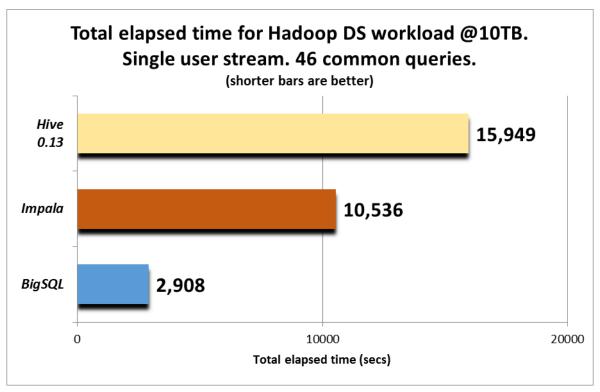


Chart 3: Hadoop-DS workload elapsed time @10TB across 46 common queries

Page | 14

⁷ The TPC-DS specification calls for 2 multi-stream runs, separated by the Data Maintenance phase of the benchmark. Since the Hadoop-DS benchmark does not have a Data Maintenance phase, the second multi-stream run would work on identical data to the first, and consequently would not provide any additional useful insight. Therefore the 2nd multi-stream run was dropped from the Hadoop-DS benchmark.

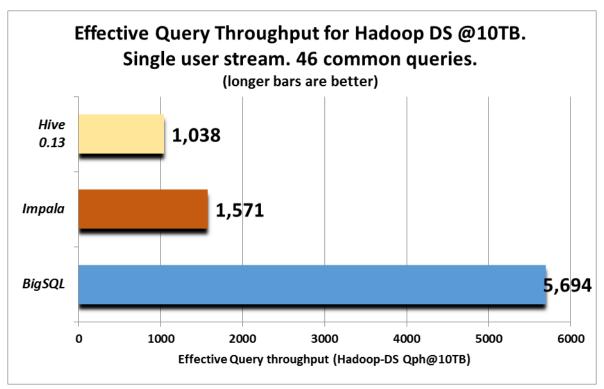


Chart 4: Hadoop-DS effective query throughput @10TB across 46 common queries

The results at 10TB show that Big SQL is 3.6x faster than Impala and 5.4x faster than Hive 0.13.

This is an impressive result for Big SQL, especially given the fact that many of the long running queries (where Big SQL is strongest) were removed from the workload because Impala or Hive could not complete them.

Charts 5 and 6 shows the elapsed time and effective query throughput for the Hadoop-DS workload executing 4 concurrent query streams at 10TB:

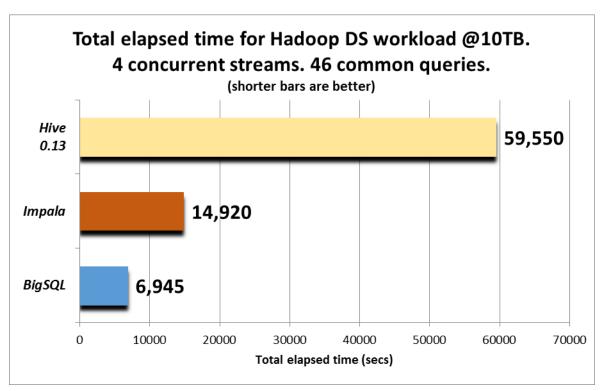


Chart 5: Workload elapsed time for 4 concurrent streams at 10TB across 46 common queries

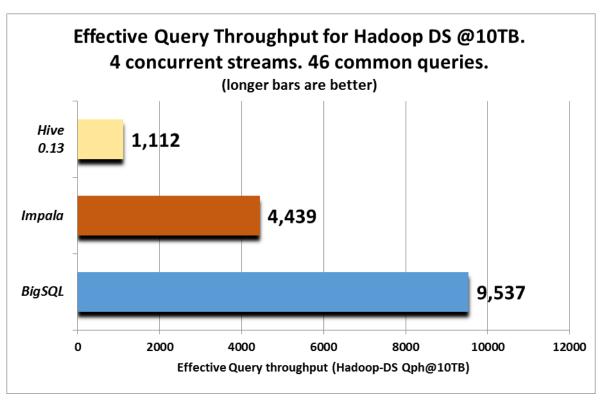


Chart 6: Effective query throughput for 4 concurrent streams at 10TB across 46 common queries

The results mirror those from the single stream run. Big SQL achieves the highest throughput and has 2.1x greater throughput than Impala and 8.5x more throughput than Hive.

10TB scale factor using all 99 queries

Since Big SQL supports a richer SQL syntax and was able to execute all 99 compliant queries in the workload, the team also executed a single stream and multi-stream run using the full set of 99 queries from the Hadoop-DS workload. For the single stream run, Big SQL completed execution of all 99 queries in approximately 8hrs and 59mins with an effective throughput rate of 1,101 Hadoop-DS Qph@10TB – as highlighted in chart 7:

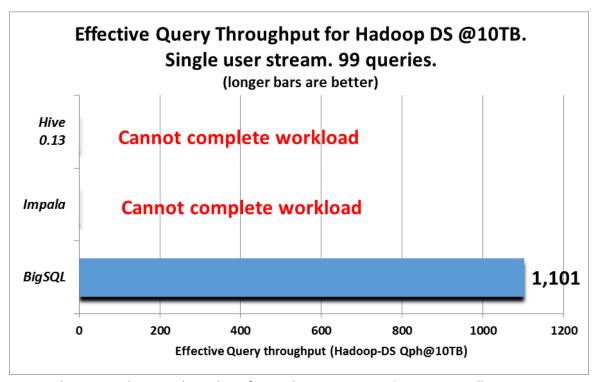


Chart 7: Hadoop-DS throughput for single query stream @10TB using all 99 queries

We used 6 concurrent query streams for the multi-stream run, in which Big SQL completed all 594 queries in just over 24hrs 39mins. This equates to an effective throughput rate of 2,409 Hadoop-DS Qph@10TB – as illustrated in Chart 8:

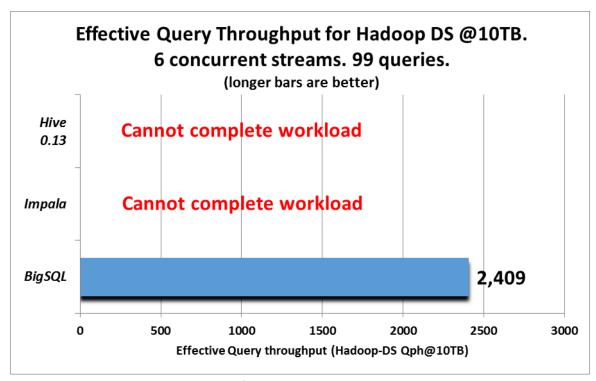


Chart 8: Hadoop DS throughput for 6 concurrent streams at 10TB using all 99 queries

Although the query throughput measured when running all 99 queries was lower than what was measured when running the subset of 46 queries, we need to consider that these are actually two different workloads. The entire 99 query workload included all of the complex, longer running queries which failed with Hive and/or Impala. This is highlighted by comparing the average elapsed times of the 46 queries in the common subset (64 sec. for Big SQL) with the average elapsed time of the remaining 53 queries from the full set (549 sec. for Big SQL). So not only are these throughput numbers for Big SQL very impressive, they were achieved on a workload that includes queries that neither Impala nor Hive could run in our testing.

Even more impressive is the scalability of Big SQL when running the complete workload. **Big SQL** took just 2.7x longer to complete the 6 times volume of queries compared to the single query stream run – as illustrated in chart 9:

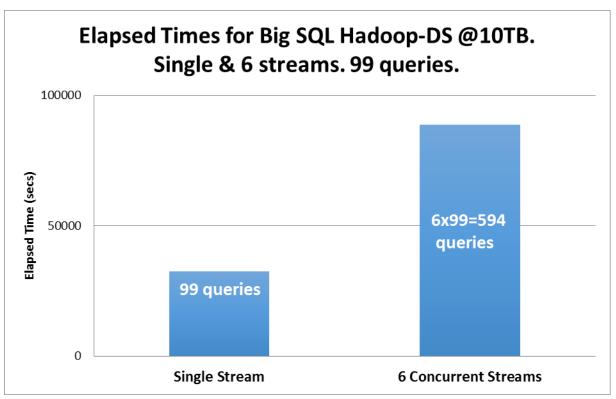


Chart 9: Big SQL multi-user scalability using 6 query streams @10TB

30TB scale factor using all 99 queries

Our original goal was to run a complete benchmark at 30TB, using all 99 queries, across all vendors. The 10TB dataset was to act as a stepping stone to achieve this. However, due to SQL limitations, query timeouts and errors that occurred when running Hive and Impala we had to compromise on the query set in order to get an apples-to-apples comparison at 10TB. It was the team's view that moving Hive and Impala to 30TB would reduce the common query set to such an extent as to make any comparison between the vendors meaningless. Consequently the team proceeded with a 30TB Hadoop-DS benchmark for Big SQL only.

At 30TB, (on the same 17-node cluster) Big SQL was again able to execute all 99 queries successfully in both the single and multi-user workloads. For the single user performance run, Big SQL's effective throughput rate was 1,023 Hadoop-DS Qph@30TB. For the multi-stream run using 4 concurrent streams Big SQL achieved a throughput of 2,274 Hadoop-DS Qph@30TB. Chart 10 summarises the elapsed times for the single and multi-user runs at 30TB:

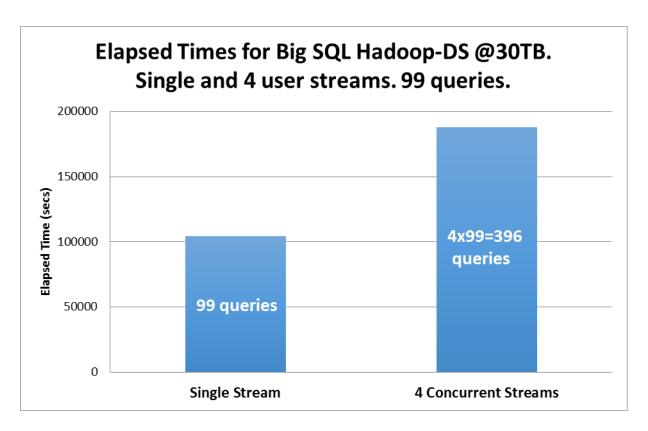


Chart 10: Big SQL multi-user scalability using 4 query streams @30TB

Of particular note is the fact that 4 concurrent query streams (and therefore 4 times more queries) only takes 1.8x longer than a single query stream at 30TB. Once again highlighting Big SQL's impressive multi-user scalability – this time at 30TB.

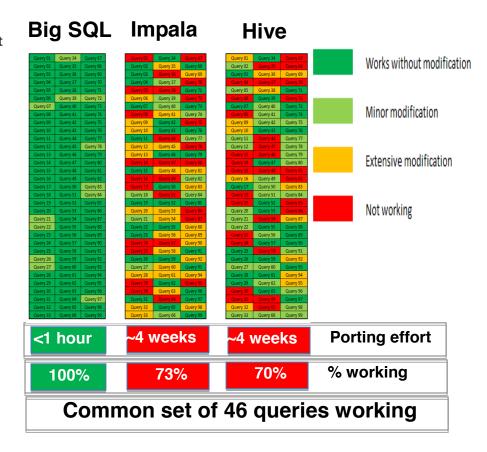
Summary

TPC-DS is the leading standardized analytic workload for RDBMs, which we adapted for Hadoop. We have named this derivative workload Hadoop-DS to avoid confusion and retained the services of a TPC certified auditor to validate fair comparisons.

As we have demonstrated in this white paper, IBM Big SQL was able to implement rich SQL with outstanding performance for both single and multiple concurrent users.

The heat map
highlights the fact that
Big SQL could run all
99 TPC-DS queries
without major
modification.
Something we were
not able to get the
other offerings to do.

The heat map also shows the enormous variance in effort it took for us to rewrite the queries in the benchmark and tune the other systems to be able to run the benchmark at 10TB with 4 concurrent users.



We have also shown leadership performance results with Big SQL for single user and multi user tests at 10TB of raw data. In addition Big SQL was the only tested solution able to complete the full workload at 10 and 30TB.

			Elapsed time (s)		Hadoop-DS Qph@10TB		Big SQL Advantage (x times faster) ⁸	
	# Queries		Single Stream	4-streams	Single Stream	4-streams	Single Stream	4-streams
Big SQL 3.0	46		2,908	6,945	5,694	9,537	-	-
Impala 1.4.1	46		10,536	14,920	1,571	4,439	3.6	2.1
Hive 0.13	46		15,949	59,550	1,038	1,112	5.4	8.5
i								
				All 99 d	queries	@ 10TB		
Big SQL 3.0	99		32,361	88,764	1,101	2,409		
Impala 1.4.1	99				Not Po	ossible		
Hive 0.13	99				Not Po	ossible		
·								
			All 99 queries @ 30TB					
Big SQL 3.0	99		104,445	187,993	1,023	2,274		
Impala 1.4.1	99		Not Possible					
Hive 0.13	99				Not Po	ossible		

These findings will be compelling to organizations augmenting data warehouse environments with Hadoop-based technologies. Not only was IBM's Big SQL the only Hadoop solution tested able to actually run the complete set of Hadoop-DS queries, but also we found it to be the fastest, the most scalable, and the most reliable as well.

Strict SQL compliance can translate into significant cost savings by allowing customers to leverage existing investments in databases, applications and skills and take advantage of SQL-over-Hadoop with minimal disruption to existing environments. Enterprise customers cannot afford to have different dialects of SQL across different data management platforms. In this testing, IBM's Big SQL demonstrated the highest degree of SQL language compatibility by a wide margin.

Not only was IBM Big SQL more compatible, but also it was significantly faster. This means that customers can realize business results faster, ask more complex questions, and realize great efficiencies per unit investment in infrastructure. All of these factors help provide a competitive advantage.

-

⁸ X times faster calculation based on comparison of Qph values.

IBM Big SQL is the industry leading SQL over Hadoop solution available today. We believe the performance and language richness coupled with strict adherence to the TPC rules demonstrate this capability. With Big SQL we have set the bar an order of magnitude higher than previous SQL over Hadoop approaches.

Appendix A: Workload times:

Table A.1 provides a summary of the benchmark results for the 46 common queries:

	IBM System x	Hadoop-DS ⁹ Nov 14, 2014			
Elapsed Times (secs)					
Test	Big SQL 3.0	Impala 1.4.1	Hive 0.13		
Database Ingest	11,100	6,420	39,780		
Run 1 – Single Stream	2,908	10,536	15,949		
Run 2 – Multi-Stream	6,945 14,920		59,550		
Effective Query Throughpu	t (Hadoop-DS Qph@10	DTB)			
Run 1 – Single Stream	5,694	1,571	1,038		
Run 2 – Multi-Stream	9,537	4,439	1,112		
Query Numbers					
Queries working	99	72	69		
Queries in common set	46				
Number Query Streams	4				

Table A.1: Summary of Hadoop-DS benchmark results @10TB

-

⁹ The Hadoop-DS benchmark is derived from TPC Benchmark DS (TPC-DS) and is not comparable to published TPC-DS results. TPC Benchmark is a trademark of the Transaction Processing Performance Council.

Appendix B: Cluster topology and hardware configuration



IBM System x3650

Each cluster consisted of 17 IBM x3650 M4 BD servers with 1 master node and 16 data nodes. Each server was configured with:

- CPU: e5-2680@2.8GHz v2 2 sockets, 10 cores each, hyper threading enabled = 40 logical CPUs
- Memory: 128 GB RAM at 1866 MHz
- Storage: 10 x 2TB 3.5" Serial SATA, 7200RPM. One disk for OS, 9 for data
- Storage: 4 x 128GB SSD (not used during benchmark)
- Network: Dual port 10 Gb Ethernet
- OS: Red Hat Enterprise Linux Server release 6.4 (Santiago)

Appendix C: Tuning

C.1 Big SQL Configuration

Installation options:

During install, the following Big SQL properties were set. *Node resource percentage* was set to 90% in order to provide as much of the cluster resources as possible to Big SQL:

```
Big SQL administrator user: bigsq
Big SQL FCM start port: 62000
Big SQL 1 server port: 7052
Scheduler service port: 7053
Scheduler administration port: 7054
Big SQL server port: 51000
Node resources percentage: 90%
```

The following are in-line with current BigInsights and Big SQL 3.0 best practices which recommend distributing all i/o for the Hadoop cluster across all disks:

```
BigSQL2 data directory:
```

 $\label{logsql} $$ \data1/db2/bigsql,\data2/db2/bigsql,\data3/db2/bigsql,\data4/db2/bigsql,\data5/db2/bigsql,\data5/db2/bigsql,\data5/db2/bigsql,\data7/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2/bigsql,\data9/db2$

```
Cache directory:
```

 $\label{local} $$ \data1/hadoop/mapred/local,/data2/hadoop/mapred/local,/data3/hadoop/mapred/local,/data4/hadoop/mapred/local,/data5/hadoop/mapred/local,/data6/hadoop/mapred/local,/data7/hadoop/mapred/local,/data8/hadoop/mapred/local,/data9/hadoop/mapred/local,/data9/hadoop/mapred/local,$

```
DataNode data directory:
```

 $\label{lem:data} $$ \data1/hadoop/hdfs/data, \data2/hadoop/hdfs/data, \data3/hadoop/hdfs/data, \data4/hadoop/hdfs/data, \data5/hadoop/hdfs/data, \data6/hadoop/hdfs/data, \data6/hadoop/hdfs/data, \data7/hadoop/hdfs/data, \data8/hadoop/hdfs/data, \data9/hadoop/hdfs/data$

Big SQL tuning options:

```
## Configured for 128 GB of memory per node
## 30 GB bufferpool
## 3.125 GB sortheap / 50 GB sheapthres shr
## reader memory: 20% of total memory by default (user can raise it to 30%)
##
## other useful conf changes:
    mapred-site.xml
##
       mapred.tasktracker.map.tasks.maximum=20
##
       mapred.tasktracker.reduce.tasks.maximum=6
       mapreduce.map.java.opts="-Xmx3000m ..."
       mapreduce.reduce.java.opts="-Xmx3000m ..."
##
    bigsql-conf.xml
       dfsio.num scanner threads=12
##
       dfsio.read size=4194304
       dfsio.num threads per disk=2
       scheduler.client.request.timeout=600000
##
```

```
db2 connect to ${DBNAME}
db2 -v "call syshadoop.big sql service mode('on')"
db2 -v "alter bufferpool IBMDEFAULTBP size 891520 "
## db2 -v "alter tablespace TEMPSPACE1 no file system caching"
db2 -v "update db cfg for ${DBNAME} using sortheap 819200 sheapthres shr 13107200"
db2 -v "update db cfg for ${DBNAME} using dft degree 8"
db2 -v "update dbm cfg using max_querydegree ANY"
db2 -v "update dbm cfg using aslheapsz 15"
db2 -v "update dbm cfg using cpuspeed 1.377671e-07"
db2 -v "update dbm cfg using INSTANCE_MEMORY 85"
## Added by Simon on 31/09. Useful for T'put run
db2 -v "update dbm cfg using CONN ELAPSE 18"
\#\# Added by Simon on 31/09. Disable auto maintenance
db2 -v "update db cfg for bigsql using AUTO MAINT OFF AUTO TBL MAINT OFF AUTO RUNSTATS OFF
AUTO STMT STATS OFF"
db2 terminate
```

BigInsights mapred-site.xml tuning:

The following changes (highlighted) were made to the Hadoop mapred-site.xml file to tune the number of mapreduce slots, and the maximum memory allocated to these slots. In Big SQL, Map-Reduce is used for the LOAD and ANALYZE commands only, not query execution. The properties were tuned in order to get the best possible performance from these commands.

```
cproperty>
    <!-- The maximum number of map tasks that will be run simultaneously by a
         task tracker. Default: 2. Recommendations: set relevant to number of
         CPUs and amount of memory on each data node. -->
    <name>mapred.tasktracker.map.tasks.maximum</name>
    <!--value><%= Math.max(2, Math.ceil(0.66 * Math.min(numOfDisks, numOfCores, totalMem/1000)
* 1.75) - 2) %></value-->
    <value>20</value>
 </property>
 property>
   <!-- The maximum number of reduce tasks that will be run simultaneously by
         a task tracker. Default: 2. Recommendations: set relevant to number of
         CPUs and amount of memory on each data node, note that reduces usually
         take more memory and do more I/O than maps. -->
    <name>mapred.tasktracker.reduce.tasks.maximum</name>
    <!--value><%= Math.max(2, Math.ceil(0.33 * Math.min(numOfDisks, numOfCores, totalMem/1000)
* 1.75) - 2)%></value-->
    <value>6</value>
```

```
</property>
  property>
    <!-- Max heap of child JVM spawned by tasktracker. Ideally as large as the
          task machine can afford. The default -Xmx200m is usually too small. -->
    <name>mapreduce.map.java.opts
    <value>-Xmx3000m -Xms1000m -Xmn100m -Xtune:virtualized -
Xshareclasses:name=mrscc_%g,groupAccess,cacheDir=/var/ibm/biginsights/hadoop/tmp,nonFatal -
Xscmx20m -Xdump:java:file=/var/ibm/biginsights/hadoop/tmp/javacore.%Y%m%d.%H%M%S.%pid.%seq.txt
-Xdump:heap:file=/var/ibm/biginsights/hadoop/tmp/heapdump.%Y%m%d.%H%M%S.%pid.%seq.phd</value>
  </property>
  property>
    <!-- Max heap of child JVM spawned by tasktracker. Ideally as large as the
          task machine can afford. The default -Xmx200m is usually too small. -->
    <name>mapreduce.reduce.java.opts
    <value>-Xmx3000m -Xms1000m -Xmn100m -Xtune:virtualized -
Xshareclasses:name=mrscc_%g,groupAccess,cacheDir=/var/ibm/biginsights/hadoop/tmp,nonFatal -
Xscmx20m -Xdump:java:file=/var/ibm/biginsights/hadoop/tmp/javacore.%Y%m%d.%H%M%S.%pid.%seq.txt
-Xdump:heap:file=/var/ibm/biginsights/hadoop/tmp/heapdump.%Y%m%d.%H%M%S.%pid.%seq.phd</value>
  </property>
```

Big SQL dfs reader options:

The following properties were changed in the Big SQL bigsql-conf.xml file to tune dfs reader properties:

```
cproperty>
    <name>scheduler.client.request.timeout</name>
    <value>600000</value>
    <description>
      Scheduler clients will wait for scheduler to respond for
      these many milli-seconds before timing out.
  </description>
  </property>
  property>
    <!-- Number of threads reading from each disk.
         Set this to 0 to use default values. -->
    <name>dfsio.num threads per disk</name>
    <value>2</value>
    <!--value>0</value-->
  </property>
  cproperty>
    <!-- Read Size (in bytes) - Size of the reads sent to Hdfs (i.e., also the max I/O read
buffer size).
         Default is 8*1024*1024 = 8388608 bytes -->
    <name>dfsio.read_size</name>
    <value>4194304</value>
    <!--value>8388608</value-->
  </property>
```

Big SQL dfs logging:

The minLogLevel property was changed in the Big SQL glog-dfsio.properties file to reduce the amount of logging by the dfs readers:

```
glog_enabled=true
log_dir=/var/ibm/biginsights/bigsql/logs
log_filename=bigsql-ndfsio.log

# 0 - INFO
# 1 - WARN
# 2 - ERROR
# 3 - FATAL
minloglevel=3
```

OS kernel changes:

```
echo 0 > /proc/sys/vm/swappiness
echo "net.ipv6.conf.all.disable_ipv6 = 1" >> /etc/sysctl.conf
```

Active Hadoop components:

In order to release valuable resources on the cluster only the following BigInsights components were started during the single and multi-stream runs: bigsql, Hadoop, hive, catalog, zookeeper and console.

C.2: Impala Configuration

Linux Configuration changes:

The following Linux kernel configuration changes were made to the Impala cluster:

```
echo 0 > /proc/sys/vm/swappiness

sysctl -w net.core.somaxconn=1024

echo "net.ipv6.conf.all.disable_ipv6 = 1" >> /etc/sysctl.conf
```

Cloudera Configuration:

The following table describes the Cloudera and Impala tuning undertaken during the benchmark.

Category	Property	Value	Description
Cloudera Manager/Hosts Configuration/Other	Memory Overcommit Validation Threshold	0.98	Threshold used when validating the allocation of RAM on a host. 0 means all of the memory is reserved for the system. 1 means none is reserved. Values can range from 0 to 1.
	Enable Cgroup- based Resource Management	true	Enables resource management using control groups (cgroups) for this host. Once toggled, roles on this host must be restarted for cgroups to be enabled or disabled. Per-resource controls can be found in the configuration pages of role configuration groups and individual roles. Cgroups are a feature of the Linux kernel, and as such, support varies by distribution; consult the Cloudera Manager documentation for details.
Dynamic Resource Pools	Name	tpcds	
	Scheduling Policy	FAIR	Schedules resources fairly based only on memory
	Max Running Queries	3	Maximum number of concurrently running queries in the pool (optional)

HDFS Configuration/Service- Wide	Zookeeper Service	Zookeeper	Name of the ZooKeeper service that this HDFS service depends on.
HDFS Configuration/Service- Wide/Advanced	HDFS Service Advanced Configuration Snippet (Safety Valve) for hdfs- site.xml	<pre><pre><pre><pre><pre><pre><pre><name>dfs.datanode.hdfs -blocks-metadata- enabled</name></pre></pre> </pre> <pre><value>true</value> </pre> </pre> <pre></pre> </pre></pre></pre>	For advanced use only, a string to be inserted into hdfs-site.xml. Applies to configurations of all roles in this service except client configuration.
HDFS Configuration/Service- Wide/Performance	DataNode Local Path Access Users	impala	Comma separated list of users allowed to do short circuit read. A short circuit read allows a client colocated with the data to read HDFS file blocks directly from HDFS. If empty, will default to the DataNode process' user.
HDFS Configuration/Balancer Default Group	Rebalancing Policy	BlockPool	The policy that should be used to rebalance HDFS storage. The default DataNode policy balances the storage at the DataNode level. This is similar to the balancing policy from prior releases. The BlockPool policy balances the storage at the block pool level as well as at the Datanode level. The BlockPool policy is relevant only to a Federated HDFS

			service.
HDFS	DataNode Data	/data1/dfa/dia	Comma-delimited list
Configuration/DataNode	DataNode Data Directory	/data1/dfs/dn, /data2/dfs/dn,	of directories on the
Default Group	Directory	/data3/dfs/dn,	local file system where
Delauit Group		/data4/dfs/dn,	the DataNode stores
		/data5/dfs/dn,	HDFS block data.
		/data6/dfs/dn,	Typical values are
		/data7/dfs/dn,	/data/N/dfs/dn for N =
		/data8/dfs/dn,	1, 2, 3 These
		/data9/dfs/dn	directories should be
			mounted using the
			noatime option and
			the disks should be
			configured using JBOD.
			RAID is not
			recommended.
	DataNode	3	The number of
	Failed Volumes		volumes that are
	Tolerated		allowed to fail before a
			DataNode stops
			offering service. By
			default, any volume
			failure will cause a
			DataNode to
			shutdown.
HDFS	Java Heap Size	512 MiB	Maximum size in bytes
Configuration/DataNode	of DataNode in		for the Java Process
Default Group/Resource	Bytes		heap memory. Passed
Management			to Java –Xmx.
	Mayira	1 C:D	The manifestory of the state of
	Maximum	1 GiB	The maximum amount
	Memory Used for Caching		of memory a DataNode
	TOT CACTILING		may use to cache data blocks in memory.
			·
			Setting it to 0 will disable caching.
			uisable cacillig.
HDFS	DataNode Data	755	Permissions for the
Configuration/DataNode	Directory		directories on the local
Default Group/Security	Permissions		file system where the
			DataNode stores its
			blocks. The
			permissions must be

			octal. 755 and 700 are
			typical values.
HDFS	DataNode Data	/data1/dfs/dn,	Comma-delimited list
Configuration/DataNode	Directory	/data2/dfs/dn,	of directories on the
Group 1		/data3/dfs/dn,	local file system where
		/data4/dfs/dn,	the DataNode stores
		/data5/dfs/dn,	HDFS block data.
		/data6/dfs/dn,	Typical values are
		/data7/dfs/dn,	/data/N/dfs/dn for N =
		/data8/dfs/dn,	1, 2, 3 These
		/data9/dfs/dn	directories should be
			mounted using the
			noatime option and
			the disks should be
			configured using JBOD.
			RAID is not
			recommended.
	DataNode	3	The number of
	Failed Volumes		volumes that are
	Tolerated		allowed to fail before a
	Toteracea		DataNode stops
			offering service. By
			default, any volume
			failure will cause a
			DataNode to
			shutdown.
			0.1000
HDFS	Java Heap Size	920 MiB	Maximum size in bytes
Configuration/DataNode	of DataNode in		for the Java Process
Group 1/Resource	Bytes		heap memory. Passed
Management			to Java –Xmx.
	Maximum	3658 MiB	The maximum amount
	Memory Used		of memory a DataNode
	for Caching		may use to cache data
			blocks in memory.
			Setting it to 0 will
			disable caching.
HDFS	DataNode Data	755	Permissions for the
Configuration/DataNode	Directory	, , , ,	directories on the local
Group 1/Security	Permissions		file system where the
Stoup 1/Jeculity	1 (11113310113		DataNode stores its
			blocks. The
			DIOCKS. THE

			permissions must be octal. 755 and 700 are typical values.
HDFS Configuration/Gateway Default Group	Use Trash	true	Move deleted files to the trash so that they can be recovered if necessary. The client side configuration takes effect only if the HDFS service-wide trash is disabled (NameNode Filesystem Trash Interval set to 0) and is ignored otherwise. The trash is not automatically emptied when enabled with this configuration.
HDFS Configuration/NameNode Default Group	NameNode Data Directories	/data1/dfs/nn, /data2/dfs/nn	Determines where on the local file system the NameNode should store the name table (fsimage). For redundancy, enter a comma-delimited list of directories to replicate the name table in all of the directories. Typical values are /data/N/dfs/nn where N=13.
HDFS Configuration/NameNode Default Group/Performance	NameNode Handler Count	55	The number of server threads for the NameNode.
	NameNode Service Handler Count	55	The number of server threads for the NameNode used for service calls. Only used when NameNode Service RPC Port is

			configured.
HDFS Configuration/NameNode Default Group/Ports and Addresses	NameNode Service RPC Port	8022	Optional port for the service-rpc address which can be used by HDFS daemons instead of sharing the RPC address used by clients.
HDFS Configuration/NameNode Default Group/Resource Management	Java Heap Size of Namenode in Bytes	2111 MiB	Maximum size in bytes for the Java Process heap memory. Passed to Java -Xmx.
HDFS Configuration/SecondaryN ameNode Default Group	HDFS Checkpoint Directory	/data1/dfs/snn	Determines where on the local file system the DFS SecondaryNameNode should store the temporary images to merge. For redundancy enter a comma-delimited list of directories to replicate the image in all of the directories. Typical values are /data/N/dfs/snn for N = 1, 2, 3
HDFS Configuration/SecondaryN ameNode Default Group/Resource Management	Java Heap Size of Secondary namenode in Bytes	2111 MiB	Maximum size in bytes for the Java Process heap memory. Passed to Java –Xmx.
Impala Configuration/Service- Wide	HDFS Service	HDFS	Name of the HDFS service that this Impala service instance depends on
	Hive Service	Hive	Name of the Hive service that this Impala service instance

			depends on
	Yarn Service for Resource Management	none	Name of YARN service to use for resource management integration between Impala and YARN. This service dependency and the existence of a Llama role is required for using said integration.
Impala Configuration/Service- Wide/Advanced	Maximum HBase Client Retries	30	Maximum number of HBase client retries for Impala. Used as a maximum for all operations such as fetching of the root region from the root RegionServer, getting a cell's value, and starting a row update. Overrides configuration in the HBase service.
	HBase RPC Timeout	30 seconds	Timeout in milliseconds for all HBase RPCs made by Impala. Overrides configuration in HBase service.
	Impala Command Line Argument Advanced Configuration Snippet (Safety Valve)	- Ilama_host=bigaperf180.s vl.ibm.com - queue_wait_timeout_ms= 1800000	For advanced use only, key-value pairs (one on each line) to be added (verbatim) to Impala Daemon command-line flags. Applies to all roles in this service. Key names should begin with a hyphen(-). For example: - log_filename=foo.log.

	Impala Service Environment Advanced Configuration Snippet (Safety	JAVA_TOOL_OPTIONS="- Xmx2G"	For advanced use only, key-value pairs (one on each line) to be inserted into a role's environment. Applies
	Valve)		to configuration of all roles in this service except client configuration.
Impala Configuration/Service- Wide/Performance	StateStore Subscriber Timeout	4 hour(s)	Time in seconds before Impala Daemon or Catalog Server times out with the StateStore.
Impala Configuration/Impala Catalog Server Default Group/Advanced	Catalog Server Hive Metastore Connection Timeout	4 hour(s)	Timeout for requests to the Hive Metastore Server from Catalog Server. Consider increasing this if you have tables with a lot of metadata and see timeout errors.
Impala Configuration/Impala Catalog Server Default Group/Monitoring	Catalog Server Connectivity Tolerance at startup	30 minutes(s)	The amount of time to wait for the Catalog Server to fully start up and connect to the StateStore before enforcing the connectivity check.
Impala Configuration/Impala Daemon Default Group	Impala Daemon Scratch Directories	/data1/impala/impalad, /data2/impala/impalad, /data3/impala/impalad, /data4/impala/impalad, /data5/impala/impalad, /data6/impala/impalad, /data7/impala/impalad, /data8/impala/impalad	Directories where Impala Daemon will write data such as spilling information to disk to free up memory. This can potentially be large amounts of data.
Impala	Impala Daemon	RESERVATION_REQUEST_	A list of key-value pairs

Configuration/Impala Daemon Default Group Impala Configuration/Impala Daemon Default Group/Advanced	Query Options Advanced Configuration Snippet (Safety Valve) Impala Daemon Hive Metastore Connection Timeout	TIMEOUT=1800000; 4 hour(s)	of additional query options to pass to the Impala Daemon command line, separated by ','. Timeout for requests to the Hive Metastore Server from Impala. Consider increasing this if you have a lot of metadata and see timeout errors.
Impala Configuration/Impala Daemon Default Group/Monitoring	Impala Daemon Connectivity Tolerance at Startup Query Monitoring Timeout: 50 second(s) The timeout used by Cloudera Manager Agent's query monitor when communicating with the Impala Daemon web server, specified in seconds.	Query Monitoring Timeout: 50 second(s) The timeout used by Cloudera Manager Agent's query monitor when communicating with the Impala Daemon web server, specified in seconds.	The amount of time to wait for the Impala Daemon to fully start up and connect to the StateStore before enforcing the connectivity check. Query Monitoring Timeout: 50 second(s) The timeout used by Cloudera Manager Agent's query monitor when communicating with the Impala Daemon web server, specified in seconds.
Impala Configuration/Impala Daemon Default Group/Monitoring	Impala Configuration/I mpala Daemon Default Group/Monitor	Impala Configuration/Impala Daemon Default Group/Monitoring	Impala Configuration/Impala Daemon Default Group/Monitoring

Query Monitoring Period: 1 second(s) The polling period of the Impala query Manager Agent, specified in seconds. If set to zero, query monitoring is disabled.	ing Query Monitoring Period: 1 second(s) The polling period of the Impala query Manager Agent, specified in seconds. If set to zero, query monitoring is disabled.	Query Monitoring Period: 1 second(s) The polling period of the Impala query Manager Agent, specified in seconds. If set to zero, query monitoring is disabled.	Query Monitoring Period: 1 second(s) The polling period of the Impala query Manager Agent, specified in seconds. If set to zero, query monitoring is disabled.
Impala Configuration/Impala Llama ApplicationMaster Default Group/Advanced	Resource Caching Idle Timeout	1 minute(s)	Timeout policy for resources being cached.
	Maximum Client Notification Retries	50	Maximum number of retries for a client notification. After the maximum number of client notification retries has been reached without success the client is considered lost ad all its reservations are released. A successful client notification resets.
Impala Configuration/Impala Llama ApplicationMaster Default Group/Performance	Thrift Transport Timeout	15 minute(s)	Socket time, in milliseconds, used Llama ApplicationMaster auxiliary service for all its server and client

			Thrift connections.
Hive Configuration/Service- Wide	MapReduce Service	YARN (MR2 Included)	MapReduce jobs are run against this service.
	Zookeeper Service	Zookeeper	Name of the ZooKeeper service hat this Hive service instance depends on.
Hive Configuration/Service- Wide/Hive Metastore Database	Hive Metastore Database Type	postgresql	Type of Hive Metastore database. Note that Derby is not recommended and Cloudera Impala does not support Derby.
	Hive Metastore Database Name	Hive	Name of Hive Metastore database.
	Hive Metastore Database Host	BigAPerf180.svl.ibm.com	Host name of Hive Metastore database.
	Hive Metastore Database Port	7432	Port number of Hive Metastore Database.
	Hive Metastore Database Password	xxxxxx	Password for Hive Metastore database.
ZooKeeper Configuration/Server Default Group/Resource Management	Java Heap Size of ZooKeeper Server in Bytes	920 MiB	Maximum size in bytes for the Java Process heap memory. Passed to Java –Xmx.

C.3: Hive Configuration

Tez was used as the Hive 0.13 execution engine.

The following list provides a summary of the configuration changes, after which the detailed tuning is documented:

- 1. The number of containers was set to 21

- Size of containers was 5GB.
 The available memory was set to 105GB, to satisfy 21 containers
 The Java heap for Reduce tasks was set 2x to Java heap Map tasks

- 5. Tez session timeout (both client and submit) were set to 14400, or 4 hrs timeout, so they never timeout during submit. When the system is very busy, the job remains in submit mode, and waits until it gets enough resource to run. If the job waited for a long time,. It would timeout.
- 6. A number of memory parameters in mapred-site.xml, yarn-site.xml, tez-site.xml andd hive-site.xml were set based on calculation based on the documentation and through experimentation based on 5G container.
- 7. HBase & Ozzie were stopped to free-up memory
- 8. Bucket tables with appropriate split size were used

mapred-site:

mapreduce.map.memory.mb=5120 mapreduce.map.java.opts=4096 mapreduce.reduce.memory.mb=10240 mapreduce.reduce.java.opts=8192 mapreduce.task.io.sort.mb=1792

yarn-site.xml:

yarn.scheduler.minimum-allocation-mb=5120 yarn.scheduler.maximum-allocation-mb=108544 yarn.nodemanager.resource.memory-mb=108544 yarn.app.mapreduce.am.resource.mb=10240 yarn.app.mapreduce.am.command-opts=8192

tez-site.xml:

tez.am.resource.memory.mb=5120 tez.am.java.opts=4096 tez.session.am.dag.submit.timeout.secs=14400 tez.session.client.timeout.secs=14400

hive-site.xml:

hive.tez.container.size=5120 hive.tez.java.opts=4608 hive.auto.convert.join.noconditionaltask.size=1342177000

Execution settings:

These settings were passed for each query execution in Hive. Since they are the same for all queries they do not constitue per query tuning.

set ambari.hive.db.schema.name=hive;
set fs.file.impl.disable.cache=true;
set fs.hdfs.impl.disable.cache=true;
set hive.auto.convert.join.noconditionaltask=true;
set hive.auto.convert.join=true;
set hive.auto.convert.sortmerge.join.noconditionaltask=true;
set hive.auto.convert.sortmerge.join=true;
set hive.auto.convert.sortmerge.join=true;
set hive.compactor.abortedtxn.threshold=1000;
set hive.compactor.check.interval=300;
set hive.compactor.delta.num.threshold=10;
set hive.compactor.delta.pct.threshold=0.1f;
set hive.compactor.initiator.on=false;
set hive.compactor.worker.threads=0;
set hive.compactor.worker.timeout=86400;

```
set hive.enforce.bucketing=true;
set hive.enforce.sorting=true;
set hive.enforce.sortmergebucketmapjoin=true;
set hive.exec.failure.hooks=org.apache.hadoop.hive.ql.hooks.ATSHook;
set hive.exec.post.hooks=org.apache.hadoop.hive.ql.hooks.ATSHook;
set hive.exec.pre.hooks=org.apache.hadoop.hive.ql.hooks.ATSHook;
set hive.execution.engine=tez;
set hive.limit.pushdown.memory.usage=0.04;
set hive.map.aggr=true;
set hive.mapjoin.bucket.cache.size=10000;
set hive.mapred.reduce.tasks.speculative.execution=false;
set hive.metastore.cache.pinobitypes=Table,Database,Type,FieldSchema,Order;
set hive.metastore.client.socket.timeout=60:
set hive.metastore.execute.setugi=true;
set hive.metastore.warehouse.dir=/apps/hive/warehouse;
set hive.optimize.bucketmapjoin.sortedmerge=false;
set hive.optimize.bucketmapjoin=true;
set hive.enforce.bucketmapjoin=true;
set hive.optimize.index.filter=true;
set hive.optimize.mapjoin.mapreduce=true;
set hive.optimize.reducededuplication.min.reducer=4;
set hive.optimize.reducededuplication=true;
set hive.orc.splits.include.file.footer=false;
set hive.security.authorization.enabled=false;
hive.security.metastore.authorization.manager=org.apache.hadoop.hive.ql.security.authorization.Stor
ageBasedAuthorizationProvider;
set hive.semantic.analyzer.factory.impl=org.apache.hivealog.cli.HCatSemanticAnalyzerFactory;
set hive.server2.enable.doAs=false;
set hive.server2.tez.default.queues=default;
set hive.server2.tez.initialize.default.sessions=false;
set hive.server2.tez.sessions.per.default.queue=1;
set hive.tez.input.format=org.apache.hadoop.hive.gl.io.HiveInputFormat:
set hive.input.format=org.apache.hadoop.hive.ql.io.BucketizedHiveInputFormat;
set hive.txn.manager=org.apache.hadoop.hive.ql.lockmgr.DummyTxnManager;
set hive.txn.max.open.batch=1000;
set hive.txn.timeout=600:
set hive.vectorized.execution.enabled=true;
set hive.vectorized.groupby.checkinterval=1024;
set hive.vectorized.groupby.flush.percent=1;
set hive.vectorized.groupby.maxentries=1024;
set hive.exec.parallel=true;
set hive.merge.mapfiles=true;
set mapred.output.compress=true;
set hive.optimize.tez=true;
set dfs.blocksize=1024000000;
```

C.4 OS Storage Configuration:

set hive.compute.query.using.stats=true;

The following script was used to create ext4 filesystems on all disks (used to store data) on all nodes in the cluster (inc. the master) for each product.

All three vendors used the same storage layout.

Note that the SSDs were not used during the benchmark.

```
#!/bin/bash
# READ / WRITE Performance tests for EXT4 file systems
# Author - Stewart Tate, tates@us.ibm.com
# Copyright (C) 2013, IBM Corp. All rights reserved.:
# the follow is server unique and MUST be adjusted!
drives=(b g h i j k l m n)
SSDdrives=(c d e f)
echo "Create EXT4 file systems, version 130213b"
pause()
 sleep 2
 # make ext4 file systems on HDDs
echo "Create EXT4 file systems on HDDs"
 for dev range in ${drives[@]}
  echo "y" | mkfs.ext4 -b 4096 -O dir index,extent /dev/sd$dev range
done
for dev range in ${drives[@]}
 parted /dev/sd$dev range print
done
pause
 # make ext4 file systems on SSDs
 echo "Create EXT4 file systems on SSDs"
 for dev range in ${SSDdrives[@]}
  echo "y" | mkfs.ext4 -b 4096 -O dir_index,extent /dev/sd$dev_range
for dev range in ${SSDdrives[@]}
 parted /dev/sd$dev range print
 echo "Partitions aligned(important for performance) if following returns 0:"
 blockdev --getalignoff /dev/sd$dev range
done
exit
The filesystems are then mounted using the following script:
#!/bin/bash
# READ / WRITE Performance tests for EXT4 file systems
# Author - Stewart Tate, tates@us.ibm.com
# Copyright (C) 2013, IBM Corp. All rights reserved.:
# the follow is server unique and MUST be adjusted!
drives=(b g h i j k l m n)
SSDdrives=(c d e f)
echo "Mount EXT4 file systems, version 130213b"
echo " "
pause()
```

```
sleep 2
echo "Create EXT4 mount points for HDDs"
for i in ${drives[@]}
  let j++
  mkdir /data$j
  mount -vs -t ext4 -o
nobarrier, noatime, nodiratime, nobh, nouser\_xattr, data=writeback, commit=100 \ /dev/sd\$i
/data$j
done
j=0
echo "Create EXT4 mount points for SSDs"
for i in ${SSDdrives[@]}
  let j++
  mkdir /datassd$j
  mount -vs -t ext4 -o
nobarrier,noatime,nodiratime,discard,nobh,nouser_xattr,data=writeback,commit=100
/dev/sd$i /datassd$j
done
echo "Done."
exit
```

Appendix D: Database Schema

D.1 Big SQL

Since the Parquet format does not support Date data type, VARCHAR(10) was used to store dates.

set	schema \$schema;			cc_zip	varchar(10)
crea	ate hadoop table call_cent	er	,	cc_country	varchar(20)
not	cc_call_center_sk null,	bigint	,	cc_gmt_offset	double
	cc_call_center_id null,	varchar(16)	,	cc_tax_percentage	double
1100	·	varchar(10)	STO	RED AS PARQUETFILE;	
	cc_rec_end_date	varchar(10)	crea	ate hadoop table catalog_p	age
	cc_closed_date_sk	bigint	not.	<pre>cp_catalog_page_sk null,</pre>	bigint
	cc_open_date_sk	bigint		cp_catalog_page_id null,	varchar(16)
,	cc_name	varchar(50)		•	bigint
,	cc_class	varchar(50)	,	cp_end_date_sk	bigint
,	cc_employees	bigint	,	cp_department	varchar(50)
,	cc_sq_ft	bigint	,	cp_catalog_number	bigint
,	cc_hours	varchar(20)	,	cp_catalog_page_number	bigint
,	cc_manager	varchar(40)	,	cp_description	varchar(100)
,	cc_mkt_id	bigint	,	cp_type	varchar(100)
,	cc_mkt_class	varchar(50)	STO	RED AS PARQUETFILE;	
,	cc_mkt_desc	varchar(100)	crea	ate hadoop table catalog_r	eturns
,	cc_market_manager	varchar(40)	(cr_returned_date_sk	bigint
,	cc_division	bigint	,	cr_returned_time_sk	bigint
,	cc_division_name	varchar(50)	,	cr_item_sk	bigint
,	cc_company	bigint	not	<pre>null, cr_refunded_customer_sk</pre>	bigint
,	cc_company_name	varchar(50)	,	cr_refunded_cdemo_sk	bigint
,	cc_street_number	varchar(10)	,	cr_refunded_hdemo_sk	bigint
,	cc_street_name	varchar(60)	,	cr_refunded_addr_sk	bigint
,	cc_street_type	varchar(15)	,	cr_returning_customer_sk	bigint
,	cc_suite_number	varchar(10)	,	cr_returning_cdemo_sk	bigint
,	cc_city	varchar(60)	,	cr_returning_hdemo_sk	bigint
,	cc_county	varchar(30)	′	cr_returning_addr_sk	bigint
′	cc_state	varchar(2)	,	cr_call_center_sk	bigint
,			,		

	cr_catalog_page_sk	bigint		cs_promo_sk	bigint
,	cr_ship_mode_sk	bigint	,	cs_order_number null,	bigint
,	cr_warehouse_sk	bigint		cs_quantity	bigint
,	cr_reason_sk	bigint	,	cs_wholesale_cost	double
,	cr_order_number	bigint	,	cs_list_price	double
noc	null, cr_return_quantity	bigint	,	cs_sales_price	double
,	cr_return_amount	double	,	cs_ext_discount_amt	double
,	cr_return_tax	double	,	cs_ext_sales_price	double
,	cr_return_amt_inc_tax	double	,	cs_ext_wholesale_cost	double
,	cr_fee	double	,	cs_ext_list_price	double
,	cr_return_ship_cost	double	,	cs_ext_tax	double
,	cr_refunded_cash	double	,	cs_coupon_amt	double
,	cr_reversed_charge	double	,	cs_ext_ship_cost	double
,	cr_store_credit	double	,	cs_net_paid	double
,	cr_net_loss	double	,	cs_net_paid_inc_tax	double
STO	RED AS PARQUETFILE;		,	cs_net_paid_inc_ship	double
cre	ate hadoop table catalog_s	ales	,	cs_net_paid_inc_ship_tax	double
(cs_sold_date_sk	bigint	,	cs_net_profit	double
,	cs_sold_time_sk	bigint	STO	RED AS PARQUETFILE;	
,	cs_ship_date_sk	bigint	crea	ate hadoop table customer	
,	cs_bill_customer_sk	bigint	no+	c_customer_sk null,	bigint
,	cs_bill_cdemo_sk	bigint		c_customer_id	varchar(16)
,	cs_bill_hdemo_sk	bigint	1100	null, c_current_cdemo_sk	bigint
,	cs_bill_addr_sk	bigint	,	c_current_hdemo_sk	bigint
,	cs_ship_customer_sk	bigint	,	c_current_addr_sk	bigint
,	cs_ship_cdemo_sk	bigint	,	c_first_shipto_date_sk	bigint
,	cs_ship_hdemo_sk	bigint	,	c_first_sales_date_sk	bigint
,	cs_ship_addr_sk	bigint	,	c_salutation	varchar(10)
,	cs_call_center_sk	bigint	,	c_first_name	varchar(20)
,	cs_catalog_page_sk	bigint	,	c_last_name	varchar(30)
,	cs_ship_mode_sk	bigint	,	c_preferred_cust_flag	varchar(1)
,	cs_warehouse_sk	bigint	,	c_birth_day	bigint
, no+	cs_item_sk null,	bigint	,	c_birth_month	bigint
110 C		I	,		

	c_birth_year	bigint	not	<pre>d_date_sk null,</pre>	bigint
,	c_birth_country	varchar(20)		d_date_id null,	varchar(16)
,	c_login	varchar(13)	1100	d_date	varchar(10)
,	c_email_address	varchar(50)	,	d_month_seq	bigint
,	c_last_review_date	bigint	,	d_week_seq	bigint
STO	RED AS PARQUETFILE;		,	d_quarter_seq	bigint
cre	ate hadoop table customer_	address	,	d_year	bigint
not	<pre>ca_address_sk null,</pre>	bigint	,	d_dow	bigint
	•	varchar(16)	,	d_moy	bigint
1100		varchar(10)	,	d_dom	bigint
,	ca_street_name	varchar(60)	,	d_qoy	bigint
,	ca_street_type	varchar(15)	,	d_fy_year	bigint
,	ca_suite_number	varchar(10)	,	d_fy_quarter_seq	bigint
,	ca_city	varchar(60)	,	d_fy_week_seq	bigint
,	ca_county	varchar(30)	,	d_day_name	varchar(9)
,	ca_state	varchar(2)	,	d_quarter_name	varchar(6)
,	ca_zip	varchar(10)	,	d_holiday	varchar(1)
,	ca_country	varchar(20)	,	d_weekend	varchar(1)
,	ca_gmt_offset	double	,	d_following_holiday	varchar(1)
,	ca_location_type	varchar(20)	,	d_first_dom	bigint
STO	RED AS PARQUETFILE;		,	d_last_dom	bigint
	ate hadoop table customer_	demographics	,	d_same_day_ly	bigint
(cd_demo_sk	bigint	,	d_same_day_lq	bigint
1100	•	varchar(1)	,	d_current_day	varchar(1)
,	cd_marital_status	varchar(1)	,	d_current_week	varchar(1)
,	cd_education_status	varchar(20)	,	d_current_month	varchar(1)
,	cd_purchase_estimate	bigint	,	d_current_quarter	varchar(1)
,	cd_credit_rating	varchar(10)	,	d_current_year	varchar(1)
,	cd_dep_count	bigint	STO	RED AS PARQUETFILE;	
,	cd_dep_employed_count	bigint	crea	ate hadoop table household	_demographics
,	cd_dep_college_count	bigint	,	hd_demo_sk	bigint
) STO	RED AS PARQUETFILE;		1101	null, hd_income_band_sk	bigint
cre (ate hadoop table date_dim		,	hd_buy_potential	varchar(15)

	hd_dep_count	bigint		i_units	varchar(10)
,	hd_vehicle_count	bigint	,	i_container	varchar(10)
) STO	RED AS PARQUETFILE;		,	i_manager_id	bigint
cre	ate hadoop table income_ba	nd	,	i_product_name	varchar(50)
(bigint	STO	RED AS PARQUETFILE;	
not	null, ib_lower_bound	bigint	cre	ate hadoop table promotion	
,	ib_upper_bound	bigint	no+	<pre>p_promo_sk null,</pre>	bigint
STO	RED AS PARQUETFILE;			p_promo_id	varchar(16)
cre	ate hadoop table inventory		not	null, p_start_date_sk	bigint
no+	<pre>inv_date_sk null,</pre>	bigint	,	p_end_date_sk	bigint
	inv_item_sk	bigint	,	p_item_sk	bigint
		bigint	,	p_cost	double
not	null, inv_quantity_on_hand	bigint	,	p_response_target	bigint
) STO	RED AS PARQUETFILE;		,	p_promo_name	varchar(50)
	ate hadoop table item		,	p_channel_dmail	varchar(1)
(-	bigint	,	p_channel_email	varchar(1)
		varchar(16)	,	p_channel_catalog	varchar(1)
not	null, i_rec_start_date	varchar(10)	,	p_channel_tv	varchar(1)
,	i_rec_end_date	varchar(10)	,	p_channel_radio	varchar(1)
,	i_item_desc	varchar(200)	,	p_channel_press	varchar(1)
,	i_current_price	double	,	p_channel_event	varchar(1)
,	i_wholesale_cost	double	,	p_channel_demo	varchar(1)
,	i_brand_id	bigint	,	p_channel_details	varchar(100)
,	i_brand	varchar(50)	,	p_purpose	varchar(15)
,	i_class_id	bigint	,	p_discount_active	varchar(1)
,	i_class	varchar(50)	STO	RED AS PARQUETFILE;	
,	i_category_id	bigint	crea	ate hadoop table reason	
,	i_category	varchar(50)	not	r_reason_sk null,	bigint
,	i_manufact_id	bigint		r_reason_id null,	varchar(16)
,	i_manufact	varchar(50)		r_reason_desc	varchar(100)
,	i_size	varchar(20)) STO	RED AS PARQUETFILE;	
,	i_formulation	varchar(20)	crea	ate hadoop table ship_mode	
,	i_color	varchar(20)	,	<pre>sm_ship_mode_sk null,</pre>	bigint

not	<pre>sm_ship_mode_id null,</pre>	varchar(16)	s_country	varchar(20)
1100	sm_type	varchar(30)	s_gmt_offset	double
,	sm_code	varchar(10)	s_tax_precentage	double
,	sm_carrier	varchar(20)	STORED AS PARQUETFILE;	
,	sm_contract	varchar(20)	create hadoop table store_ret	urns
) STO	RED AS PARQUETFILE;		sr_returned_date_sk	bigint
crea	ate hadoop table store	,	sr_return_time_sk	bigint
not	s_store_sk null,	bigint	sr_item_sk not null,	bigint
	·	varchar(16)	sr_customer_sk	bigint
1100	s_rec_start_date	varchar(10)	sr_cdemo_sk	bigint
,	s_rec_end_date	varchar(10)	sr_hdemo_sk	bigint
,	s_closed_date_sk	bigint	sr_addr_sk	bigint
,	s_store_name	varchar(50)	sr_store_sk	bigint
,	s_number_employees	bigint	sr_reason_sk	bigint
,	s_floor_space	bigint	sr_ticket_number not null,	bigint
,	s_hours	varchar(20)	sr_return_quantity	bigint
,	s_manager	varchar(40)	sr_return_amt	double
,	s_market_id	bigint	sr_return_tax	double
,	s_geography_class	varchar(100)	sr_return_amt_inc_tax	double
,	s_market_desc	varchar(100)	sr_fee	double
,	s_market_manager	varchar(40)	sr_return_ship_cost	double
,	s_division_id	bigint	sr_refunded_cash	double
,	s_division_name	varchar(50)	sr_reversed_charge	double
,	s_company_id	bigint	sr_store_credit	double
,	s_company_name	varchar(50)	sr_net_loss	double
,	s_street_number	varchar(10)	STORED AS PARQUETFILE;	
	s_street_name	varchar(60)	create hadoop table store_sal	es
,	s_street_type	varchar(15)	ss_sold_date_sk	bigint
,	s_suite_number	varchar(10)	ss_sold_time_sk	bigint
,	s_city	varchar(60)	ss_item_sk not null,	bigint
,	s_county	varchar(30)	ss_customer_sk	bigint
,	s_state	varchar(2)	ss_cdemo_sk	bigint
	s_zip	varchar(10)	ss_hdemo_sk	bigint
,		Ι ,		

ss_addr_sk	bigint	w_warehouse_name	varchar(20)
, ss_store_sk	bigint	w_warehouse_sq_ft	bigint
, ss_promo_sk	bigint	w_street_number	varchar(10)
ss_ticket_number	er bigint	w_street_name	varchar(60)
ss_quantity	bigint	w_street_type	varchar(15)
ss_wholesale_co	ost double	w_suite_number	varchar(10)
ss_list_price	double	w_city	varchar(60)
ss_sales_price	double	w_county	varchar(30)
ss_ext_discount	_amt double	w_state	varchar(2)
ss_ext_sales_pr	rice double	w_zip	varchar(10)
ss_ext_wholesal	e_cost double	w_country	varchar(20)
ss_ext_list_pri	ce double	w_gmt_offset	double
ss_ext_tax	double	STORED AS PARQUETFILE;	
ss_coupon_amt	double	create hadoop table web_page	
, ss_net_paid	double	<pre>wp_web_page_sk not null,</pre>	bigint
ss_net_paid_ind	c_tax double	wp_web_page_id not null,	varchar(16)
ss_net_profit	double	wp_rec_start_date	varchar(10)
STORED AS PARQUETF	LE;	wp_rec_end_date	varchar(10)
create hadoop table	e time_dim	wp_creation_date_sk	bigint
t_time_sk not null,	bigint	wp_access_date_sk	bigint
t_time_id not null,	varchar(16)	wp_autogen_flag	varchar(1)
t_time	bigint	wp_customer_sk	bigint
t_hour	bigint	wp_url	varchar(100)
t_minute	bigint	wp_type	varchar(50)
t_second	bigint	wp_char_count	bigint
t_am_pm	varchar(2)	wp_link_count	bigint
t_shift	varchar(20)	wp_image_count	bigint
t_sub_shift	varchar(20)	wp_max_ad_count	bigint
t_meal_time	varchar(20)	STORED AS PARQUETFILE;	
STORED AS PARQUETFI	LE;	create hadoop table web_retur	ns
create hadoop table	e warehouse	wr_returned_date_sk	bigint
w warehouse sk	bigint	wr_returned_time_sk	bigint
not null,	3	,	

	wr_refunded_customer_sk	bigint		ws_ship_hdemo_sk	bigint
,	wr_refunded_cdemo_sk	bigint	,	ws_ship_addr_sk	bigint
,	wr_refunded_hdemo_sk	bigint	,	ws_web_page_sk	bigint
,	wr_refunded_addr_sk	bigint	,	ws_web_site_sk	bigint
,	wr_returning_customer_sk	bigint	,	ws_ship_mode_sk	bigint
,	wr_returning_cdemo_sk	bigint	,	ws_warehouse_sk	bigint
,	wr_returning_hdemo_sk	bigint	,	ws_promo_sk	bigint
,	wr_returning_addr_sk	bigint	not	ws_order_number null,	bigint
,	wr_web_page_sk	bigint	-	ws_quantity	bigint
,	wr_reason_sk	bigint	,	ws_wholesale_cost	double
not	<pre>wr_order_number null,</pre>	bigint	,	ws_list_price	double
	wr_return_quantity	bigint	,	ws_sales_price	double
,	wr_return_amt	double	,	ws_ext_discount_amt	double
,	wr_return_tax	double		ws_ext_sales_price	double
,	wr_return_amt_inc_tax	double	,	ws_ext_wholesale_cost	double
,	wr_fee	double	,	ws_ext_list_price	double
,	wr_return_ship_cost	double	,	ws_ext_tax	double
,	wr_refunded_cash	double	,	ws_coupon_amt	double
,	wr_reversed_charge	double	,	ws_ext_ship_cost	double
,	wr_account_credit	double	,	ws_net_paid	double
)	wr_net_loss	double	,	ws_net_paid_inc_tax	double
STO	RED AS PARQUETFILE;		,	ws_net_paid_inc_ship	double
cre	ate hadoop table web_sales		,	ws_net_paid_inc_ship_tax	double
	ws_sold_date_sk	bigint)	ws_net_profit	double
,	ws_sold_time_sk	bigint	STO	RED AS PARQUETFILE;	
,	ws_ship_date_sk	bigint	crea	ate hadoop table web_site	
not	ws_item_sk null,	bigint	not	web_site_sk null,	bigint
	ws_bill_customer_sk	bigint		web_site_id null,	varchar(16)
,	ws_bill_cdemo_sk	bigint	,	web_rec_start_date	varchar(10)
	ws_bill_hdemo_sk	bigint	,	web_rec_end_date	varchar(10)
,	ws_bill_addr_sk	bigint		web_name	varchar(50)
,	ws_ship_customer_sk	bigint		web_open_date_sk	bigint
	ws_ship_cdemo_sk	bigint		web_close_date_sk	bigint
,		ı	,		

```
web class
                          varchar(50)
                                                   web suite number
                                                                              varchar(10)
web manager
                          varchar(40)
                                                   web city
                                                                              varchar(60)
web mkt id
                                                   web county
                                                                              varchar(30)
                          bigint
web mkt class
                          varchar(50)
                                                   web state
                                                                              varchar(2)
web mkt desc
                          varchar(100)
                                                   web zip
                                                                              varchar(10)
web market manager
                          varchar(40)
                                                   web country
                                                                              varchar(20)
web_company_id
                          bigint
                                                   web_gmt_offset
                                                                              double
web company name
                          varchar(50)
                                                   web_tax_percentage
                                                                              double
                                                STORED AS PARQUETFILE;
web street number
                          varchar(10)
web street name
                          varchar(60)
                                                commit;
                          varchar(15)
web street type
```

D.2 Impala

```
#!/bin/bash
impala-shell -d tpcds10000g <<EOF</pre>
create external table et store sales
  ss sold date sk
                             int,
  ss sold time_sk
                            int,
  ss item sk
                            int.
  ss customer sk
                             int.
  ss_cdemo_sk
                            int,
  ss hdemo sk
                             smallint,
  ss addr sk
                             int,
  ss_store_sk
                             smallint,
                             smallint,
  ss_promo_sk
  ss ticket number
                            bigint,
                             bigint,
  ss quantity
 ss wholesale cost
decimal(7,2),
  ss list price
decimal(7,2),
  ss_sales_price
decimal(7,2),
  ss ext discount amt
decimal(7,2),
  ss_ext_sales_price
decimal(7,2),
 ss_ext_wholesale_cost
decimal(7,2),
 ss ext list price
decimal(7,2),
 ss_ext_tax
decimal(7,2),
 ss coupon amt
decimal(7,2),
  ss net paid
decimal(7,2),
```

```
ss_net_paid_inc_tax
decimal(7,2),
  ss net profit
decimal(7,2)
row format delimited fields terminated
bv 'l'
location '/tpcds10000g/store sales'
tblproperties
('serialization.null.format'='')
create external table
et_customer_demographics
 cd demo sk
                           int,
 cd gender
                           string,
 cd marital status
                           string,
 cd_education_status
                           string,
  cd purchase estimate
                           bigint,
 cd credit rating
                            string,
 cd_dep_count
                            bigint,
                            bigint,
  cd_dep_employed_count
  cd_dep_college_count
                            bigint
row format delimited fields terminated
by '|'
location
'/tpcds10000g/customer_demographics'
tblproperties
('serialization.null.format'='')
create external table et date dim
```

```
d_date_sk
                           int,
 d date id
                           string,
-- d date
                            string,
                                           create external table et item
-- YYYY-MM-DD format
                                            i_item sk
 d date
                           timestamp,
                                                                      int,
-- YYYY-MM-DD format
                                             i_item_id
                                                                      string,
 d month seq
                          bigint,
                                            i rec start date
                                                                      timestamp,
 d week seq
                          bigint,
                                            i rec end date
                                                                      timestamp,
 d quarter seq
                          bigint,
                                             i item desc
                                                                      string,
 d year
                          bigint,
                                             i current price
 d dow
                          bigint,
                                           decimal(7,2),
                                             i wholesale cost
 d moy
                          bigint,
 d dom
                          bigint,
                                           decimal(7,2),
                          bigint,
                                             i brand id
                                                                      bigint,
 d qoy
 d_fy_year
                          bigint,
                                             i brand
                                                                      string,
 d_fy_quarter_seq
                          bigint,
                                             i_class_id
                                                                      bigint,
                                             i_class
 d_fy_week_seq
                          bigint,
                                                                      string,
 d_day_name
                          string,
                                             i category id
                                                                     bigint,
 d quarter name
                          string,
                                             i category
                                                                     string,
                                             i manufact id
 d holiday
                          string,
                                                                     bigint,
 d weekend
                                             i manufact
                          string,
                                                                     string,
 d following holiday
                                             i size
                          string,
                                                                      string,
 d first dom
                                             i formulation
                          bigint,
                                                                      string,
 d last dom
                          bigint,
                                             i color
                                                                      string,
 d same day ly
                          bigint,
                                             i units
                                                                      string,
 d same day lq
                          bigint,
                                             i container
                                                                      string,
 d current day
                          string,
                                             i manager id
                                                                      bigint,
                                                                      string
 d_current_week
                                             i_product_name
                          string,
 d_current_month
                          string,
 d current quarter
                          string,
                                           row format delimited fields terminated
 d_current_year
                                           by '|'
                           string
                                           location '/tpcds10000g/item'
row format delimited fields terminated
                                           tblproperties
                                           ('serialization.null.format'='')
location '/tpcds10000g/date dim'
tblproperties
('serialization.null.format'='')
                                           create external table et store
;
                                             s store sk
                                                                      smallint,
create external table et time dim
                                             s store id
                                                                      string,
                                             s rec start date
                                                                      timestamp,
 t time sk
                           int,
                                             s_rec_end_date
                                                                      timestamp,
 t time id
                          string,
                                             s_closed_date_sk
                                                                      int,
 {\sf t\_time}
                          bigint,
                                             s_store_name
                                                                      string,
                                             s number employees
 t hour
                          bigint,
                                                                      bigint,
 t minute
                          bigint,
                                             s_floor_space
                                                                      bigint,
 t second
                          bigint,
                                             s hours
                                                                      string,
 t_am_pm
                          string,
                                             s manager
                                                                      string,
 t_shift
                          string,
                                             s market id
                                                                      bigint,
                                             s geography_class
 t sub shift
                          string,
                                                                     string,
 t meal time
                          string
                                             s market desc
                                                                     string,
                                             s market manager
                                                                      string,
row format delimited fields terminated
                                             s division id
                                                                     bigint,
                                             s division name
                                                                      string,
location '/tpcds10000g/time_dim'
                                             s_company_id
                                                                      bigint,
tblproperties
                                             s_company_name
                                                                      string,
('serialization.null.format'='')
                                             s street number
                                                                      string,
```

```
s_street_name
                         string,
                                           p_promo_name
                                                                    string,
                                            p_channel_dmail
 s street type
                          string,
                                                                    string,
 s suite number
                          string,
                                            p_channel_email
                                                                    string,
 s city
                                            p_channel_catalog
                          string,
                                                                    string,
                                           p_channel_tv
 s_county
                          string,
                                                                   string,
 s_state
                         string,
                                           p_channel_radio
                                                                   string,
                         string,
                                           p_channel_press
 s_zip
                                                                   string,
                                           p channel event
 s country
                          string,
                                                                   string,
 s gmt offset
                                            p channel demo
decimal(5,2),
                                            p channel details
                                                                   string,
                                            p_purpose
 s tax precentage
                                                                    string,
decimal(5,2)
                                            p_discount_active
                                                                   string
row format delimited fields terminated
                                          row format delimited fields terminated
by '|'
                                          by '|'
location '/tpcds10000g/store'
                                          location '/tpcds10000g/promotion'
tblproperties
                                          tblproperties
                                          ('serialization.null.format'='')
('serialization.null.format'='')
                                          create external table
create external table et customer
                                          et household demographics
 c customer sk
                          int.
 c_customer_id
                                           hd demo sk
                          string,
                                                                    smallint,
                                           hd income_band_sk
 c current cdemo sk
                          int,
                                                                    tinyint,
 c current hdemo sk
                          smallint,
                                           hd buy potential
                                                                    string,
 c current addr sk
                          int,
                                           hd dep count
                                                                    bigint,
                                           hd_vehicle_count
 c_first_shipto_date_sk int,
                                                                   bigint
 c_first_sales_date_sk int,
                                          row format delimited fields terminated
 c salutation
                          string,
 c first name
                                          by '|'
                         string,
 c last name
                         string,
                                          location
                                          '/tpcds10000g/household_demographics'
 c_preferred_cust_flag string,
 c_birth_day
                         bigint,
                                          tblproperties
 c birth month
                                          ('serialization.null.format'='')
                         bigint,
                         bigint,
 c birth year
 c birth country
                         string,
 c login
                         string.
 c_email address
                         string,
                                          create external table
 c last review date
                                          et customer address
                         int.
row format delimited fields terminated
                                            ca address sk
                                            ca address id
                                                                   string,
location '/tpcds10000g/customer'
                                            ca_street_number
                                                                   string,
                                            ca_street_name
                                                                    string,
tblproperties
('serialization.null.format'='')
                                            ca_street_type
                                                                    string,
                                            ca_suite_number
                                                                    string,
                                            ca city
                                                                    string,
                                            ca county
                                                                    string,
create external table et promotion
                                            ca_state
                                                                    string,
                                            ca zip
                                                                    string,
(
 p promo sk
                          smallint,
                                            ca country
                                                                    string,
 p promo id
                          string,
                                            ca gmt offset
                                          decimal(5,2),
 p start date sk
                         int,
 p end date sk
                                           ca location type string
                          int.
 p_item_sk
                          int,
                                          row format delimited fields terminated
 p cost
                          double,
 p response target
                       bigint,
                                          by '|'
```

```
location
'/tpcds10000g/customer address'
tblproperties
('serialization.null.format'='')
create external table et inventory
 inv date sk
 inv item sk
                          int,
inv warehouse_sk
                         tinyint,
inv quantity on hand
                        bigint
row format delimited fields terminated
location '/tpcds10000g/inventory'
tblproperties
('serialization.null.format'='')
create external table et call center
 cc call center sk
                         tinyint,
 cc call center id
                          string,
                         string,
 cc rec start date
 cc rec end date
                         string,
 cc_closed_date_sk
                         int,
 cc open date sk
                          int,
 cc name
                          string,
 cc class
                          string,
 cc employees
                         bigint,
                         bigint,
 cc_sq_ft
 cc_hours
                         string,
 cc manager
                         string,
 cc_mkt_id
                         bigint,
 cc_mkt class
                         string,
                         string,
 cc mkt desc
 cc_market_manager
                         string,
 cc division
                         bigint,
 cc division name
                         string,
                         bigint,
 cc company
 cc_company_name
                         string,
 cc_street_number
                         string,
 cc_street_name
                          string,
 cc_street_type
                          string,
 ca_suite_number
                          string,
 cc city
                          string,
 cc county
                          string,
 cc_state
                          string,
 cc zip
                          string,
 cc country
                          string,
 cc gmt offset
decimal(5,2),
 cc tax precentage
decimal(5,2)
)
```

```
row format delimited fields terminated
by '|'
location '/tpcds10000g/call center'
tblproperties
('serialization.null.format'='')
create external table et catalog page
 cp catalog page sk
 cp_catalog_page_id
                          string,
 cp start date sk
                          int,
 cp end date sk
                          int,
 cp department
                          string,
 cp_catalog_number
                          bigint,
 cp_catalog_page_number bigint,
 cp_description
                          string,
 cp_type
                          string
row format delimited fields terminated
location '/tpcds10000g/catalog_page'
tblproperties
('serialization.null.format'='')
create external table
et catalog returns
cr returned date sk
                          int,
cr returned time sk
                          int,
 cr_item_sk
                          int,
 cr_refunded_customer_sk int,
 cr refunded cdemo sk
                          int,
                          smallint,
 cr refunded hdemo sk
 cr refunded addr sk
                          int,
 cr returning customer sk int,
 cr_returning_cdemo_sk int,
 cr returning hdemo sk
                          smallint,
 cr returning addr sk
                          int,
 cr call center sk
                          tinyint,
 cr_catalog_page_sk
                          int,
 cr_ship_mode_sk
                          tinyint,
 cr_warehouse_sk
                          tinyint,
 cr reason sk
                          tinyint,
 cr_order_number
                          bigint,
 cr return quantity
                          bigint,
 cr return amount
decimal(7,2),
 cr return tax
decimal(7,2),
cr return amt inc tax
decimal(7,2),
 cr_return fee
decimal(7,2),
  cr return ship cost
decimal(7,2),
```

```
cr_refunded_cash
decimal(7,2),
 cr reversed charge
decimal(7,2),
 cr store credit
decimal(7,2),
 cr_net_loss
decimal(7,2)
row format delimited fields terminated
location
'/tpcds10000g/catalog returns'
tblproperties
('serialization.null.format'='')
;
create external table et_catalog_sales
 cs sold date sk
 cs sold time sk
                          int,
 cs ship date sk
                          int,
 cs bill customer sk
                          int,
                         int,
 cs bill cdemo sk
 cs bill hdemo sk
                           smallint,
 cs bill addr sk
                           int,
 cs ship customer sk
                           int,
 cs_ship_cdemo_sk
                           int,
 cs ship hdemo sk
                           smallint,
 cs ship addr sk
                          int,
 cs call center sk
                         tinyint,
 cs_catalog_page_sk
                         int,
 cs_ship_mode_sk
                          tinyint,
 cs_warehouse_sk
                         tinyint,
                          int,
 cs item sk
 cs_promo_sk
                           smallint,
 cs order number
                          bigint,
 cs quantity
                           bigint,
 cs_wholesale_cost
decimal(7,2),
 cs list price
decimal(7,2),
 cs sales price
decimal(7,2),
 cs_ext_discount_amt
decimal(7,2),
 cs_ext_sales_price
decimal(7,2),
 cs ext wholesale cost
decimal(7,2),
 cs ext list price
decimal(7,2),
 cs ext tax
decimal(7,2),
 cs coupon amt
decimal(7,2),
 cs ext ship cost
decimal(7,2),
```

```
cs_net_paid
decimal(7,2),
 cs net paid inc tax
decimal(7,2),
 cs net paid inc ship
decimal(7,2),
  cs_net_paid_inc_ship_tax
decimal(7,2),
  cs net profit
decimal(7,2)
row format delimited fields terminated
by '|'
location '/tpcds10000g/catalog sales'
tblproperties
('serialization.null.format'='')
create external table et income band
  ib income band sk
                          tinvint,
  ib lower bound
                          bigint,
 ib_upper_bound
                          bigint
row format delimited fields terminated
by '|'
location '/tpcds10000g/income band'
tblproperties
('serialization.null.format'='')
create external table et_ship_mode
 sm ship mode sk
                           tinyint,
 sm ship mode id
                          string,
 sm type
                          string,
                          string,
 sm code
                          string,
 sm carrier
  sm contract
                          string
row format delimited fields terminated
location '/tpcds10000g/ship mode'
tblproperties
('serialization.null.format'='')
create external table et web page
 wp web page sk
                          smallint,
  wp web page id
                          string,
  wp rec start date
                          timestamp,
                          timestamp,
  wp rec end date
                          int,
  wp_creation_date_sk
  wp access date sk
                          int,
  wp autogen flag
                           string,
```

```
wp_customer_sk
                          int,
 wp url
                           string,
 wp type
                           string,
 wp_char_count
                           bigint,
 wp_link_count
                           bigint,
 wp_image_count
                           bigint,
 wp_max_ad_count
                           bigint
row format delimited fields terminated
location '/tpcds10000g/web page'
tblproperties
('serialization.null.format'='')
;
create external table et_store_returns
 sr returned date sk
                          int,
 sr return time sk
                          int,
 sr item sk
                           int,
 sr customer sk
                          int,
 sr cdemo sk
                           int,
                           smallint,
 sr hdemo sk
 sr addr sk
                           int,
 sr store sk
                           smallint,
 sr reason sk
                           tinyint,
                           bigint,
 sr ticket number
 sr_return_quantity
                           bigint,
 sr return amt
decimal(7,2),
 sr return tax
decimal(7,2),
 sr_return_amt_inc_tax
decimal(7,2),
 sr return fee
decimal(7,2),
 sr return ship cost
decimal(7,2),
 sr refunded cash
decimal(7,2),
 sr reversed charge
decimal(7,2),
 sr store credit
decimal(7,2),
 sr_net_loss
decimal(7,2)
row format delimited fields terminated
by '|'
location '/tpcds10000g/store returns'
tblproperties
('serialization.null.format'='')
create external table et web returns
 wr returned date sk
                           int,
```

```
wr_returned_time_sk
                           int,
 wr item sk
                           int,
 wr refunded customer sk
                           int,
 wr refunded cdemo sk
                           int.
 wr_refunded_hdemo_sk
                           smallint,
 wr refunded addr sk
                           int,
 wr_returning_customer_sk int,
 wr returning cdemo sk
                           int,
 wr returning hdemo sk
                           smallint,
 wr_returning_addr_sk
                          int,
 wr web page sk
                           smallint,
 wr reason sk
                           tinyint,
 wr order number
                           bigint,
  wr return quantity
                           bigint,
  wr return amt
decimal(7,2),
 wr return tax
decimal(7,2),
 wr_return_amt_inc_tax
decimal(7,2),
wr fee
decimal(7,2),
 wr return ship cost
decimal(7,2),
 wr refunded cash
decimal(7,2),
wr reversed charge
decimal(7,2),
wr_account_credit
decimal(7,2),
 wr net loss
decimal(7,2)
row format delimited fields terminated
by '|'
location '/tpcds10000g/web returns'
tblproperties
('serialization.null.format'='')
create external table et web sales
 ws sold date sk
 ws_sold_time_sk
                           int,
 ws_ship_date_sk
                           int,
 ws item sk
                           int,
 ws_bill_customer_sk
                           int,
 ws_bill_cdemo_sk
                           int,
 ws bill hdemo sk
                           smallint,
 ws_bill_addr_sk
                           int,
 ws ship customer sk
                           int,
 ws ship cdemo sk
                           int,
  ws ship hdemo sk
                           smallint,
 ws ship addr sk
                           int,
 ws web page sk
                           smallint,
 ws_web_site_sk
                           tinyint,
 ws_ship_mode_sk
                           tinyint,
 ws warehouse sk
                           tinyint,
```

```
ws_promo_sk
                           smallint,
                                             ws_net_profit
 ws order number
                           bigint,
                                            decimal(7,2)
 ws quantity
                           bigint,
 ws wholesale cost
                                            row format delimited fields terminated
                                            by '|'
decimal(7,2),
 ws list price
                                            location '/tpcds10000g/web sales'
decimal(7,2),
                                            tblproperties
                                            ('serialization.null.format'='')
 ws sales price
decimal(7,2),
 ws ext discount amt
decimal(7,2),
 ws ext sales price
                                            create external table et web site
decimal(7,2),
 ws ext wholesale cost
                                              web site sk
                                                                       tinyint,
decimal(7,2),
                                              web site id
                                                                       string,
 ws_ext_list_price
                                                                      timestamp,
                                              web_rec_start_date
decimal(7,2),
                                              web_rec_end_date
                                                                      timestamp,
 ws_ext_tax
                                              web_name
                                                                      string,
decimal(7,2),
                                              web open date sk
                                                                       int,
 ws coupon amt
                                              web close date sk
decimal(7,2),
                                              web class
                                                                      string,
 ws ext_ship_cost
                                              web manager
                                                                       string,
decimal(7,2),
                                              web mkt id
                                                                      bigint,
 ws net paid
                                              web mkt class
                                                                       string,
decimal(7,2),
                                              web mkt desc
                                                                       string,
 ws net paid inc tax
                                              web market manager
                                                                       string,
decimal(7,2),
                                              web company id
                                                                       bigint,
 ws net paid inc ship
                                              web company name
                                                                       st.
decimal(7,2),
 ws net paid inc ship tax
decimal(7,2),
 web street number
                          string,
                                            row format delimited fields terminated
                                            by '|'
 web_street_name
                          string,
                                            location '/tpcds10000g/reason'
 web_street_type
                          string,
                         string,
                                            tblproperties
 web suite number
                          string,
                                            ('serialization.null.format'='')
 web city
                          string,
 web county
                          string,
 web state
 web_zip
                          string,
                                            create external table et warehouse
 web country
                           string,
 web gmt offset
decimal(5,2),
                                              w warehouse sk
                                                                       tinyint,
 web tax precentage
                                              w warehouse id
                                                                      string,
                                              w_warehouse_name
decimal(5,2)
                                                                       string,
                                              w_warehouse_sq_ft
                                                                      bigint,
row format delimited fields terminated
                                              w street number
                                                                       string,
by '|'
                                              w_street_name
                                                                       string,
                                              w street_type
location '/tpcds10000g/web site'
                                                                       string,
tblproperties
                                              w suite number
                                                                       string,
('serialization.null.format'='')
                                              w city
                                                                       string,
                                              w county
                                                                       string,
;
                                              w state
                                                                       string,
                                              w zip
                                                                       string,
create external table et reason
                                              w country
                                                                       string,
                                              w gmt offset
                                            decimal(5,2)
 r reason sk
                           tinyint,
 r reason id
                          string,
 r reason desc
                           string
```

```
row format delimited fields terminated
by '|'
location '/tpcds10000g/warehouse'
tblproperties
('serialization.null.format'='')
;
```

D.3 Hive 0.13

```
-- Use the following to execute this
script and create the tables in Hive:
-- $HIVE HOME/bin/hive -hiveconf
DB NAME=300 -f
$testhome/ddl/065.hive.create-
tables.ddl
CREATE DATABASE IF NOT EXISTS
TPCDS${hiveconf:DB NAME}G HIVE ORC B N
 COMMENT 'For TPCDS tables at
${hiveconf:DB NAME} scale factor';
TPCDS${hiveconf:DB NAME}G HIVE ORC B N
create external table customer address
    ca address sk
   ca address id
                              string,
    ca_street_number
                             string,
    ca_street_name
                              string,
    ca_street_type
                              string,
    ca suite number
                              string,
    ca city
                              string,
    ca county
                              string,
    ca state
                              string,
    ca zip
                              string,
    ca country
                              string,
    ca gmt offset
                              double,
    ca_location_type
                             string
row format delimited
fields terminated by '|'
stored as ORC
tblproperties
('serialization.null.format'='')
create external table
customer demographics
    cd demo sk
                              int.
    cd_gender
                              string,
    cd_marital_status
                              string,
                              string,
    cd education status
    cd purchase estimate
                              int,
    cd credit rating
                              string,
```

```
cd dep count
                               int,
    cd_dep_employed_count
                               int,
    cd dep college count
                               int
)
row format delimited
fields terminated by '|'
stored as ORC
tblproperties
('serialization.null.format'='')
create external table date dim
    d date sk
                               int,
    d date id
                               string,
    d_date
                              string,
    {\tt d\_month\_seq}
                              int,
    d week seq
                               int,
    d quarter seq
                              int,
   d year
                              int,
   d dow
                              int,
    d moy
                              int,
    d_dom
                              int,
    d qoy
                              int,
    d fy year
                              int,
   d_fy_quarter_seq
                              int,
   d fy week seq
                              int,
    d day name
                              string,
    d quarter name
                              string,
    d holiday
                              string,
    d weekend
                              string,
   d_following_holiday
                              string,
    d first dom
                              int,
    d_last_dom
                              int.
    d_same_day_ly
                              int,
    d_same_day_lq
                               int,
    d current_day
                              string,
    d current week
                              string,
    d current month
                              string,
    d_current_quarter
                              string,
    d current year
                              string
row format delimited
fields terminated by '|'
stored as ORC
tblproperties
('serialization.null.format'='')
```

```
create external table warehouse
    w warehouse sk
    w warehouse id
                              string,
    w warehouse name
                              string,
    w_warehouse_sq_ft
                              int,
    w_street_number
                              string,
    w street name
                              string,
                             string,
    w street type
    w suite number
                             string,
   w city
                             string.
    w county
                              string,
    w state
                              string,
    w zip
                              string,
                              string,
    w country
    w_gmt_offset
                              double
)
row format delimited
fields terminated by '|'
stored as ORC
tblproperties
('serialization.null.format'='')
create external table ship mode
    sm ship mode sk
                              int,
    sm_ship_mode_id
                              string,
    sm_type
                              string,
    {\it sm}\ {\it code}
                              string,
    sm carrier
                              string,
    sm contract
                              string
row format delimited
fields terminated by '|'
stored as ORC
tblproperties
('serialization.null.format'='')
;
create external table time dim
    t time sk
                              int,
    t time id
                              string,
    t time
                              int,
    t hour
                              int.
    t minute
                              int,
    t second
                              int,
    t am pm
                              string,
   t shift
                              string,
   t_sub_shift
                              string,
    t meal time
                              string
row format delimited
fields terminated by '|'
stored as ORC
tblproperties
('serialization.null.format'='')
```

```
create external table reason
   r reason sk
   r reason id
                             string,
   r reason desc
                             string
row format delimited
fields terminated by '|'
stored as ORC
tblproperties
('serialization.null.format'='')
create external table income band
   ib_income_band_sk
                             int.
   ib_lower_bound
                             int,
   ib_upper_bound
                             int
row format delimited
fields terminated by '|'
stored as ORC
tblproperties
('serialization.null.format'='')
create external table item
   i_item_sk
i_item_id
   i item sk
                             int,
                             string,
   i rec start date
                             string,
   i rec end date
                            string,
   i_item_desc
                            string,
   i_current_price
                             double,
   i wholesale cost
                             double,
   i brand id
                             int,
   i brand
                             string,
   i class id
                             int,
   i_class
                             string,
                            int,
   i category id
   i category
                            string,
   i manufact id
   i manufact
                            string,
   i size
                            string,
   i_{	extsf{formulation}}
                             string,
   i color
                             string,
   i units
                             string,
   i container
                             string,
   i manager id
                             int,
   i product name
                             string
row format delimited
fields terminated by '|'
stored as ORC
tblproperties
('serialization.null.format'='')
```

```
cc_market_manager
                                                                     string,
                                              cc division
                                                                      int,
                                              cc division name
                                                                      string,
create external table store
                                              cc company
                                                                      int.
                                              cc_company_name
                                                                      string,
   s store sk
                                             cc_street_number
                                                                     string,
                           int,
   s_store_id
                           string,
                                             cc_street_name
                                                                     string,
   s rec start date
                           string,
                                             cc street type
                                                                     string,
   s rec end date
                           string,
                                             cc suite number
   s closed date sk
                          int,
                                             cc city
                                                                     string,
                                             cc county
   s store name
                           string,
                                                                     string.
   s number employees
                           int,
                                             cc state
                                                                     string,
   s floor space
                           int,
                                             cc zip
                                                                      string,
   s hours
                            string,
                                              cc country
                                                                      string,
                            string,
                                              cc gmt offset
                                                                      double,
   s manager
   s_market_id
                            int,
                                              cc_tax_percentage
                                                                      double
                          string,
   s_geography_class
   s_market_desc
                           string,
                                          row format delimited
                                          fields terminated by '|'
   s market manager
                          string,
   s division id
                           int,
                                          stored as ORC
   s division name
                                          tblproperties
                          string,
                                          ('serialization.null.format'='')
   s company id
                           int,
   s company name
                           string,
   s_street_number
                           strina.
                                          create external table customer
   s street name
                            string,
   s street type
                            string,
                                             c customer_sk
   s suite number
                            string,
                                                                      int,
                                             c_customer_id
                                                                     string,
   s city
                           string,
                                             c_current_cdemo_sk
                                                                      int,
   s_county
                           string,
                                             c_current_hdemo_sk
                                                                      int,
   s state
                           string,
                           string,
                                             c current addr sk
                                                                      int,
   s zip
                                             c first shipto date sk
                                                                      int,
   s country
                           string,
   s gmt offset
                           double,
                                             c first sales date sk
                                                                      int,
                           double
                                             c salutation
   s_tax_precentage
                                                                      string,
                                             c_first_name
                                                                      string,
row format delimited
                                             c last name
                                                                      string,
fields terminated by '|'
                                             c_preferred_cust_flag string,
stored as ORC
                                             c birth day
                                                                      int,
tblproperties
                                             c birth month
                                                                      int,
('serialization.null.format'='')
                                             c_birth_year
                                                                      int,
;
                                             c birth country
                                                                      string,
                                             c login
                                                                      string,
create external table call center
                                              c email address
                                                                      string,
                                              c_last_review_date
                                                                     string
   cc_call_center_sk
                            int,
                                          row format delimited
   cc_call_center_id
                           string,
                                          fields terminated by ' \mid '
   cc rec start date
                            string,
                                          stored as ORC
   cc_rec_end_date
                            string,
   cc closed date sk
                                          tblproperties
                            int,
                                          ('serialization.null.format'='')
   cc open date sk
                            int,
   cc_name
                            string,
   cc class
                           string,
                                          create external table web site
   cc employees
                           int,
   cc sq ft
                           int,
                                             web site sk
   cc hours
                           string,
                                                                      int,
                                             web_site id
   cc manager
                           string,
                                                                      string,
   cc_mkt_id
                           int,
                                             web_rec_start_date
                                                                      string,
   cc mkt class
                                             web rec end date
                           string,
                                                                      string,
   cc mkt desc
                           string,
                                             web name
                                                                      string,
```

```
int,
                                           tblproperties
    web_open_date_sk
    web_close_date_sk
                            int,
                                            ('serialization.null.format'='')
    web class
                             string,
    web manager
                             string,
                                           create external table
    web mkt id
                            int,
    web_mkt_class
                            string,
                                           household demographics
    web_mkt_desc
                            string,
    web market manager
                                               hd demo sk
                                                                        int,
                           string,
    web company id
                                               hd income band sk
                                                                        int,
                            int,
                                               hd_buy_potential
    web company name
                           string,
                                                                        string,
    web street number
                           string,
                                               hd dep count
                                                                        int,
    web street name
                            string,
                                               hd vehicle count
                                                                        int
    web_street_type
                            string,
                                            row format delimited
    web suite number
                           string,
                                           fields terminated by '|'
    web city
                            string,
                                           stored as ORC
    web county
                            string,
                                            tblproperties
   web_state
                            string,
                                            ('serialization.null.format'='')
    web_zip
                            string,
    web country
                            string,
    web_gmt_offset
                            double,
                            double
                                           create external table web page
    web_tax_percentage
row format delimited
                                               wp web page sk
                                                                        int,
fields terminated by '|'
                                               wp web page id
                                                                        string,
stored as ORC
                                               wp rec start date
                                                                        string,
tblproperties
                                               wp rec end date
                                                                        string,
('serialization.null.format'='')
                                               wp creation date sk
                                                                        int,
                                               wp access date sk
                                                                        int.
                                                                        string,
                                               wp_autogen_flag
                                               wp_customer_sk
create external table store returns
                                                                        int,
                                               wp url
                                                                        string,
    sr returned date sk
                                               wp type
                                                                        string,
   sr return time sk
                            int,
                                               wp char count
                                                                        int,
    sr_item_sk
                            int,
                                               wp_link_count
                                                                        int,
    sr_customer_sk
                           int,
                                               wp_image_count
                                                                        int,
    sr cdemo sk
                            int,
                                               wp max ad count
                                                                        int
                            int,
    sr hdemo sk
                                           row format delimited
    sr addr sk
                            int,
                                           fields terminated by '|'
    sr store sk
                            int,
                                           stored as ORC
    sr reason sk
                            int,
    sr ticket number
                            int,
                                           tblproperties
    sr return quantity
                            int,
                                            ('serialization.null.format'='')
    sr_return_amt
sr_return_tax
                            double,
                            double,
                                           create external table promotion
    sr_return_amt_inc_tax
                            double.
    sr fee
                             double,
    sr_return_ship_cost
                             double,
                                               p promo sk
                                                                         int,
    sr_refunded_cash
                             double,
                                               p_promo_id
                                                                        string,
    sr reversed charge
                             double,
                                               p start_date_sk
                                                                        int,
                                               p_end_date sk
    sr store credit
                             double,
                                                                        int,
                                               p_item_sk
    sr net loss
                             double
                                                                        int,
                                               p cost
                                                                        double,
--row format delimited
                                               p response target
                                                                        int,
--fields terminated by '|'
                                               p promo name
                                                                        string,
clustered by (sr item sk)
                                               p channel dmail
                                                                        string,
sorted by (sr ticket number,
                                               p channel email
                                                                        string,
sr_item_sk) into 271 buckets
                                               p_channel_catalog
                                                                        string,
stored as ORC
                                               p channel tv
                                                                        string,
                                               p channel radio
                                                                        string,
```

```
p_channel_press
                             string,
                                                 cr_returning_customer_sk int,
    p_channel_event
                             string,
                                                 cr_returning_cdemo_sk
                                                                           int,
    p channel demo
                                                 cr returning hdemo sk
                             string,
                                                                           int,
    p channel details
                                                 cr returning addr sk
                             string,
                                                                           int.
    p_purpose
                             string,
                                                 cr_call_center_sk
                                                                           int,
    p_discount_active
                                                 cr_catalog_page_sk
                                                                           int,
                             string
                                                 cr_ship_mode_sk
                                                                           int,
row format delimited
                                                 cr warehouse sk
                                                                           int,
fields terminated by '|'
                                                 cr reason sk
                                                                           int,
stored as ORC
                                                cr order number
                                                                           int,
tblproperties
                                                cr return quantity
                                                                           int.
('serialization.null.format'='')
                                                cr return amount
                                                                           double,
                                                cr return tax
                                                                           double,
                                                 cr return amt inc tax
                                                                           double,
create external table catalog page
                                                 cr fee
                                                                           double,
                                                cr_return_ship_cost
                                                                           double,
                                                cr_refunded_cash
    cp_catalog_page_sk
                             int,
                                                                           double,
    cp_catalog_page_id
                             string,
                                                cr_reversed_charge
                                                                           double,
    cp start date sk
                             int,
                                                 cr store credit
                                                                           double,
    cp end date sk
                                                 cr net loss
                                                                           double
    cp department
                             string,
    cp catalog number
                                             --row format delimited
                             int,
                                             --fields terminated by '|'
    cp catalog page number
                            int,
                                             clustered by (cr item sk)
    cp description
                             string,
                                             sorted by (cr order number,
    cp type
                             string
                                             cr item sk) into 271 buckets
row format delimited
                                             stored as ORC
fields terminated by '|'
                                             tblproperties
stored as ORC
                                             ('serialization.null.format'='')
tblproperties
('serialization.null.format'='')
                                             create external table web returns
create external table inventory
                                                 wr_returned_date_sk
                                                                           int.
                                                wr_returned_time_sk
                                                                           int,
    inv date sk
                             int,
                                                wr item sk
                                                                           int,
    inv item sk
                             int,
                                                 wr refunded customer sk
                                                                           int,
                                                wr refunded cdemo sk
    inv warehouse sk
                             int,
                                                                           int,
    inv_quantity_on_hand
                             bigint
                                                wr refunded hdemo sk
                                                                           int.
                                                wr refunded addr sk
                                                                           int,
--row format delimited
                                                 wr returning customer sk int,
--fields terminated by '|'
                                                 wr returning cdemo sk
                                                                           int,
clustered by (inv item sk)
                                                wr returning hdemo sk
                                                                           int.
sorted by (inv date sk, inv item sk,
                                                wr returning addr sk
                                                                           int,
inv_warehouse_sk) into 89 buckets
                                                 wr_web_page_sk
                                                                           int,
stored as ORC
                                                 wr_reason_sk
                                                                           int.
tblproperties
                                                 wr_order_number
                                                                           int,
('serialization.null.format'='')
                                                 wr_return_quantity
                                                                           int,
                                                 wr return amt
                                                                           double,
                                                 wr return_tax
                                                                           double,
create external table catalog returns
                                                 wr_return_amt_inc_tax
                                                                           double,
                                                 wr fee
                                                                           double,
    cr returned date sk
                             int,
                                                 wr return ship cost
                                                                           double,
    cr returned time sk
                             int,
                                                wr refunded cash
                                                                           double,
    cr item sk
                             int,
                                                wr reversed charge
                                                                           double,
    cr refunded customer sk int,
                                                 wr account credit
                                                                           double,
    cr_refunded_cdemo_sk
                                                 wr_net_loss
                                                                           double
                             int,
    cr refunded hdemo sk
                             int,
    cr refunded addr sk
                                             --row format delimited
                             int,
```

```
--fields terminated by '|'
                                              cs_sold_time_sk
                                                                       int,
clustered by (wr item sk)
                                              cs_ship_date_sk
                                                                       int,
sorted by (wr order number,
                                              cs bill customer sk
                                                                       int,
wr item sk) into 271 buckets
                                              cs_bill_cdemo_sk
                                                                       int.
stored as ORC
                                              cs_bill_hdemo_sk
                                                                       int,
tblproperties
                                              cs_bill_addr_sk
                                                                       int,
('serialization.null.format'='')
                                              cs_ship_customer_sk
                                                                       int,
                                              cs ship cdemo sk
                                                                       int,
                                              cs ship hdemo sk
                                                                       int,
create external table web sales
                                              cs ship addr sk
                                                                       int,
                                              cs call center sk
                                                                       int.
   ws sold date sk
                           int.
                                              cs_catalog_page_sk
                                                                       int.
   ws sold time sk
                           int,
                                              cs ship mode sk
                                                                       int,
   ws ship date sk
                           int,
                                              cs warehouse sk
                                                                       int,
   ws item sk
                                              cs item sk
                                                                       int.
                                              cs_promo sk
   ws_bill_customer_sk
                           int,
                                                                       int,
   ws_bill_cdemo_sk
                           int,
                                              cs_order_number
                                                                      int,
   ws_bill_hdemo_sk
                           int,
                                              cs_quantity
                                                                       int,
   ws_bill_addr_sk
                           int,
                                              cs wholesale cost
                                                                      double,
                                             cs_list_price
   ws ship customer sk
                                                                       double,
                           int,
                                             cs_sales price
   ws ship cdemo sk
                           int,
                                                                      double,
   ws ship hdemo sk
                           int,
                                             cs ext discount amt
                                                                      double,
                           int,
   ws ship addr sk
                                              cs ext sales price
                                                                      double,
                           int,
                                              cs_ext_wholesale_cost
   ws web page sk
                                                                      double,
                           int,
                                              cs ext list price
   ws web site sk
                                                                       double,
   ws ship mode sk
                            int,
                                              cs ext tax
                                                                       double,
   ws warehouse sk
                           int,
                                              cs_coupon_amt
                                                                       double,
   ws promo sk
                           int,
                                                                       double,
                                              cs ext ship cost
   ws_order_number
                                              cs_net_paid
                                                                       double,
                           int,
                                             cs_net_paid_inc_tax
                                                                       double,
   ws quantity
                           int,
                                              cs_net_paid_inc_ship
   ws wholesale cost
                           double,
                                                                       double,
   ws list price
                           double,
                                              cs_net_paid_inc_ship_tax double,
   ws_sales_price
                            double,
                                              cs_net_profit
                                                                       double
   ws_ext_discount_amt
                           double,
   ws_ext_sales_price
                            double,
                                           --row format delimited
                                           --fields terminated by '|'
   ws ext wholesale cost
                            double,
   ws ext list price
                            double,
                                           clustered by (cs item sk)
                                           sorted by (cs order number,
   ws ext tax
                            double,
   ws coupon amt
                            double,
                                          cs item sk) into 271 buckets
                           double,
                                           stored as ORC
   ws_ext_ship_cost
   ws net paid
                            double,
                                           tblproperties
   ws net paid inc tax
                            double,
                                           ('serialization.null.format'='')
   ws net paid inc ship
                            double,
   ws_net_paid_inc_ship_tax double,
   ws_net_profit
                            double
                                          create external table store_sales
--row format delimited
                                              ss sold date sk
                                                                       int,
--fields terminated by '|'
                                              ss_sold_time_sk
                                                                       int,
clustered by (ws_item_sk)
                                              ss item sk
                                                                       int,
sorted by (ws order number,
                                              ss customer sk
                                                                       int,
ws item sk) into 271 buckets
                                              ss cdemo sk
                                                                       int,
stored as ORC
                                              ss hdemo sk
                                                                       int,
tblproperties
                                              ss addr sk
                                                                       int,
('serialization.null.format'='')
                                              ss store sk
                                                                       int,
                                              ss promo sk
                                                                       int,
                                              ss ticket number
                                                                       int,
create external table catalog_sales
                                              ss_quantity
                                                                       int,
                                              ss wholesale cost
                                                                       double,
   cs_sold_date_sk
                           int,
                                              ss list price
                                                                       double,
```

```
double,
ss_sales_price
ss ext discount amt
                        double,
                                       --row format delimited
ss ext sales price
                        double,
                                       --fields terminated by '|'
ss ext_wholesale_cost
                        double,
                                       clustered by (ss item sk)
ss_ext_list_price
                        double,
                                       sorted by (ss_ticket_number,
ss ext tax
                        double,
                                       ss item sk) into 271 buckets
ss_coupon_amt
                        double,
                                       stored as ORC
                        double,
ss net paid
                                       tblproperties
ss net paid inc tax
                       double,
                                       ('serialization.null.format'='')
                        double
ss net profit
```

Appendix E: Query Text

Queries for all vendors are generated from query templates. Specific parameter values depend on both the context the query is run (scale factor, single or multi-stream), and the seed for the random number generator. A common seed (20140815) for the random number generator was used across the 3 distributions, thus making all queries across all distributions the same.

The queries were executed in a different order for each stream based on the standard TPC-DS specification. However, the query order for each stream was identical across all vendors.

Following are the query text for the 46 common queries, as used during the single-stream test:

E.1 Big SQL Queries:

```
-- start query 1 in stream 0 using template
                                                          cd marital status = 'W' and
                                                          cd_education_status = '2 yr Degree'
query96.tpl and seed 550831069
select count(*)
                                                   and
from store sales
                                                          (p channel email = 'N' or
                                                   p channel event = 'N') and
   ,household_demographics
                                                         d_year = 1999
    ,time dim, store
where ss sold time sk = time dim.t time sk
                                                   group by i item id
   and ss hdemo sk =
                                                   order by i item id
household demographics.hd demo sk
                                                    fetch first 100 rows only;
    and ss store sk = s store sk
    and time \dim.t hour = 15
                                                   -- end query 2 in stream 0 using template
    and time dim.t minute >= 30
                                                   query7.tpl
    and household demographics.hd_dep_count
                                                   -- start query 5 in stream 0 using template
   and store.s_store_name = 'ese'
                                                   query39.tpl and seed 1420791654
                                                   with inv as
order by count(*)
fetch first 100 rows only;
                                                   (select
                                                   w warehouse name, w warehouse sk,i item sk,d
-- end query 1 in stream 0 using template
                                                          ,stdev,mean, case mean when 0 then
query96.tpl
-- start query 2 in stream 0 using template
                                                  null else stdev/mean end cov
query7.tpl and seed 997258328
                                                   from(select
                                                   w_warehouse_name,w_warehouse_sk,i_item_sk,d
select i_item_id,
        avg(cast(ss quantity as double))
agg1,
        avg(ss_list_price) agg2,
                                                   ,stddev_samp(inv_quantity_on_hand)
        avg(ss coupon amt) agg3,
                                                   stdev,avg(cast(inv quantity on hand as
        avg(ss sales price) agg4
                                                   double)) mean
from store sales, customer demographics,
                                                        from inventory
date dim, item, promotion
                                                            ,item
 where ss sold date sk = d date sk and
                                                             ,warehouse
       ss item sk = i item sk and
                                                            ,date dim
       ss cdemo sk = cd demo sk and
                                                         where inv item sk = i item sk
       ss promo sk = p_promo_sk and
                                                          and inv warehouse sk =
       cd gender = 'M' and
                                                   w warehouse sk
```

```
and inv date sk = d date sk
                                                           ,inv2.d moy,inv2.mean, inv2.cov
        and d year =2000
      group by
w warehouse name, w warehouse sk,i item sk,d
                                                   -- end query 5 in stream 0 using template
_moy) foo
                                                   query39.tpl
where case mean when 0 then 0 else
stdev/mean end > 1)
                                                   -- start query 7 in stream 0 using template
                                                   query32.tpl and seed 944563352
inv1.w warehouse sk,inv1.i item sk,inv1.d m
                                                   select sum(cs ext discount amt) as
oy, inv1.mean, inv1.cov
                                                   "excess discount amount"
                                                   from
,inv2.w warehouse sk,inv2.i item sk,inv2.d
                                                      catalog sales
                                                      ,item
moy, inv2.mean, inv2.cov
from inv inv1, inv inv2
                                                      ,date_dim
where inv1.i item sk = inv2.i item sk
                                                   where
 and inv1.w warehouse sk =
                                                   i manufact id = 353
inv2.w warehouse sk
                                                   and i_item_sk = cs_item_sk
                                                   and d_{date} between '2000-01-16' and
 and inv1.d moy=2
 and inv2.d moy=2+1
                                                            (cast('2000-01-16' as date) + 90
order by
                                                   days)
inv1.w warehouse sk,inv1.i item sk,inv1.d m
                                                   and d date sk = cs sold date sk
oy, inv1.mean, inv1.cov
                                                   and cs ext discount amt
        ,inv2.d moy,inv2.mean, inv2.cov
                                                        > (
                                                            select
                                                               1.3 * avg(cs_ext_discount_amt)
with inv as
(select
w warehouse name, w warehouse sk,i item sk,d
                                                               catalog sales
                                                              ,date dim
       ,stdev,mean, case mean when 0 then
                                                            where
null else stdev/mean end cov
                                                                 cs item sk = i item sk
                                                             and d_date between '2000-01-16'
from(select
w_warehouse_name,w_warehouse_sk,i item sk,d
                                                   and
                                                                                 (cast('2000-
_moy
                                                   01-16' as date) + 90 days)
,stddev_samp(inv_quantity_on_hand)
                                                             and d_date_sk = cs_sold_date_sk
stdev, avg(cast(inv quantity on hand as
double)) mean
                                                    fetch first 100 rows only;
     from inventory
         ,item
                                                   -- end query 7 in stream 0 using template
          ,warehouse
                                                   query32.tpl
          ,date dim
      where inv item sk = i item sk
                                                   -- start query 14 in stream 0 using
       and inv warehouse sk =
                                                   template query21.tpl and seed 614834996
                                                   select *
w warehouse sk
       and inv date sk = d date sk
                                                    from(select w_warehouse_name
        and d_year =2000
                                                               ,i_item_id
                                                               ,sum(case when (cast(d date as
      group by
                                                   date) < cast ('1998-06-27' as date))
w warehouse name, w warehouse sk,i item sk,d
mov) foo
                                                                           t.hen
where case mean when 0 then 0 else
                                                   inv_quantity_on_hand
stdev/mean end > 1)
                                                                          else 0 end) as
                                                   inv before
inv1.w_warehouse_sk,inv1.i_item_sk,inv1.d_m
                                                               ,sum(case when (cast(d_date as
oy, inv1.mean, inv1.cov
                                                   date) >= cast ('1998-06-27' as date))
                                                                          then
,inv2.w_warehouse_sk,inv2.i_item_sk,inv2.d_
                                                   inv_quantity_on_hand
moy, inv2.mean, inv2.cov
                                                                          else 0 end) as
from inv inv1, inv inv2
                                                   inv after
where inv1.i_item_sk = inv2.i_item_sk
                                                      from inventory
                                                          ,warehouse
 and inv1.w_warehouse_sk =
                                                          ,item
inv2.w warehouse sk
                                                          ,date_dim
 and inv1.d moy=2
 and inv2.d_moy=2+1
                                                      where i_current_price between 0.99 and
 and inv1.cov > 1.5
order by
                                                        and i item sk
                                                                               = inv item sk
inv1.w warehouse sk,inv1.i item sk,inv1.d m
                                                        and inv warehouse sk
oy, invl.mean, invl.cov
                                                   w warehouse sk
```

```
and inv date sk = d date sk
                                                         ss item sk = i item sk and
    and d date between (cast ('1998-06-27'
                                                         ss store sk = s store sk and
as date) - 30 days)
                                                          ss cdemo sk = cd demo sk and
                   and (cast ('1998-06-27'
                                                          cd gender = 'F' and
as date) + 30 days)
                                                          cd marital status = 'W' and
   group by w warehouse name, i item id) x
                                                          cd education status = '4 yr Degree'
where (case when inv before > 0
                                                   and
                                                          d year = 1999 and
           then cast(inv after as double)
/ cast(inv before as double)
                                                         s state in ('OH', 'IL', 'LA', 'GA',
                                                   'CO', 'AL')
            else null
            end) between 2.0/3.0 and
                                                    group by rollup (i_item_id, s_state)
3.0/2.0
                                                    order by i item id
order by w_warehouse_name
                                                           ,s state
       ,i_item_id
                                                     fetch first 100 rows only;
 fetch first 100 rows only;
                                                   -- end query 16 in stream 0 using template
-- end query 14 in stream 0 using template
                                                   query27.tpl
query21.tpl
-- start query 15 in stream 0 using
                                                   -- start query 19 in stream 0 using
template query43.tpl and seed 959608359
                                                   template query58.tpl and seed 1844319395
                                                   with ss items as
select s store name, s store id,
       sum(case when (d day name='Sunday')
                                                   (select i item id item id
then ss sales price else null end)
                                                         , sum (ss ext sales price)
                                                   ss item rev
                                                    from store_sales
      sum(case when (d_day_name='Monday')
                                                      ,item
then ss sales price else null end)
mon sales,
                                                       ,date dim
                                                    where ss item sk = i item sk
       sum(case when
(d day name='Tuesday') then ss sales price
                                                      and d date in (select d date
else null end) tue sales,
                                                                     from date dim
       sum(case when
                                                                     where d_week_seq =
(d_day_name='Wednesday') then
                                                   (select d week seq
ss sales price else null end) wed sales,
                                                                                         from
       sum(case when
                                                   date_dim
(d_day_name='Thursday') then ss_sales_price
                                                                                         where
else null end) thu sales,
                                                   d date = '1998-05-29'))
       sum(case when (d day name='Friday')
                                                    and ss sold date sk = d date sk
then ss_sales_price else null end)
                                                    group by i_item_id),
                                                    cs items as
fri_sales,
                                                    (select i_item_id item_id
       sum(case when
(d day name='Saturday') then ss sales price
                                                          , sum (cs ext sales price)
                                                   cs_item_rev
else null end) sat_sales
                                                     from catalog_sales
from date dim, store sales, store
                                                        ,item
where d date sk = ss sold date sk and
      s_store_sk = ss_store_sk and
                                                        ,date dim
      s_gmt_offset = -8 and
                                                    where cs_item_sk = i_item_sk
      d year = 1998
                                                    and d date in (select d date
group by s store name, s store id
                                                                     from date dim
order by s store name,
                                                                     where d week seq =
s store id, sun sales, mon sales, tue sales, we
                                                   (select d_week_seq
d sales, thu sales, fri sales, sat sales
                                                                                         from
 fetch first 100 rows only;
                                                   date dim
                                                                                         where
-- end query 15 in stream 0 using template
                                                   d date = '1998-05-29'))
query43.tpl
                                                    and cs_sold_date_sk = d_date_sk
                                                    group by i_item_id),
-- start query 16 in stream 0 using
template query27.tpl and seed 331218716
                                                    ws items as
select i item id,
                                                    (select i item id item id
       s_state, grouping(s_state) g_state,
                                                         , sum (ws_ext_sales_price)
       avg(cast(ss_quantity as double))
                                                   ws_item_rev
                                                     from web sales
agg1,
                                                        ,item
        avg(ss list price) agg2,
       avg(ss_coupon_amt) agg3,
                                                         ,date dim
       avg(ss sales price) agg4
                                                   where ws item sk = i item sk
from store sales, customer demographics,
                                                   and d date in (select d date
                                                                     from date dim
date dim, store, item
where ss\_sold\_date\_sk = d\_date\_sk and
```

where d_week_s	seq =(select	and	ss_sold_date_sk	=	
d_week_seq		d_date_	sk		
	from	and	d year	= 2	2001
date dim		and	d moy	= 6	j
-	where	and	ss addr sk	=	
d date = '1998-05-29'))		ca addr	<u> </u>		
and ws sold date sk = d date	a sk	and	ca gmt offset	= -	-7
group by i item id)	_31				,
			by i_manufact_id),		
select ss_items.item_id		cs as			
,ss_item_rev		select			
					,
,ss_item_rev/(ss_item_rev+cs_ite	em_rev+ws_it	_	<pre>fact_id,sum(cs_ext_sale</pre>	s_price	÷)
em_rev)/3 * 100 ss_dev		total_s	sales		
,cs_item_rev		from			
			catalog_sales,		
<pre>,cs_item_rev/(ss_item_rev+cs_ite</pre>	em_rev+ws_it		date_dim,		
em_rev)/3 * 100 cs_dev			customer_address,		
,ws item rev			item		
		where			
,ws item rev/(ss item rev+cs ite	em rev+ws it		i manufact id		in
em rev)/3 * 100 ws dev		(select			
S101//0 100uov			ufact id		
(se item revites item revitue ite	om ross)/3	from	.u1u00_1u		
,(ss_item_rev+cs_item_rev+ws_ite	EIII_IEV//J				
average		item		,	
from ss_items,cs_items,ws_items		_	_category in ('Books')		
where ss_items.item_id=cs_items	_	and	cs_item_sk	=	
and ss_items.item_id=ws_items	_	i_item_	sk		
and ss_item_rev between 0.9 *	*	and	cs_sold_date_sk	=	
cs_item_rev and 1.1 * cs_item_re	ev	d_date_	sk		
and ss_item_rev between 0.9 7	*	and	d_year	= 2	2001
ws_item_rev and 1.1 * ws_item_re	ev	and	d_moy	= 6	j
and cs item rev between 0.9 *	*	and	cs bill addr sk	=	
ss item rev and 1.1 * ss item re	ev	ca addr	ess sk		
and cs item rev between 0.9 *		and	_ ca gmt offset	= -	-7
ws item rev and 1.1 * ws item re			by i manufact id),		
and ws item rev between 0.9		ws as			
ss item rev and 1.1 * ss item re		select			
		361600	•		
and ws_item_rev between 0.9 *					
cs_item_rev and 1.1 * cs_item_re	€V		fact_id,sum(ws_ext_sale	s_price	1)
order by item_id		total_s	sales		
,ss_item_rev		from			
fetch first 100 rows only;			web_sales,		
			date_dim,		
end query 19 in stream 0 usir	ng template		customer_address,		
query58.tpl			item		
		where			
start query 22 in stream 0 us	sing		i manufact id		in
template query33.tpl and seed 24	18487088	(select	 ;		
with ss as (ufact id		
select		from	.u_u_u_		
Select		item			
			(ID-1-1-1)	,	
i_manufact_id,sum(ss_ext_sales_r	price)		_category in ('Books')		
total_sales		and	ws_item_sk	=	
from		i_item_	sk		
store_sales,		and	ws_sold_date_sk	=	
date_dim,		d_date_	sk		
customer_address,		and	d_year	= 2	2001
item		and	d moy	= 6	j
where		and	ws bill addr sk	=	
i manufact id in (selec	ct	ca addr			
i manufact id		and	ca gmt offset	= -	-7
			by i manufact id)		,
from				(+o+o1 -	10100
item			t i_manufact_id ,sum	,cotal_S	ales.
where i_category in ('Books'))		total_s			
and ss_item_sk	=	from	(select * from ss		
i_item_sk			union all		
			select * from cs		

```
union all
                                                                 ,avg(sum(ss sales price)) over
        select * from ws) tmp1
                                                    (partition by i manager id)
 group by i manufact id
                                                    avg monthly sales
order by total sales
                                                          from item
 fetch first 100 rows only;
                                                              ,store sales
                                                              ,date dim
-- end query 22 in stream 0 using template
                                                              ,store
query33.tpl
                                                          where ss item sk = i item sk
                                                            and ss sold date sk = d date sk
-- start query 24 in stream 0 using
                                                            and ss store sk = s store sk
template query62.tpl and seed 800775315
                                                            and d month_seq in
                                                    (1178, 1178+1, 1178+2, 1178+3, 1178+4, 1178+5, 11
  substr(w_warehouse_name,1,20)
                                                    78+6,1178+7,1178+8,1178+9,1178+10,1178+11)
                                                           and (( i_category in
 ,sm_type
 ,web name
                                                    ('Books','Children','Electronics')
 ,sum(case when (ws ship date sk -
                                                                  and i class in
ws_sold_date_sk \le 30 ) then 1 else 0 end)
                                                    ('personal', 'portable', 'refernece', 'self-
as "30 days"
                                                    help')
 ,sum(case when (ws ship date sk -
                                                                  and i brand in
ws sold date sk > 30) and
                                                    ('scholaramalgamalg #14','scholaramalgamalg
                  (ws ship date sk -
ws sold date sk \le 60) then 1 else 0 end )
as "31-60 days"
                                                    'exportiunivamalg #9','scholaramalgamalg
 , sum (case when (ws ship date sk -
                                                    #9'))
ws_sold_date_sk > 60) and
                                                                    i category in
                                                               or(
                 (ws ship date sk -
                                                    ('Women','Music','Men')
ws sold date sk <= 90) then 1 else 0 end)
                                                                 and i class in
as "61-90 days"
                                                    ('accessories','classical','fragrances','pa
 , sum (case when (ws ship date sk -
                                                    nts')
ws_sold_date_sk > 90) and
                                                                  and i brand in ('amalgimporto
                                                    #1','edu packscholar #1','exportiimporto
                (ws_ship_date_sk -
ws_sold_date_sk <= 120) then 1 else 0 end)
                                                    #1',
as "91-120 days"
 ,sum(case when (ws_ship_date_sk -
                                                    'importoamalg #1')))
ws_sold_date_sk > 120) then 1 else 0 end)
                                                    group by i_manager_id, d_moy) tmp1
as ">120 days"
                                                    where case when avg monthly sales > 0 then
                                                    abs (sum sales - avg monthly sales) /
  web sales
                                                    avg monthly sales else null end > 0.1
 ,warehouse
                                                    order by i_manager_id
 ,ship mode
                                                            ,avg_monthly_sales
 ,web site
                                                            , sum sales
                                                     fetch first 100 rows only;
  ,date_dim
where
   d month seq between 1201 and 1201 + 11
                                                    -- end query 27 in stream 0 using template
and ws_ship_date_sk = d_date_sk
                                                    query63.tpl
and ws_warehouse_sk = w_warehouse_sk
and ws_ship_mode_sk = sm_ship_mode_sk
                                                    -- start query 28 in stream 0 using
                                                    template query69.tpl and seed 1390437346
and ws web site sk
                      = web site sk
                                                    select
                                                      cd gender,
group by
                                                      cd marital status,
  substr(w warehouse name, 1, 20)
                                                      cd education status,
 ,sm type
 ,web name
                                                      count(*) cnt1,
order by substr(w_warehouse_name,1,20)
                                                      cd purchase estimate,
       ,sm type
                                                      count(*) cnt2,
       ,web name
                                                      cd_credit_rating,
fetch first 100 rows only;
                                                      count(*) cnt3
                                                     from
-- end query 24 in stream 0 using template
                                                     customer c, customer address
query62.tpl
                                                    ca, customer demographics
                                                    where
-- start query 27 in stream 0 using
                                                     c.c current addr sk = ca.ca address sk
template query63.tpl and seed 812633773
select *
                                                      ca state in ('AL','VA','GA') and
from (select i manager id
                                                     cd demo sk = c.c current cdemo sk and
             ,sum(ss sales price) sum sales
                                                      exists (select *
                                                              from store sales, date dim
```

```
where c.c customer sk =
                                                        i item id, sum(cs ext sales price)
ss customer sk and
                                              total sales
              ss sold date sk = d date sk
                                               from
and
                                                      catalog_sales,
              d year = 2004 and
                                                      date dim,
              d moy between 3 and 3+2)
                                                       customer address,
and
                                                       item
  (not exists (select *
                                                where
          from web sales, date dim
                                                       i item id
                                                                      in (select
          where c.c_customer_sk =
                                                i item id
ws_bill_customer_sk and
                                               from
                ws sold date sk =
                                                item
d date sk and
                                               where i_category in ('Jewelry'))
                d year = 2004 and
                                               and cs_item_sk
                d moy between 3 and 3+2)
                                              i_item_sk
                                               and cs sold date sk
and
   not exists (select ^{\star}
                                               d_date_sk
                                              and d_year
                                                                              = 2002
         from catalog_sales,date_dim
                                                      d_moy
          where c.c_customer_sk =
                                               and
                                               and
cs ship customer sk and
                                                       cs bill addr sk
                                               ca address sk
                cs sold date sk =
d date sk and
                                               and ca gmt offset
                d year = 2004 and
                                               group by i item id),
                d moy between 3 and 3+2))
                                              ws as (
group by cd gender,
                                               select
         cd marital status,
                                                        i item id, sum(ws ext sales price)
         cd education status,
                                               total sales
         cd purchase estimate,
                                               from
         cd credit rating
                                                      web sales,
order by cd gender,
                                                      date dim,
         cd_marital_status,
                                                       customer_address,
         cd education status,
                                                       item
         cd purchase estimate,
                                                where
         cd credit rating
                                                       i item id
                                                                    in (select
 fetch first 100 rows only;
                                                i_item_id
                                               from
-- end query 28 in stream 0 using template
                                                item
                                               where i_category in ('Jewelry'))
query69.tpl
                                                       ws_item_sk
-- start query 29 in stream 0 using
                                               and
template query60.tpl and seed 374071684
                                               i item sk
with ss as (
                                                and
                                                       ws sold date sk
                                               d_date_sk
select
         i item id, sum(ss ext sales price)
                                               and d_year
total sales
                                                and d moy
                                                                              = 10
from
                                                and
                                                      ws_bill_addr_sk
      store sales,
                                               ca address sk
      date dim,
                                               and ca gmt offset
       customer address,
                                               group by i item id)
                                                select
                                                i item id
where
       i item id in (select
                                               ,sum(total sales) total sales
                                                from (select * from ss
 i item id
from
                                                      union all
                                                      select * from cs
                                                      union all
where i_category in ('Jewelry'))
                                                      select * from ws) tmp1
and ss_item_sk
i_item_sk
                                                group by i item id
and ss_sold_date sk
                                                order by i item id
d date sk
                                                    , total_sales
and d_year
                              = 2002
                                                fetch first 100 rows only;
      d_moy
ss_addr_sk
and
                              = 10
and
                                               -- end query 29 in stream 0 using template
ca_address_sk
                                               query60.tpl
and ca gmt offset
                          = -5
                                               -- start query 30 in stream 0 using
group by i item id),
                                               template query59.tpl and seed 1976435349
cs as (
                                               with wss as
select
                                                (select d_week_seq,
```

```
d month seg between 1189+ 12 and
       ss store sk.
       sum(case when (d day name='Sunday')
                                                   1189 + 23) x
then ss sales price else null end)
                                                    where s store id1=s store id2
sun_sales,
                                                      and d_week_seq1=d_week_seq2-52
       sum(case when (d day name='Monday')
                                                    order by
then ss sales price else null end)
                                                   s store name1,s store id1,d week seq1
                                                    fetch first 100 rows only;
mon sales,
       sum(case when
(d day name='Tuesday') then ss sales price
                                                   -- end query 30 in stream 0 using template
else null end) tue sales,
                                                   query59.tpl
       sum(case when
(d day name='Wednesday') then
                                                   -- start query 32 in stream 0 using
ss_sales_price else null end) wed_sales,
                                                   template query98.tpl and seed 1900673199
                                                   select i_item_desc
       sum(case when
(d day name='Thursday') then ss_sales_price
                                                         ,i_category
                                                         ,i_class
else null end) thu sales,
       sum(case when (d day name='Friday')
                                                         ,i current price
then ss_sales_price else null end)
                                                         , sum(ss\_ext\_sales\_price) as
                                                   itemrevenue
fri sales,
       sum(case when
(d day name='Saturday') then ss sales price
                                                   ,sum(ss ext sales price) *100/sum(sum(ss ext
                                                   _sales_price)) over
else null end) sat sales
from store sales, date dim
                                                             (partition by i class) as
where d_date_sk = ss_sold date sk
                                                   revenueratio
group by d_week_seq,ss_store_sk
                                                   from
                                                           store sales
 select
                                                           ,item
s store name1,s store id1,d week seq1
                                                           ,date dim
                                                   where
                                                           ss item sk = i item sk
,sun sales1/sun sales2,mon sales1/mon sales
                                                          and i_category in ('Shoes',
                                                    'Music', 'Home')
,tue sales1/tue sales1,wed sales1/wed sales
                                                           and ss sold date sk = d date sk
2,thu sales1/thu sales2
                                                           and d date between cast('1999-05-
                                                   25' as date)
,fri_sales1/fri_sales2,sat_sales1/sat_sales
                                                                                  and
2
                                                   (cast('1999-05-25' as date) + 30 days)
from
(select s_store_name
                                                          i item id
s_store_name1,wss.d_week_seq d_week_seq1
                                                           ,i_item_desc
       ,s store id s store id1, sun sales
                                                           ,i category
                                                           ,i class
sun sales1
       ,mon sales mon sales1, tue sales
                                                           ,i_current_price
tue_sales1
                                                   order by
       ,wed sales wed sales1,thu sales
                                                           i category
thu_sales1
                                                           ,i_class
       ,fri sales fri sales1,sat sales
                                                           ,i_item_id
                                                           ,i item desc
sat sales1
 from wss, store, date dim d
                                                           ,revenueratio;
  where d.d week seq = wss.d week seq and
       ss store sk = s store sk and
                                                   -- end query 32 in stream 0 using template
       d month seq between 1189 and 1189 +
                                                   query98.tpl
11) y,
                                                   -- start query 36 in stream 0 using
 (select s store name
s_store_name2,wss.d_week_seq d_week_seq2
                                                   template query28.tpl and seed 24799953
                                                   select *
        ,s_store_id s_store_id2,sun_sales
sun sales2
                                                   from (select avg(ss list price) B1 LP
       ,mon sales mon sales2, tue sales
                                                               ,count(ss list price) B1 CNT
tue sales2
                                                                ,count(distinct ss_list_price)
                                                   B1 CNTD
       ,wed sales wed sales2,thu sales
                                                         from store sales
thu sales2
       ,fri sales fri sales2,sat sales
                                                         where ss quantity between 0 and 5
                                                           and (ss_list_price between 47 and
sat sales2
 from wss, store, date dim d
  where d.d week seq = wss.d week seq and
                                                                or ss coupon amt between 11713
       ss store sk = s store sk and
                                                   and 11713+1000
```

```
or ss wholesale cost between
                                                  -- start query 39 in stream 0 using
55 and 55+20)) B1,
                                                  template query66.tpl and seed 1688498284
     (select avg(ss list price) B2 LP
                                                   select
           ,count(ss_list_price) B2_CNT
                                                            w warehouse name
            , count(distinct ss list price)
                                                          ,w warehouse sq ft
B2 CNTD
                                                          ,w city
     from store sales
                                                          ,w county
                                                          ,w_state
     where ss quantity between 6 and 10
       and (ss list price between 93 and
                                                          ,w country
                                                          ,ship carriers
         or ss_coupon amt between 7733 and
                                                          ,year
7733+1000
                                                          ,sum(jan_sales) as jan sales
         or ss_wholesale_cost between 43
                                                          ,sum(feb_sales) as feb sales
and 43+20)) B2,
                                                          ,sum(mar_sales) as mar sales
                                                          ,sum(apr_sales) as apr sales
     (select avg(ss_list_price) B3_LP
           ,count(ss list price) B3 CNT
                                                          ,sum(may sales) as may sales
           , count (distinct ss list price)
                                                          ,sum(jun_sales) as jun_sales
B3 CNTD
                                                          ,sum(jul_sales) as jul_sales
     from store_sales
                                                          ,sum(aug_sales) as aug_sales
     where ss quantity between 11 and 15
                                                          ,sum(sep sales) as sep sales
                                                          ,sum(oct sales) as oct sales
       and (ss list price between 32 and
                                                          ,sum(nov sales) as nov sales
         or ss coupon amt between 11517
                                                          ,sum(dec sales) as dec sales
and 11517+1000
                                                          ,sum(jan_sales/w_warehouse_sq_ft)
         or ss wholesale cost between 26
                                                   as jan_sales_per_sq_foot
and 26+20)) B3,
                                                          , sum (feb sales/w warehouse sq ft)
    (select avg(ss list price) B4 LP
                                                   as feb sales per sq foot
           ,count(ss list price) B4 CNT
                                                          ,sum(mar sales/w warehouse sq ft)
           ,count(distinct ss list price)
                                                   as mar sales per sq foot
B4 CNTD
                                                          ,sum(apr sales/w warehouse sq ft)
      from store_sales
                                                   as apr_sales_per_sq_foot
     where ss_quantity between 16 and 20
                                                         , sum (may_sales/w_warehouse_sq_ft)
        and (ss list price between 147 and
                                                   as may_sales_per_sq_foot
147+10
                                                          ,sum(jun_sales/w_warehouse_sq_ft)
         or ss_coupon_amt between 509 and
                                                   as jun_sales_per_sq_foot
509+1000
                                                          ,sum(jul_sales/w_warehouse_sq_ft)
         or ss wholesale cost between 78
                                                   as jul sales per sq foot
and 78+20)) B4,
                                                         ,sum(aug_sales/w_warehouse_sq_ft)
     (select avg(ss_list_price) B5_LP
                                                   as aug_sales_per_sq_foot
          ,count(ss_list_price) B5 CNT
                                                         , sum(sep_sales/w_warehouse_sq_ft)
           , count (distinct ss list price)
                                                   as sep sales per sq foot
B5 CNTD
                                                          ,sum(oct_sales/w_warehouse_sq_ft)
                                                   as oct_sales_per_sq_foot
     from store sales
     where ss quantity between 21 and 25
                                                         ,sum(nov sales/w warehouse sq ft)
                                                   as nov_sales_per_sq_foot
       and (ss list price between 16 and
16+10
                                                          ,sum(dec_sales/w_warehouse_sq_ft)
         or ss coupon amt between 2401 and
                                                   as dec sales per sq foot
2401+1000
                                                          ,sum(jan net) as jan net
         or ss wholesale cost between 32
                                                          ,sum(feb net) as feb net
                                                          , sum (mar net) as mar net
and 32+20)) B5,
     (select avg(ss list price) B6 LP
                                                          ,sum(apr net) as apr net
                                                          ,sum(may_net) as may_net
           ,count(ss list price) B6 CNT
           ,count(distinct ss_list_price)
                                                          ,sum(jun_net) as jun_net
B6 CNTD
                                                          ,sum(jul_net) as jul_net
     from store_sales
                                                          ,sum(aug_net) as aug_net
     where ss_quantity between 26 and 30
                                                          ,sum(sep_net) as sep_net
                                                          , sum(oct_net) as oct net
       and (ss_list_price between 11 and
                                                          ,sum(nov net) as nov net
         or ss coupon amt between 916 and
                                                          ,sum(dec_net) as dec_net
916+1000
                                                    from (
         or ss wholesale cost between 6
                                                      (select
and 6+20)) B6
                                                          w warehouse name
fetch first 100 rows only;
                                                          ,w_warehouse_sq_ft
                                                          ,w_city
-- end query 36 in stream 0 using template
                                                          ,w county
query28.tpl
                                                          ,w state
                                                          ,w_country
```

```
, sum(case when d moy = 8)
       ,'MSC' || ',' || 'USPS' as
ship carriers
                                                                 then
      ,d year as year
                                                  ws net paid inc ship tax * ws quantity else
       , sum (case when d moy = 1
                                                  0 end) as aug_net
              then ws sales price*
                                                          , sum(case when d moy = 9)
ws quantity else 0 end) as jan sales
       , sum (case when d moy = 2
                                                  ws net paid inc ship tax * ws quantity else
             then ws sales price*
                                                  0 end) as sep net
ws quantity else 0 end) as feb sales
                                                         , sum(case when d moy = 10
       , sum(case when d moy = 3)
                                                                 then
             then ws_sales_price*
                                                  ws_net_paid_inc_ship_tax * ws_quantity else
ws quantity else 0 end) as mar sales
                                                  0 end) as oct net
       , sum(case when d_moy = 4
                                                         , sum(case when d moy = 11
           then ws_sales_price*
                                                                 then
ws quantity else 0 end) as apr_sales
                                                  ws_net_paid_inc_ship_tax * ws_quantity else
       , sum(case when d moy = 5
                                                  0 end) as nov net
                                                         ,sum(case when d_moy = 12
             then ws_sales_price*
ws_quantity else 0 end) as may_sales
                                                               then
                                                  ws_net_paid_inc_ship_tax * ws_quantity else
       , sum(case when d moy = 6)
              then ws_sales_price*
                                                  0 end) as dec net
ws quantity else 0 end) as jun sales
                                                       from
       , sum(case when d moy = 7)
                                                            web sales
                                                           ,warehouse
              then ws sales price*
ws quantity else 0 end) as jul sales
                                                           ,date dim
       , sum (case when d moy = 8
                                                           ,time dim
             then ws sales price*
                                                           ,ship mode
ws quantity else 0 end) as aug sales
                                                       where
       , sum(case when d moy = 9)
                                                              ws warehouse sk =
           then ws sales price*
                                                  w warehouse sk
ws quantity else 0 end) as sep_sales
                                                         and ws sold date sk = d date sk
       , sum(case when d_moy = 10)
                                                          and ws_sold_time_sk = t_time_sk
              then ws_sales_price*
                                                         and ws_ship_mode_sk =
ws quantity else 0 end) as oct sales
                                                  sm ship mode sk
       , sum(case when d_moy = 11
                                                          and d year = 2002
              then ws_sales_price*
                                                         and t_time between 18036 and
ws quantity else 0 end) as nov sales
                                                  18036+28800
       , sum (case when d moy = 12
                                                         and sm carrier in ('MSC', 'USPS')
             then ws_sales_price*
                                                       group by
                                                          w_warehouse_name
ws_quantity else 0 end) as dec_sales
       , sum(case when d moy = 1)
                                                          ,w_warehouse_sq_ft
              then
                                                          ,w city
ws_net_paid_inc_ship_tax * ws_quantity else
                                                          ,w_county
                                                         ,w_state
0 end) as jan net
      , sum (case when d moy = 2
                                                         ,w country
              then
                                                         ,d_year
ws_net_paid_inc_ship_tax * ws_quantity else
                                                         )
0 end) as feb net
                                                   union all
       , sum(case when d moy = 3)
                                                      (select
              t.hen
                                                         w warehouse name
ws net paid inc ship tax * ws quantity else
                                                         ,w warehouse sq ft
                                                          ,w_city
0 end) as mar net
       , sum(case when d moy = 4
                                                          ,w_county
                                                          ,w_state
              then
ws_net_paid_inc_ship_tax * ws_quantity else
                                                          ,w country
                                                          ,'MSC' || ',' || 'USPS' as
0 end) as apr net
       ,sum(case when d_moy = 5
                                                  ship_carriers
                                                         ,d_year as year
              then
ws net paid inc ship tax * ws quantity else
                                                          ,sum(case when d_moy = 1
0 end) as may net
                                                                 then cs_ext_sales_price*
       , sum(case when d_moy = 6
                                                  cs_quantity else 0 end) as jan_sales
                                                          , sum(case when d moy = 2
              then
ws net paid inc ship tax * ws quantity else
                                                                 then cs ext sales price*
                                                  cs_quantity else 0 end) as feb_sales
0 end) as jun_net
                                                          , sum (case when d moy = 3
       , sum(case when d moy = 7)
                                                                 then cs ext sales price*
                                                  cs quantity else 0 end) as mar sales
ws_net_paid_inc_ship_tax * ws_quantity else
                                                          , sum(case when d_moy = 4
0 end) as jul net
```

then cs ext sales price*	,ship mode
cs quantity else 0 end) as apr sales	where
, sum(case when d moy = 5	cs warehouse sk =
then cs_ext_sales_price*	w_warehouse_sk
cs_quantity else 0 end) as may_sales	and cs_sold_date_sk = d_date_sk
<pre>,sum(case when d_moy = 6</pre>	and cs_sold_time_sk = t_time_sk
then cs_ext_sales_price*	and cs_ship_mode_sk =
cs_quantity else 0 end) as jun_sales	sm_ship_mode_sk
$, sum(case when d_moy = 7)$	and $d_year = 2002$
then cs ext sales price*	and t time between 18036 AND
cs quantity else 0 end) as jul sales	18036+28800
<pre>, sum(case when d_moy = 8</pre>	<pre>and sm_carrier in ('MSC','USPS')</pre>
then cs_ext_sales_price*	group by
cs quantity else 0 end) as aug sales	w warehouse name
, sum(case when d moy = 9	,w_warehouse_sq_ft
then cs_ext_sales_price*	
cs_quantity else 0 end) as sep_sales	,w county
, sum(case when d moy = 10	,w_state
then cs ext sales price*	,w country
cs quantity else 0 end) as oct sales	,d year
, sum(case when d moy = 11)
then cs ext sales price*) x
cs quantity else 0 end) as nov sales	group by
sum(case when d moy = 12	w warehouse name
then cs ext sales price*	,w warehouse sq ft
cs quantity else 0 end) as dec sales	,w city
sum(case when d moy = 1	,w county
then cs_net_profit *	,w state
cs quantity else 0 end) as jan net	,w country
, sum(case when d moy = 2	,ship_carriers
then cs_net_profit *	, year
cs_quantity else 0 end) as feb_net	order by w warehouse name
sum(case when d moy = 3	fetch first 100 rows only;
then cs net profit *	-
cs quantity else 0 end) as mar net	end query 39 in stream 0 using template
, sum(case when d moy = 4	query66.tpl
then cs_net_profit *	start query 40 in stream 0 using
cs_quantity else 0 end) as apr_net	template query90.tpl and seed 1949014749
sum(case when d moy = 5	select cast(amc as decimal(15,4))/cast(pmc
then cs net profit *	as decimal(15,4)) am pm ratio
cs quantity else 0 end) as may net	from (select count(*) amc
, sum(case when d moy = 6	from web_sales,
then cs_net_profit *	household_demographics , time_dim, web_page
cs quantity else 0 end) as jun net	where ws_sold_time_sk =
, sum (case when d moy = 7	time dim.t time sk
then cs net profit *	and ws ship hdemo sk =
cs_quantity else 0 end) as jul_net	household_demographics.hd_demo_sk
sum(case when d moy = 8	and ws_web_page_sk =
then cs net profit *	web page.wp web page sk
cs quantity else 0 end) as aug net	and time_dim.t_hour between 11 and
, sum(case when d_moy = 9	11+1
then cs_net_profit *	and
cs quantity else 0 end) as sep net	household_demographics.hd_dep_count = 9
, sum(case when d moy = 10	and web_page.wp_char_count between
then cs_net_profit *	5000 and 5200) at,
cs quantity else 0 end) as oct net	(select count(*) pmc
, sum (case when d moy = 11	from web sales,
then cs_net_profit *	household_demographics , time_dim, web_page
cs_quantity else 0 end) as nov_net	where ws_sold_time_sk =
, sum (case when d moy = 12	time dim.t time sk
then cs_net_profit *	and ws ship hdemo sk =
cs_quantity else 0 end) as dec_net from	household_demographics.hd_demo_sk
	and ws_web_page_sk =
catalog_sales	web_page.wp_web_page_sk
,warehouse	and time_dim.t_hour between 18 and 18+1
,date_dim	TOLT
,time_dim	

```
and
                                                     fetch first 100 rows only;
household demographics.hd dep count = 9
         and web_page.wp_char_count between
                                                    -- end query 45 in stream 0 using template
5000 and 5200) pt
                                                   query3.tpl
order by am pm ratio
 fetch first 100 rows only;
                                                    -- start query 49 in stream 0 using
                                                    template query9.tpl and seed 937436805
-- end query 40 in stream 0 using template
                                                    select case when (select count(*)
query90.tpl
                                                                      from store sales
                                                                      where ss quantity between
-- start query 44 in stream 0 using
                                                    1 \text{ and } 20) > 62316685
template query92.tpl and seed 643980925
                                                                then (select
select
                                                    avg(ss_ext_sales_price)
  sum(ws_ext_discount_amt) as "Excess
                                                                      from store_sales
                                                                      where ss_quantity between
Discount Amount"
                                                    1 and 20)
   web sales
                                                                else (select
   ,item
                                                    avg(ss_net_paid_inc_tax)
                                                                      from store sales
   ,date dim
                                                                      where ss quantity between
i manufact id = 926
                                                    1 and 20) end bucket1 ,
and i item sk = ws_item_sk
                                                           case when (select count(*)
and d date between '1999-01-03' and
                                                                      from store sales
        (cast('1999-01-03' as date) + 90
                                                                      where ss quantity between
                                                    21 \text{ and } 40) > 19045798
days)
and d date sk = ws sold date sk
                                                                then (select
and ws ext discount amt
                                                    avg(ss ext sales price)
    > (
                                                                      from store sales
         SELECT
                                                                      where ss quantity between
           1.3 * avg(ws ext discount amt)
                                                    21 and 40)
                                                                else (select
           web_sales
                                                    avg(ss_net_paid inc tax)
           ,date dim
                                                                      from store sales
         WHERE
                                                                      where ss_quantity between
             ws_item_sk = i_item_sk
                                                    21 and 40) end bucket2,
                                                           case when (select count(*)
          and d date between '1999-01-03'
                                                                     from store sales
                             (cast('1999-
                                                                      where ss_quantity between
01-03' as date) + 90 days)
                                                    41 and 60) > 365541424
         and d date sk = ws sold date sk
                                                               then (select
                                                    avg(ss ext sales price)
order by sum(ws_ext_discount_amt)
                                                                      from store_sales
fetch first 100 rows only;
                                                                      where ss quantity between
                                                    41 and 60)
-- end query 44 in stream 0 using template
                                                                else (select
                                                    avg(ss_net_paid_inc_tax)
query92.tpl
-- start query 45 in stream 0 using
                                                                      from store sales
template query3.tpl and seed 691662667
                                                                      where ss quantity between
select dt.d_year
                                                    41 and 60) end bucket3,
       ,item.i brand id brand id
                                                           case when (select count(*)
       ,item.i brand brand
                                                                      from store sales
       ,sum(ss net profit) sum agg
                                                                      where ss quantity between
from date dim dt
                                                    61 and 80) > 216357808
     ,store sales
                                                                then (select
      ,item
                                                    avg(ss_ext_sales_price)
where dt.d_date_sk =
                                                                      from store_sales
store sales.ss sold date sk
                                                                      where ss\_quantity between
  and store sales.ss item sk =
                                                    61 and 80)
item.i_item_sk
                                                                else (select
  and item.i_manufact_id = 596
                                                    avg(ss_net_paid_inc_tax)
   and dt.d moy=12
                                                                      from store sales
group by dt.d year
                                                                      where ss quantity between
      ,item.i_brand
                                                    61 and 80) end bucket4,
      ,item.i brand id
                                                           case when (select count(*)
 order by dt.d year
                                                                      from store sales
        ,sum agg desc
                                                                      where ss_quantity between
         ,brand id
                                                    81 and 100) > 184483884
```

```
then (select
                                                                          ,i brand
                                                                          ,i class
avg(ss ext sales price)
                  from store sales
                                                                          ,i category)
                 where ss quantity between
                                                   order by qoh, i_product_name, i_brand,
                                                   i class, i category
81 and 100)
            else (select
                                                    fetch first 100 rows only;
avg(ss net paid inc tax)
                 from store sales
                                                   -- end query 55 in stream 0 using template
                 where ss quantity between
                                                   query22.tpl
81 and 100) end bucket5
                                                   -- start query 56 in stream 0 using
                                                   template query89.tpl and seed 2079706651
from reason
where r reason sk = 1
                                                   select *
                                                   from(
                                                   select i_category, i_class, i_brand,
-- end query 49 in stream 0 using template
                                                         s_store_name, s_company_name,
query9.tpl
                                                          d moy,
                                                          sum(ss sales price) sum sales,
-- start query 52 in stream 0 using
                                                          avg(sum(ss_sales_price)) over
template query93.tpl and seed 1821797098
                                                            (partition by i_category, i_brand,
select ss customer sk
                                                   s store name, s company name)
           ,sum(act sales) sumsales
                                                           avg monthly sales
      from (select ss item sk
                                                   from item, store sales, date dim, store
                 ,ss ticket number
                                                   where ss item sk = i item sk and
                                                         ss sold date sk = d date sk and
                  ,ss customer sk
                  ,case when
                                                         ss\_store\_sk = s\_store\_sk and
sr return quantity is not null then
                                                         d year in (1999) and
(ss quantity-
                                                           ((i category in
                                                   ('Books','Jewelry','Men') and
sr return quantity)*ss sales price
                                                            i class in
                                                   ('history','birdal','pants')
else (ss quantity*ss sales price) end
act_sales
                                                          )
           from store_sales left outer
                                                         or (i_category in
                                                   ('Music', 'Home', 'Shoes') and
join store returns on (sr item sk =
ss item sk
                                                             i class in
                                                   ('pop','furniture','athletic')
and sr_ticket_number = ss_ticket_number)
                                                          ))
              ,reason
                                                   group by i category, i class, i brand,
            where sr_reason_sk =
                                                           s_store_name, s_company_name,
r_reason_sk
                                                   d moy) tmp1
             and r_reason_desc = 'reason
                                                   where case when (avg_monthly_sales <> 0)
                                                   then (abs(sum sales - avg monthly sales) /
      group by ss_customer_sk
                                                   avg_monthly_sales) else null end > 0.1
     order by sumsales, ss_customer_sk
                                                   order by sum sales - avg monthly sales,
fetch first 100 rows only;
                                                   s store name
                                                    fetch first 100 rows only;
-- end query 52 in stream 0 using template
query93.tpl
                                                   -- end query 56 in stream 0 using template
                                                   query89.tpl
-- start query 55 in stream 0 using
                                                   -- start query 59 in stream 0 using
template query22.tpl and seed 635815297
                                                   template query52.tpl and seed 223505300
select i product name
                                                   select dt.d year
                                                          ,item.i brand id brand id
            ,i brand
             ,i_class
                                                          ,item.i_brand brand
             ,i category
                                                          ,sum(ss ext sales price) ext price
             ,avg(cast(inv_quantity_on_hand
                                                    from date_dim dt
as double)) qoh
                                                       ,store_sales
      from inventory
                                                        ,item
          ,date dim
                                                    where dt.d date sk =
           ,item
                                                   store_sales.ss_sold_date_sk
          ,warehouse
                                                       and store_sales.ss_item_sk =
       where inv_date_sk=d_date_sk
                                                   item.i item sk
              and inv item sk=i item sk
                                                       and item.i manager id = 1
             and inv_warehouse_sk =
                                                       and dt.d moy=11
                                                      and dt.d year=1999
w warehouse sk
              and d month seq between 1199
                                                  group by dt.d year
and 1199 + 11
                                                          ,item.i brand
                                                          ,item.i_brand_id
      group by rollup(i_product_name
```

```
order by dt.d year
                                                   order by s store name
       ,ext_price desc
                                                          ,s company id
       ,brand id
                                                           ,s street number
fetch first 100 rows only;
                                                           ,s_street_name
                                                           ,s street type
                                                           ,s_suite_number
-- end query 59 in stream 0 using template
querv52.tpl
                                                           ,s city
-- start query 60 in stream 0 using
                                                           ,s_county
template query50.tpl and seed 1718577076
                                                           ,s_state
                                                           ,s zip
  s_store name
                                                    fetch first 100 rows only;
 ,s_company id
  ,s street number
                                                   -- end query 60 in stream 0 using template
  ,s_street_name
                                                   query50.tpl
 ,s_street_type
                                                   -- start query 61 in stream 0 using
                                                   template query42.tpl and seed 709936855
 ,s suite number
 ,s_city
                                                   select dt.d year
 ,s_county
                                                          ,item.i_category_id
                                                          ,item.i_category
 ,s_state
                                                          , sum (ss ext sales price)
 s zip,
                                                    from date_dim dt
 ,sum(case when (sr returned date sk -
                                                          ,store sales
ss sold date sk \le 30 ) then 1 else 0 end)
                                                          ,item
                                                    where dt.d date sk =
 ,sum(case when (sr_returned_date_sk -
ss_sold_date_sk > 30) and
                                                   store_sales.ss_sold date sk
                (sr returned date sk -
                                                          and store sales.ss item sk =
ss sold date sk \le 60) then 1 else 0 end )
                                                   item.i item sk
as "31-60 days"
                                                          and item.i manager id = 1
 ,sum(case when (sr returned date sk -
                                                          and dt.d mov=12
ss_sold_date_sk > 60) and
                                                          and dt.d year=2000
               (sr_returned_date_sk -
                                                    group by
                                                                 dt.d_year
ss_sold_date_sk <= 90) then 1 else 0 end)
                                                                  ,item.i_category_id
                                                                  ,item.i_category
as "61-90 days"
 ,sum(case when (sr_returned_date_sk -
                                                    order by
                                                                   sum(ss_ext_sales_price)
ss\_sold\_date\_sk > 90) and
                                                   desc,dt.d_year
                 (sr returned date sk -
                                                                  ,item.i_category_id
ss sold date sk \leq 120) then 1 else 0 end)
                                                                  ,item.i category
as "91-120 days"
                                                    fetch first 100 rows only;
 ,sum(case when (sr_returned_date_sk -
ss sold date sk > 120) then 1 else 0 end)
                                                   -- end query 61 in stream 0 using template
as ">120 days"
                                                   query42.tpl
                                                   -- start query 62 in stream 0 using
from
                                                   template query41.tpl and seed 944250029
  store sales
                                                   select distinct(i product name)
 ,store returns
 ,store
                                                    from item i1
 ,date_dim d1
                                                    where i_manufact_id between 716 and 716+40
 ,date dim d2
                                                      and (select count(*) as item cnt
                                                           from item
   d2.d_year = 1999
                                                           where (i manufact = i1.i manufact
and d2.d mov = 10
                                                   and
and ss ticket number = sr ticket number
                                                           ((i category = 'Women' and
and ss item sk = sr item sk
                                                           (i color = 'spring' or i color =
and ss_sold_date_sk = d1.d_date_sk
                                                   'hot') and
and sr_returned_date_sk = d2.d_date sk
                                                           (i units = 'Carton' or i units =
and ss_customer_sk = sr_customer_sk
                                                   'Tbl') and
and ss_store_sk = s_store_sk
                                                           (i_size = 'large' or i_size =
group by
                                                   'N/A')
  s store name
                                                           (i\_category = 'Women' and
 ,s_company_id
                                                           (i_color = 'magenta' or i_color =
 ,s_street_number
 ,s_street_name
                                                   'goldenrod') and
                                                           (i units = 'Cup' or i units = 'Oz')
  ,s street type
  ,s_suite_number
                                                   and
 ,s_city
                                                           (i size = 'economy' or i size =
                                                   'extra large')
 ,s county
 ,s_state
                                                           (i category = 'Men' and
  ,s_zip
```

```
(i color = 'cyan' or i color =
                                                  from
'antique') and
                                                          web sales
        (i units = 'Dozen' or i units =
                                                          ,item
'Case') and
                                                          ,date dim
       (i size = 'medium' or i size =
                                                  where
'petite')
                                                          ws item sk = i item sk
                                                          and i category in ('Jewelry',
       ) or
        (i\_category = 'Men' and
                                                   'Men', 'Books')
        (i color = 'moccasin' or i color =
                                                          and ws sold date sk = d date sk
                                                         and d date between cast('2002-06-
'black') and
       (i_units = 'Box' or i units =
                                                  11' as date)
'Pallet') and
                                                   (cast('2002-06-11' as date) + 30 days)
        (i_size = 'large' or i_size =
'N/A')
                                                   group by
       ))) or
                                                          i item id
                                                          ,i_item_desc
       (i manufact = i1.i manufact and
        ((i\_category = 'Women' and
                                                          ,i_category
        (i color = 'azure' or i color =
                                                          ,i_class
                                                          ,i_current price
'light') and
                                                  order by
        (i units = 'Gross' or i units =
'Each') and
                                                          i category
       (i_size = 'large' or i size =
                                                          ,i_class
'N/A')
                                                          ,i_item_id
        ) or
                                                          ,i item desc
        (i_category = 'Women' and
                                                          ,revenueratio
        (i color = 'mint' or i color =
                                                   fetch first 100 rows only;
'burnished') and
       (i units = 'N/A' or i units =
                                                  -- end query 64 in stream 0 using template
'Unknown') and
                                                  query12.tpl
       (i size = 'economy' or i size =
                                                   -- start query 65 in stream 0 using
'extra large')
                                                  template query20.tpl and seed 711739272
       ) or
                                                  select i_item_desc
                                                         ,i_category
        (i category = 'Men' and
        (i_color = 'floral' or i_color =
                                                         ,i_class
'midnight') and
                                                         ,i_current_price
       (i_units = 'Pound' or i units =
                                                         ,sum(cs_ext_sales_price) as
       (i_size = 'medium' or i_size =
'petite')
                                                   , sum(cs_ext_sales_price)*100/sum(sum(cs_ext
                                                  _sales_price)) over
       ) or
        (i category = 'Men' and
                                                             (partition by i class) as
        ...
(i color = 'navy' or i_color =
                                                  revenueratio
                                                   from catalog_sales
'blue') and
                                                       ,item
       (i units = 'Bundle' or i units =
'Ounce') and
                                                       ,date dim
       (i_size = 'large' or i size =
                                                   where cs item sk = i item sk
                                                    and i category in ('Jewelry', 'Music',
'N/A')
       ))))) > 0
                                                   'Men')
                                                     and cs sold date sk = d date sk
order by i product name
 fetch first 100 rows only;
                                                   and d date between cast('2000-02-09' as
-- end query 62 in stream 0 using template
                                                                                 and
                                                  (cast('2000-02-09' as date) + 30 days)
query41.tpl
                                                   group by i item id
-- start query 64 in stream 0 using
                                                           ,i_item_desc
template query12.tpl and seed 918962166
                                                            ,i_category
select i_item_desc
                                                           ,i_class
     ,i category
                                                           ,i current price
     ,i_class
                                                   order by i_category
                                                           ,i_class
     ,i_current_price
                                                           ,i_item_id
     , sum(ws_ext_sales_price) as
                                                           ,i item desc
                                                            ,revenueratio
,sum(ws ext sales price) *100/sum(sum(ws ext
                                                   fetch first 100 rows only;
_sales_price)) over
      (partition by i class) as
                                                  -- end query 65 in stream 0 using template
revenueratio
                                                  query20.tpl
```

```
-- start query 66 in stream 0 using
                                                   household demographics.hd vehicle count<=1+
template query88.tpl and seed 1924183468
select *
                                                   (household demographics.hd dep count = 4
(select count(*) h8 30 to 9
from store sales, household demographics,
                                                   household demographics.hd vehicle count<=4+
time dim, store
                                                   2) or
where ss sold time_sk = time_dim.t_time_sk
    and ss hdemo sk =
                                                   (household demographics.hd dep count = 2
household demographics.hd demo sk
                                                   household demographics.hd vehicle count<=2+
    and ss_store_sk = s_store_sk
     and time dim.t hour = 8
                                                        and store.s_store_name = 'ese') s3,
     and time dim.t minute >= 30
                                                     (select count(*) h10_to_10_30
     and
((household demographics.hd_dep_count = 1
                                                    from store sales, household demographics,
                                                   time dim, store
household_demographics.hd_vehicle_count<=1+
                                                    where ss sold time sk = time dim.t time sk
2) or
                                                        and ss_hdemo_sk =
                                                   household demographics.hd demo sk
(household demographics.hd dep count = 4
                                                        and ss store sk = s store sk
                                                        and time \dim.t hour = 10
household demographics.hd vehicle count<=4+
                                                        and time dim.t minute < 30
                                                   ((household demographics.hd dep count = 1
(household demographics.hd dep count = 2
                                                   and
                                                   household demographics.hd vehicle count<=1+
household demographics.hd vehicle count<=2+
2))
    and store.s store name = 'ese') s1,
                                                   (household demographics.hd dep count = 4
(select count(*) h9 to 9 30
from store_sales, household_demographics ,
                                                   household_demographics.hd_vehicle_count<=4+
time_dim, store
where ss sold time sk = time dim.t time sk
    and ss hdemo sk =
                                                   (household demographics.hd dep count = 2
household_demographics.hd_demo_sk
    and ss_store_sk = s_store_sk
                                                   household demographics.hd vehicle count<=2+
     and time dim.t hour = 9
    and time_dim.t_minute < 30
                                                        and store.s_store_name = 'ese') s4,
                                                    (select count(*) h10_30_to_11
((household demographics.hd dep count = 1
                                                    from store sales, household demographics ,
                                                   time_dim, store
household_demographics.hd_vehicle_count<=1+
                                                    where ss_sold_time_sk = time_dim.t_time_sk
                                                        and ss hdemo sk =
2) or
                                                   household demographics.hd demo sk
(household demographics.hd dep count = 4
                                                        and ss store sk = s store sk
                                                        and time_dim.t_hour = 10
household demographics.hd vehicle count<=4+
                                                        and time dim.t minute >= 30
2) or
                                                        and
                                                   ((household demographics.hd dep count = 1
(household demographics.hd dep count = 2
                                                   household demographics.hd vehicle count<=1+
household demographics.hd vehicle count<=2+
                                                   2) or
                                                   (household demographics.hd dep count = 4
    and store.s store name = 'ese') s2,
 (select count(*) h9_30_to_10
from store_sales, household_demographics ,
                                                   household_demographics.hd_vehicle_count<=4+
time dim, store
                                                   2) or
where ss sold time sk = time dim.t time sk
    and ss hdemo sk =
                                                   (household demographics.hd dep count = 2
\verb|household_demographics.hd_demo_sk|
    and ss_store_sk = s_store_sk
                                                   household demographics.hd vehicle count<=2+
     and time dim.t hour = 9
                                                        and store.s_store_name = 'ese') s5,
     and time_dim.t_minute >= 30
                                                    (select count(*) h11 to 11 30
((household demographics.hd dep count = 1
                                                    from store sales, household demographics,
                                                   time dim, store
                                                    where ss\_sold\_time\_sk = time\_dim.t\_time\_sk
```

```
and ss hdemo sk =
household demographics.hd demo sk
                                                   household demographics.hd vehicle count<=2+
    and ss_store_sk = s_store_sk
                                                         and store.s_store name = 'ese') s8
     and time_dim.t_hour = 11
     and time dim.t minute < 30
     and
                                                   -- end query 66 in stream 0 using template
((household demographics.hd dep count = 1
                                                   query88.tpl
household demographics.hd vehicle count<=1+
                                                   -- start query 72 in stream 0 using
                                                   template query71.tpl and seed 1436004490
(household demographics.hd dep count = 4
                                                    select i brand id brand id, i brand
                                                   brand, t_hour, t_minute,
household_demographics.hd_vehicle_count<=4+
                                                           sum(ext_price) ext_price
2) or
                                                    from item, (select ws_ext_sales_price as
                                                   ext price,
(household demographics.hd dep count = 2
                                                                            ws sold date sk as
                                                   sold date sk,
household demographics.hd vehicle count<=2+
                                                                            ws item sk as
                                                   sold item sk,
     and store.s store name = 'ese') s6,
                                                                            ws sold time sk as
(select count(*) h11 30 to 12
                                                   time sk
from store sales, household demographics,
                                                                     from web sales, date dim
time dim, store
                                                                     where d date sk =
where ss_sold_time_sk = time_dim.t_time_sk
                                                   ws sold date sk
    and ss hdemo sk =
                                                                       and d moy=12
household demographics.hd demo sk
                                                                      and d year=1998
    and ss store sk = s store sk
                                                                     union all
    and time dim.t hour = 11
                                                                     select cs_ext_sales_price
    and time dim.t minute >= 30
                                                   as ext price,
    and
                                                                            cs_sold_date_sk as
((household demographics.hd_dep_count = 1
                                                   sold date sk,
                                                                            cs item sk as
household_demographics.hd_vehicle_count<=1+
                                                   sold_item_sk,
2) or
                                                                            cs_sold_time_sk as
                                                   time sk
(household demographics.hd dep count = 4
                                                                     from
                                                   catalog_sales,date_dim
household demographics.hd vehicle count<=4+
                                                                     where d date sk =
2) or
                                                   cs sold date sk
                                                                       and d moy=12
(household demographics.hd dep count = 2
                                                                      and d year=1998
                                                                     union all
household demographics.hd vehicle count<=2+
                                                                     select ss ext sales price
                                                   as ext price,
     and store.s_store_name = 'ese') s7,
                                                                            ss sold date sk as
(select count(*) h12 to 12 30
                                                   sold date sk,
 from store sales, household demographics,
                                                                            ss item sk as
time dim, store
                                                   sold item sk,
where ss sold time_sk = time_dim.t_time_sk
                                                                            ss sold time sk as
    and ss hdemo sk =
                                                   time sk
household demographics.hd demo sk
                                                                     from store sales, date dim
    and ss store_sk = s_store_sk
                                                                     where d_date_sk =
     and time dim.t hour = 12
                                                   ss sold date sk
     and time_dim.t_minute < 30
                                                                       and d moy=12
    and
                                                                       and d_year=1998
((household_demographics.hd_dep_count = 1
                                                                     ) as tmp, time dim
                                                    where
household_demographics.hd_vehicle_count<=1+
                                                      sold item sk = i item sk
                                                      and i_manager_id=1
2) or
                                                      and time_sk = t_time_sk
(household demographics.hd dep count = 4
                                                      and (t meal time = 'breakfast' or
                                                    t_meal_time = 'dinner')
household demographics.hd vehicle count<=4+
                                                    group by i brand,
                                                    i brand id, t hour, t minute
2) or
                                                    order by ext price desc, i brand id
(household demographics.hd dep count = 2
```

```
sum(ss net profit) as profit
                                                    from store_sales,
-- end query 72 in stream 0 using template
query71.tpl
                                                        date dim,
-- start query 73 in stream 0 using
                                                         store
                                                    where ss sold date sk = d date sk
template query34.tpl and seed 1451328249
select c last name
                                                          and d date between cast('2002-08-24'
       ,c first name
                                                   as date)
      ,c_salutation
                                                                     and (cast('2002-08-24' as
                                                   date) + 30 days)
      ,c preferred cust flag
       ,ss ticket number
                                                         and ss store sk = s store sk
       ,cnt from
                                                    group by s_store_sk)
   (select ss ticket number
          ,ss customer sk
                                                    sr as
          ,count(*) cnt
                                                    (select s_store_sk,
                                                            sum(sr_return_amt) as returns,
store sales, date dim, store, household demogr
                                                            sum(sr net loss) as profit loss
                                                    from store returns,
   where store_sales.ss_sold_date_sk =
                                                        date_dim,
date dim.d date sk
                                                         store
    and store sales.ss store sk =
                                                    where sr returned date sk = d date sk
store.s store sk
                                                          and d date between cast('2002-08-24'
   and store sales.ss hdemo sk =
                                                   as date)
household demographics.hd demo sk
                                                                     and (cast('2002-08-24' as
   and (date dim.d dom between 1 and 3 or
                                                   date) + 30 days)
date_dim.d_dom between 25 and 28)
                                                         and sr_store_sk = s_store_sk
   and
                                                    group by s store sk),
(household demographics.hd buy potential =
                                                    cs as
'1001-5000' or
                                                    (select cs call center sk,
                                                           sum(cs ext_sales_price) as sales,
household demographics.hd buy potential =
                                                           sum(cs net profit) as profit
'5001-10000')
                                                    from catalog_sales,
                                                         date dim
household demographics.hd vehicle count > 0
                                                    where cs sold date sk = d date sk
    and (case when
                                                          and d date between cast('2002-08-24'
household_demographics.hd_vehicle_count > 0
                                                   as date)
                                                                     and (cast('2002-08-24') as
cast(household demographics.hd dep count as
                                                   date) + 30 days)
double) /
                                                    group by cs_call_center_sk
cast(household demographics.hd vehicle coun
                                                    ),
t as double)
                                                    cr as
       else null
                                                     (select
       end) > 1.2
                                                           sum(cr_return_amount) as returns,
   and date dim.d_year in
                                                           sum(cr net loss) as profit loss
(1999, 1999+1, 1999+2)
                                                    from catalog returns,
   and store.s county in ('Sierra
                                                         date dim
County', 'Lunenburg County', 'Jackson
                                                    where cr_returned_date_sk = d_date_sk
County', 'Harmon County',
                                                          and d date between cast('2002-08-24'
                                                   as date)
County', 'Pipestone County', 'Pennington
                                                                     and (cast('2002-08-24' as
County','Perry County')
                                                   date) + 30 days)
   group by
                                                    ),
                                                    ws as
ss ticket number, ss customer sk)
dn,customer
                                                    ( select wp_web_page_sk,
    where ss_customer_sk = c_customer_sk
                                                           sum(ws ext sales price) as sales,
     and cnt between 15 and 20
                                                           sum(ws_net_profit) as profit
                                                    from web_sales,
    order by
c_last_name,c_first_name,c_salutation,c_pre
                                                        date dim,
ferred cust flag desc;
                                                         web page
                                                    where ws_sold_date_sk = d_date_sk
-- end query 73 in stream 0 using template
                                                          and d date between cast('2002-08-24'
query34.tpl
                                                   as date)
                                                                     and (cast('2002-08-24' as
                                                   date) + 30 days)
-- start query 78 in stream 0 using
template query77.tpl and seed 1879081522
                                                          and ws_web_page_sk = wp_web_page_sk
                                                    group by wp web page sk),
with ss as
 (select s store sk,
                                                    wr as
        sum(ss_ext_sales_price) as sales,
                                                    (select wp_web_page_sk,
```

```
sum(wr return amt) as returns,
                                                      where store sales.ss sold date sk =
       sum(wr net loss) as profit loss
                                                   date dim.d date sk
 from web returns,
                                                      and store sales.ss store sk =
     date_dim,
                                                   store.s_store_sk
     web page
                                                      and store sales.ss hdemo sk =
where wr returned date sk = d date sk
                                                   household demographics.hd demo sk
      and d date between cast('2002-08-24'
                                                      and date dim.d dom between 1 and 2
as date)
                 and (cast('2002-08-24' as
                                                   (household demographics.hd buy potential =
date) + 30 days)
                                                   '501-1000' or
      and wr_web_page_sk = wp_web_page_sk
group by wp_web_page_sk)
                                                   household demographics.hd buy potential =
  select channel
                                                   '5001-10000')
       , id
                                                      and
        , sum(sales) as sales
                                                   household demographics.hd vehicle count > 0
        , sum(returns) as returns
                                                      and case when
        , sum(profit) as profit
                                                   household demographics.hd vehicle count > 0
                                                   then
 from
 (select 'store channel' as channel
       , ss.s store sk as id
                                                   household demographics.hd dep count/
                                                   household demographics.hd vehicle count
        , sales
        , coalesce(returns, 0) as returns
                                                   else null end > 1
                                                      and date dim.d year in
        , (profit -
                                                   (1999,1999+1,1999+2)
coalesce(profit loss,0)) as profit
from ss left join sr
                                                      and store.s_county in ('Lea
       on ss.s store sk = sr.s store sk
                                                   County', 'West Feliciana Parish', 'Nowata
                                                   County', 'Jackson County')
union all
select 'catalog channel' as channel
                                                      group by
                                                   ss ticket number,ss_customer_sk)
       , cs call center sk as id
       , sales
                                                   dj,customer
       , returns
                                                      where ss_customer_sk = c_customer_sk
       , (profit - profit_loss) as profit
                                                        and cnt between 1 and 5
                                                      order by cnt desc;
 from cs
      , cr
union all
                                                   -- end query 79 in stream 0 using template
select 'web channel' as channel
                                                   query73.tpl
        , ws.wp web page sk as id
                                                   -- start query 80 in stream 0 using
        , sales
                                                   template query84.tpl and seed 1842474049
                                                   select c_customer_id as customer_id
        , coalesce(returns, 0) returns
        , (profit -
                                                         ,c_last_name || ', ' ||
coalesce(profit loss,0)) as profit
                                                   coalesce(c first name,'') as customername
from ws left join wr
                                                   from customer
                                                       ,customer address
       on ws.wp web page sk =
                                                       ,customer demographics
wr.wp_web_page_sk
) x
                                                       , household demographics
                                                       ,income_band
group by rollup (channel, id)
                                                       ,store returns
order by channel
                                                                          = 'Mount Zion'
        ,id
                                                    where ca city
 fetch first 100 rows only;
                                                      and c current addr sk = ca address sk
                                                      and ib lower bound >= 50749
-- end query 78 in stream 0 using template
                                                      and ib upper bound <= 50749 + 50000
                                                      and ib income band sk =
query77.tpl
-- start query 79 in stream 0 using
                                                   hd_income_band_sk
template query73.tpl and seed 413577677
                                                      and cd_demo_sk = c_current_cdemo_sk
select c last name
                                                      and hd_demo_sk = c_current_hdemo_sk
                                                      and sr_cdemo_sk = cd_demo_sk
       ,c_first_name
      ,c_salutation
                                                   order by c customer id
      ,c preferred cust flag
                                                    fetch first 100 rows only;
      ,ss_ticket_number
                                                   -- end query 80 in stream 0 using template
       ,cnt from
                                                   query84.tpl
   (select ss_ticket_number
         ,ss customer sk
          ,count(*) cnt
                                                   -- start query 82 in stream 0 using
                                                   template guery55.tpl and seed 1117454508
store sales, date dim, store, household demogr
                                                   select i brand id brand id, i brand brand,
aphics
                                                          sum(ss_ext_sales_price) ext_price
                                                   from date_dim, store_sales, item
```

```
where d date sk = ss sold date sk
                                                          item
       and ss item sk = i item sk
       and i_manager_id=48
                                                          i item id
                                                                                  in (select
       and d moy=11
                                                   i item id
       and d year=2001
                                                  from item
group by i brand, i brand id
                                                  where i color in
order by ext price desc, i brand id
                                                  ('maroon','powder','lawn'))
fetch first 100 rows only;
                                                  and ws item sk
                                                  i item sk
-- end query 82 in stream 0 using template
                                                  and
                                                          ws sold date sk
                                                  d_date_sk
query55.tpl
-- start query 83 in stream 0 using
                                                   and
                                                        d year
                                                                                  = 2000
template query56.tpl and seed 1152645577
                                                   and
                                                          d moy
                                                                                  = 1
                                                          ws_bill_addr_sk
with ss as (
                                                   and
                                                  ca_address sk
select i_item_id,sum(ss_ext_sales_price)
total sales
                                                   and ca gmt offset
from
                                                   group by i_item_id)
       store_sales,
                                                   select i_item_id ,sum(total_sales)
       date dim,
                                                  total sales
        customer address,
                                                   from (select * from ss
                                                         union all
where i item id in (select
                                                         select * from cs
    i item_id
                                                         union all
from item
                                                         select * from ws) tmp1
                                                   group by i_item_id
where i color in
('maroon','powder','lawn'))
                                                  order by total sales
and ss item sk
                                                   fetch first 100 rows only;
i item sk
                                                  -- end query 83 in stream 0 using template
and
       ss sold date sk
d date sk
                                                 query56.tpl
                                = 2000
                                                  -- start query 84 in stream 0 using
and d_year
       d_moy
and
                                = 1
                                                  template query2.tpl and seed 1528114170
and
       ss addr sk
                                                  with wscs as
ca address sk
                                                   (select sold_date_sk
and ca_gmt_offset
                                = -5
                                                        ,sales_price
group by i_item_id),
                                                   from (select ws_sold_date_sk sold_date_sk
                                                               ,ws_ext_sales_price
select i item id, sum(cs ext sales price)
                                                  sales price
                                                          from web_sales) x
total_sales
from
                                                          union all
       catalog sales,
                                                         (select cs sold date sk sold date sk
       date dim,
                                                               ,cs_ext_sales_price
                                                  sales_price
        customer address,
                                                        from catalog sales)),
where
                                                   wswscs as
        i_item_id
                                in (select
                                                   (select d week seq,
                                                         sum(case when (d_day_name='Sunday')
 i item id
from item
                                                  then sales price else null end) sun sales,
where i color in
                                                         sum(case when (d day name='Monday')
('maroon','powder','lawn'))
                                                  then sales price else null end) mon sales,
and
     cs item sk
                                                         sum(case when
i item sk
                                                  (d day name='Tuesday') then sales price
and cs_sold_date_sk
                                                  else null end) tue_sales,
d date sk
                                                         sum(case when
and
      d_year
                                = 2000
                                                  (d_day_name='Wednesday') then sales_price
       d_moy
                                = 1
                                                  else null end) wed_sales,
and
       cs bill_addr_sk
                                                        sum(case when
and
ca address sk
                                                  (d day name='Thursday') then sales price
and ca_gmt_offset
                                = -5
                                                  else null end) thu_sales,
                                                         sum(case when (d_day_name='Friday')
group by i_item_id),
                                                  then sales price else null end) fri sales,
ws as (
select i item id, sum(ws ext sales price)
                                                          sum(case when
                                                  (d_day_name='Saturday') then sales_price
total_sales
                                                  else null end) sat sales
from
       web sales,
                                                  from wscs
       date dim,
                                                      ,date dim
                                                   where d_date_sk = sold_date sk
        customer address,
```

```
-- start query 86 in stream 0 using
group by d week seg)
 select d week seq1
                                                   template query40.tpl and seed 600490395
      , round(sun sales1/sun sales2,2)
                                                   select
       , round(mon_sales1/mon_sales2,2)
                                                     w state
       ,round(tue_sales1/tue sales2,2)
                                                     ,i_item id
       , round(wed sales1/wed sales2,2)
                                                    ,sum(case when (cast(d date as date) <
       ,round(thu sales1/thu sales2,2)
                                                   cast ('2000-04-27' as date))
                                                                 then cs_sales_price -
       , round(fri sales1/fri sales2,2)
      , round(sat sales1/sat sales2,2)
                                                   coalesce(cr refunded cash,0) else 0 end) as
 from
                                                   sales before
 (select wswscs.d_week_seq d_week_seq1
                                                   ,sum(case when (cast(d date as date) >=
        ,sun sales sun sales1
                                                   cast ('2000-04-27' as date))
        ,mon sales mon sales1
                                                                 then cs sales price -
        ,tue sales tue sales1
                                                   coalesce(cr_refunded_cash,0) else 0 end) as
        ,wed sales wed sales1
                                                   sales_after
        ,thu sales thu sales1
        ,fri_sales fri_sales1
                                                     catalog_sales left outer join
        ,sat_sales sat_sales1
                                                   catalog_returns on
  from wswscs, date dim
                                                          (cs order number = cr order number
  where date dim.d week seq =
                                                          and cs item sk = cr item sk)
wswscs.d week seq and
                                                     ,warehouse
                                                    ,item
       d year = 1998) y,
 (select wswscs.d week seq d week seq2
                                                    ,date dim
        ,sun sales sun sales2
                                                    where
                                                       i_current_price between 0.99 and 1.49
        , mon_sales mon_sales2
        , tue sales tue sales2
                                                    and i item sk = cs item sk
        , wed sales wed sales2
                                                    and cs warehouse sk
                                                                          = w warehouse sk
                                                    and cs sold date sk = d date sk
        ,thu sales thu sales2
        ,fri sales fri sales2
                                                   and d date between (cast ('2000-04-27' as
       ,sat sales sat sales2
                                                   date) - 30 days)
                                                                   and (cast ('2000-04-27' as
  from wswscs
     ,date dim
                                                   date) + 30 days)
  where date dim.d week seq =
                                                   group by
wswscs.d_week_seq and
                                                       w_state,i_item_id
      d_year = 1998+1) z
                                                    order by w_state,i_item_id
where d week seq1=d week_seq2-53
                                                   fetch first 100 rows only;
order by d week seq1;
                                                   -- end query 86 in stream 0 using template
                                                   query40.tpl
-- end query 84 in stream 0 using template
query2.tpl
 - start query 85 in stream 0 using
                                                   -- start query 88 in stream 0 using
template query26.tpl and seed 1427200905
                                                   template query53.tpl and seed 1796782974
select i item id,
                                                   select * from
       avg(cast(cs quantity as double))
                                                   (select i manufact id,
agg1,
                                                   sum(ss sales price) sum sales,
       avg(cs_list_price) agg2,
                                                   avg(sum(ss_sales_price)) over (partition by
       avg(cs coupon amt) agg3,
                                                   i manufact id) avg quarterly sales
        avg(cs sales price) agg4
                                                   from item, store sales, date dim, store
from catalog sales, customer demographics,
                                                   where ss item sk = i item sk and
                                                   ss sold date sk = d date sk and
date dim, item, promotion
where cs sold date sk = d date sk and
                                                   ss store sk = s store sk and
      cs item sk = i item sk and
                                                   d month seq in
       cs_bill_cdemo_sk = cd_demo_sk and
                                                   (1198,1198+1,1198+2,1198+3,1198+4,1198+5,11
                                                   98+6,1198+7,1198+8,1198+9,1198+10,1198+11)
       cs promo sk = p promo sk and
       cd gender = 'M' and
      cd_marital_status = 'D' and
                                                   ((i_category in
      cd_education_status = 'Advanced
                                                   ('Books','Children','Electronics') and
Degree' and
                                                   i class in
      (p_channel_email = 'N' or
                                                   ('personal', 'portable', 'reference', 'self-
p_channel_event = 'N') and
                                                   help') and
      d_year = 2000
                                                   i brand in ('scholaramalgamalg
 group by i item id
                                                   #14','scholaramalgamalg #7',
order by i_item_id
                                                                  'exportiunivamalg
 fetch first 100 rows only;
                                                   #9','scholaramalgamalg #9'))
                                                   or(i category in ('Women','Music','Men')
-- end query 85 in stream 0 using template
                                                   and
query26.tpl
```

```
i class in
                                                         date dim
('accessories','classical','fragrances','pa
                                                    where sr item sk = i item sk
nts') and
                                                    and d_date in
i_brand in ('amalgimporto #1','edu
                                                          (select d date
packscholar #1', 'exportiimporto #1',
                                                           from date dim
               'importoamalg #1')))
                                                           where d week seq in
group by i manufact id, d qoy ) tmp1
                                                                  (select d week seq
where case when avg quarterly sales > 0
                                                                  from date dim
       then abs (sum sales -
                                                            where d date in ('1999-06-
                                                   14','1999-08-26','1999-11-06')))
avg quarterly sales) / avg quarterly sales
      else null end > 0.1
                                                    and sr_returned_date_sk = d_date_sk
order by avg quarterly sales,
                                                    group by i item id),
        sum sales,
                                                    cr items as
        i_manufact_id
                                                    (select i_item_id item_id,
fetch first 100 rows only;
                                                          sum(cr_return_quantity) cr_item_qty
                                                    from catalog returns,
-- end query 88 in stream 0 using template
                                                         item,
                                                         date_dim
query53.tpl
-- start query 89 in stream 0 using
                                                    where cr_item_sk = i_item_sk
template query79.tpl and seed 2112737383
                                                    and d date in
                                                           (select d date
                                                           from date dim
c last name, c first name, substr(s city, 1, 30
                                                           where d week seq in
),ss ticket number,amt,profit
                                                                  (select d week seq
  from
                                                                  from date dim
   (select ss ticket number
                                                            where d date in ('1999-06-
                                                   14','1999-08-26','1999-11-06')))
         ,ss customer sk
                                                    and cr returned date sk = d date sk
          ,store.s city
                                                    group by i item id),
          , sum (ss coupon amt) amt
          ,sum(ss net profit) profit
                                                    wr items as
    from
                                                    (select i_item_id item_id,
store_sales,date_dim,store,household_demogr
                                                           sum(wr_return_quantity) wr_item_qty
                                                    from web returns,
    where store_sales.ss_sold_date_sk =
                                                         item,
date_dim.d_date_sk
                                                         date_dim
   and store_sales.ss_store_sk =
                                                    where wr_item_sk = i_item_sk
store.s store sk
                                                    and d date in
   and store_sales.ss_hdemo_sk =
                                                          (select d date
\verb|household_demographics.hd_demo_sk|
                                                           from date_dim
                                                           where d_week_seq in
(household demographics.hd dep count = 3 or
                                                                  (select d week seq
                                                                  from date dim
household_demographics.hd_vehicle_count >
                                                                  where d date in ('1999-06-
    and date dim.d dow = 1
                                                   14','1999-08-26','1999-11-06')))
    and date_dim.d_year in
                                                    and wr returned date sk = d date sk
(2000,2000+1,2000+2)
                                                    group by i_item_id)
                                                     select sr_items.item_id
    and store.s number employees between
200 and 295
                                                          ,sr item qty
                                                          ,cast(sr item qty as
    group by
ss ticket number,ss_customer_sk,ss_addr_sk,
                                                   double) / (cast(sr_item_qty+cr_item_qty+wr_it
store.s city) ms, customer
                                                   em qty as double))/3.0 * 100 sr dev
    where ss customer sk = c customer sk
                                                          ,cr item qty
order by
                                                          ,cast(cr_item_qty as
c last name, c first name, substr(s city, 1, 30
                                                   double) / (cast(sr_item_qty+cr_item_qty+wr_it
), profit
                                                   em_qty as double))/3.0 * 100 cr_dev
fetch first 100 rows only;
                                                          ,wr_item_qty
                                                          ,cast(wr_item_qty as
-- end query 89 in stream 0 using template
                                                   double)/(cast(sr item qty+cr item qty+wr it
query79.tpl
                                                   em qty as double))/3.0 * 100 wr dev
-- start query 96 in stream 0 using
                                                   ,(sr_item_qty+cr_item_qty+wr_item_qty)/3.0
template query83.tpl and seed 593789178
with sr items as
                                                    from sr items
                                                        cr items,
 (select i item id item id,
        sum(sr return quantity) sr item qty
                                                        ,wr items
 from store returns,
                                                    where sr_items.item_id=cr_items.item_id
     item,
                                                      and sr_items.item_id=wr_items.item_id
```

```
order by sr items.item id
                                                    and s qmt offset = -7
                                                     and d year = 1999
        sr item qty
  fetch first 100 rows only;
                                                     and d moy = 12) all sales
                                                  order by promotions, total
-- end query 96 in stream 0 using template
                                                   fetch first 100 rows only;
-- start query 97 in stream 0 using
                                                  -- end query 97 in stream 0 using template
template query61.tpl and seed 1770420976
                                                  query61.tpl
select promotions, total, cast (promotions as
                                                  -- start query 99 in stream 0 using
decimal(15,4))/cast(total as
decimal(15,4))*100
                                                  template query76.tpl and seed 945056756
                                                   select channel, col name, d year, d qoy,
  (select sum(ss_ext_sales_price)
                                                   i_category, COUNT(*) sales_cnt,
                                                  SUM(ext_sales_price) sales_amt FROM (
promotions
  from store_sales
                                                          SELECT 'store' as channel,
                                                   'ss hdemo sk' col name, d year, d qoy,
       ,store
       ,promotion
                                                  i_category, ss_ext_sales_price
        ,date_dim
                                                  ext_sales_price
        ,customer
                                                           FROM store_sales, item, date_dim
                                                           WHERE ss hdemo sk IS NULL
        , customer address
                                                             AND ss sold date sk=d date sk
   where ss sold date sk = d date sk
                                                             AND ss item sk=i item sk
   and ss store sk = s store sk
                                                          UNION ALL
                                                          SELECT 'web' as channel,
   and ss_promo_sk = p_promo_sk
                                                  'ws_web_page_sk' col_name, d_year, d_qoy,
   and ss_customer_sk= c_customer_sk
        ca address sk = c current addr sk
                                                  i category, ws ext sales price
        ss item sk = i item sk
                                                  ext sales price
   and ca_gmt_offset = -7
                                                           FROM web sales, item, date dim
   and i_category = 'Electronics'
                                                           WHERE ws web page sk IS NULL
   and (p channel dmail = 'Y' or
                                                             AND ws sold date sk=d date sk
p_channel_email = 'Y' or p_channel_tv =
                                                             AND ws_item_sk=i_item_sk
'Y')
                                                          UNION ALL
   and
        s gmt offset = -7
                                                          SELECT 'catalog' as channel,
       _
d_year = 1999
   and
                                                   'cs_ship_addr_sk' col_name, d_year, d_qoy,
   and d_moy = 12) promotional_sales,
                                                  i_category, cs_ext_sales_price
  (select sum(ss ext sales_price) total
                                                  ext_sales price
   from store sales
                                                           FROM catalog sales, item, date dim
       ,store
                                                           WHERE cs_ship_addr_sk IS NULL
                                                             AND cs_sold_date_sk=d_date_sk
        ,date_dim
        , customer
                                                             AND cs_item_sk=i_item_sk) foo
        ,customer address
                                                  GROUP BY channel, col name, d year, d qoy,
        ,item
                                                  i category
                                                  ORDER BY channel, col_name, d_year, d_qoy,
   where ss sold date sk = d date sk
   and ss store sk = s store sk
                                                  i category
   and ss customer sk= c customer sk
                                                   fetch first 100 rows only;
   and ca_address_sk = c_current_addr sk
   and
        ss_item_sk = i_item_sk
                                                  -- end query 99 in stream 0 using template
        ca gmt offset = -7
                                                  query76.tpl
   and i category = 'Electronics'
```

E.2 Impala Queries:

```
-- start query 1 in stream 0 using template
                                                     order by count(*)
query96.tpl and seed 550831069
                                                      limit 100;
select count(*)
from store sales
                                                     -- end query 1 in stream 0 using template
    , household\_demographics
                                                     query96.tpl
                                                     -- start query 2 in stream 0 using template
    ,time_dim, store
where ss sold time sk = time dim.t time sk
                                                     query7.tpl and seed 997258328
   and ss_hdemo_sk =
                                                     select i item id,
                                                             avg(ss_quantity) agg1,
avg(ss_list_price) agg2,
household_demographics.hd demo sk
    and ss_store_sk = s_store_sk
    and time_dim.t_hour = 15
                                                             avg(ss_coupon_amt) agg3,
    and time_dim.t_minute >= 30
                                                             avg(ss sales price) agg4
                                                     from store sales, customer_demographics,
    and household demographics.hd dep count
                                                     date_dim, item, promotion
    and store.s store name = 'ese'
                                                      where ss sold date sk = d date sk and
```

```
,warehouse
       ss item sk = i item sk and
       ss_cdemo_sk = cd_demo_sk and
                                                               ,date_dim
       ss promo sk = p promo sk and
                                                           where inv item sk = i item sk
       cd gender = 'M' and
                                                             and inv_warehouse_sk =
                                                    cd marital status = 'W' and
       cd_education_status = '2 yr Degree'
                                                             and d_year =2000
       (p channel email = 'N' or
                                                           group by
p channel event = 'N') and
                                                    w warehouse name, w warehouse sk, i item sk, d
       d_year = 1999
                                                    _moy) foo
 group by i_item_id
                                                     where case mean when {\tt 0} then {\tt 0} else
 order by i_item_id
                                                     stdev/mean end > 1)
  limit 100;
                                                     select
                                                    inv1.w warehouse sk,inv1.i_item_sk,inv1.d_m
                                                    oy, inv1.mean, inv1.cov
-- end query 2 in stream 0 using template
-- start query 5 in stream 0 using template
                                                     ,inv2.w warehouse sk,inv2.i item sk,inv2.d
query39.tpl and seed 1420791654
                                                    moy, inv2.mean, inv2.cov
                                                    from inv inv1, inv inv2
with inv as
(select
                                                    where inv1.i_item_sk = inv2.i_item_sk
w_warehouse_name,w_warehouse_sk,i_item_sk,d
                                                      and inv1.w_warehouse_sk =
_moy
                                                     inv2.w warehouse sk
       ,stdev,mean, case mean when 0 then
                                                      and inv1.d moy=2
null else stdev/mean end cov
                                                      and inv2.d moy=2+1
                                                      and inv1.cov > 1.5
from(select
w warehouse name, w warehouse sk, i item sk, d
                                                    order by
                                                    inv1.w warehouse_sk,inv1.i_item_sk,inv1.d_m
moy
                                                    oy, inv1.mean, inv1.cov
,cast(stddev_samp(inv_quantity_on_hand) as
                                                            ,inv2.d moy,inv2.mean, inv2.cov
double) stdev, avg (inv quantity on hand)
mean
      from inventory
                                                     -- end query 5 in stream 0 using template
          ,item
                                                    query39.tpl
          ,warehouse
                                                     -- start query 7 in stream 0 using template
          ,date dim
                                                    query32.tpl and seed 944563352
                                                    select sum(cs_ext_discount_amt) as "excess
discount amount"
      where inv_item_sk = i_item_sk
        and inv warehouse sk =
w warehouse_sk
                                                     from
        and inv date sk = d date sk
                                                        catalog sales
                                                       ,item
        and d_{year} = 2000
      group by
                                                        ,date_dim
w_warehouse_name,w_warehouse_sk,i_item_sk,d
_moy) foo
                                                              select
where case mean when 0 then 0 else
                                                                 cs item sk,
stdev/mean end > 1)
                                                                 1.3 * avg(cs_ext_discount_amt)
select
                                                    avg_cs_ext_discount_amt130
inv1.w warehouse sk,inv1.i item sk,inv1.d m
                                                             from
oy, inv1.mean, inv1.cov
                                                                catalog sales
                                                                ,date dim
,inv2.w warehouse sk,inv2.i item sk,inv2.d
                                                              where
moy, inv2.mean, inv2.cov
                                                               d_{date} between '2000-01-16' and
from inv inv1, inv inv2
                                                                              (cast('2000-01-16'
where inv1.i\_item\_sk = inv2.i\_item\_sk
                                                    as timestamp) + interval 90 days)
                                                              and d_date_sk = cs_sold_date_sk
  and inv1.w_warehouse_sk =
inv2.w warehouse sk
                                                               group by cs_item_sk
  and \overline{inv1.d} moy=2
                                                          ) tmp1
 and inv2.d moy=2+1
                                                     where
order by
                                                     i_manufact_id = 353
inv1.w_warehouse_sk,inv1.i_item_sk,inv1.d_m
                                                     and i_item_sk = catalog_sales.cs_item_sk
oy, inv1.mean, inv\overline{1.cov}
                                                     and d date between '2000-01-16' and
                                                            (cast('2000-01-16' as timestamp) +
        ,inv2.d moy,inv2.mean, inv2.cov
                                                     interval 90 days)
with inv as
                                                     and d_date_sk = cs_sold_date_sk
                                                     and tmp1.cs_item_sk = i_item_sk
(select
w warehouse name, w warehouse sk, i item sk, d
                                                    and cs_ext_discount_amt
                                                    avg cs ext discount_amt130
                                                     limit 100;
       ,stdev,mean, case mean when 0 then
null else stdev/mean end cov
from(select
                                                     -- end query 7 in stream 0 using template
w_warehouse_name, w_warehouse_sk,i_item_sk,d
                                                    querv32.tpl
                                                     -- start query 14 in stream 0 using
                                                     template query21.tpl and seed 614834996
,cast(stddev samp(inv quantity on hand) as
                                                     select *
double) stdev, avg (inv quantity on hand)
                                                     from(select w warehouse name
                                                                 ,i\_item id
                                                     , sum(case when (d_date < cast ('1998-06-27' as timestamp))
      from inventory
          ,item
```

then	s state, 0 as g state,
inv_quantity_on_hand	avg(ss_quantity) agg1,
else 0 end) as	avg(ss_list_price) agg2,
inv_before	avg(ss_coupon_amt) agg3,
, sum(case when (d_date >= cast	avg(ss_sales_price) agg4
('1998-06-27' as timestamp))	from store_sales, customer_demographics,
then	date_dim, store, item
inv_quantity_on_hand	where ss_sold_date_sk = d_date_sk and
else 0 end) as	ss_item_sk = i_item_sk and
inv_after	ss_store_sk = s_store_sk and
from inventory	ss_cdemo_sk = cd_demo_sk and
,warehouse	cd_gender = 'F' and
,item	cd_marital_status = 'W' and
,date_dim	cd_education_status = '4 yr Degree'
where i_current_price between 0.99 and	and
1.49	d_year = 1999 and
and i_item_sk = inv_item_sk	s_state in ('OH','IL', 'LA', 'GA',
and inv_warehouse_sk =	'CO', 'AL')
w_warehouse_sk	group by i_item_id, s_state)
and inv_date_sk = d_date_sk	
and d_date between (cast ('1998-06-27'	select i_item_id,
as timestamp) - interval 30 days)	s_state, g_state, agg1, agg2, agg3, agg4
and (cast ('1998-06-27'	from (
as timestamp) + interval 30 days)	select i_item_id, s_state, g_state
group by w warehouse name, i item id) x	agg1, agg2, agg3, agg4 from results
where (case when inv_before $> \overline{0}$	union
then inv_after / inv_before	select i_item_id, NULL AS s_state,
else null	1 AS g_state, avg(agg1) agg1, avg(agg2)
end) between 2.0/3.0 and	agg2, avg(agg3) agg3,
3.0/2.0	avg(agg4) agg4 from results
order by w warehouse name	group by i item id
,i item id	union
limit 100;	select NULL AS i item id, NULL as
	s state, 1 as g state, avg(agg1) agg1,
end query 14 in stream 0 using template	avg(agg2) agg2, avg(agg3) agg3,
query21.tpl	avg(agg4) agg4 from results
start query 15 in stream 0 using) foo
template query43.tpl and seed 959608359	order by i item id, s state
select s store name, s store id,	limit 100;
sum(case when (d day name='Sunday')	1110 1007
then ss_sales_price else null end)	end query 16 in stream 0 using template
sun sales,	query27.tpl
sum(case when (d day name='Monday')	start query 19 in stream 0 using
then ss sales price else null end)	template query58.tpl and seed 1844319395
mon sales,	with ss items as
sum(case when	(select i item id item id
(d day name='Tuesday') then ss sales price	, sum (ss ext sales price)
else null end) tue sales,	ss item rev
sum(case when	from store sales
(d day name='Wednesday') then	,item
ss_sales_price else null end) wed_sales,	,date dim
sum(case when	JOIN (select dl.d date
	from date dim d1
<pre>(d_day_name='Thursday') then ss_sales_price else null end) thu sales,</pre>	JOIN date dim d2
sum(case when (d_day_name='Friday')	-
	ON d1.d_week_seq =
then ss_sales_price else null end)	d2.d_week_seq
fri_sales,	where d2.d_date = '1998-05-29')
sum(case when	sub
(d_day_name='Saturday') then ss_sales_price	ON date_dim.d_date = sub.d_date
else null end) sat_sales	where ss_item_sk = i_item_sk
from date_dim, store_sales, store	and ss_sold_date_sk = d_date_sk
where d_date_sk = ss_sold_date_sk and	group by i_item_id),
s_store_sk = ss_store_sk and	cs_items as
$s_gmt_offset = -8$ and	(select i_item_id item_id
d_year = 1998	<pre>, sum(cs_ext_sales_price)</pre>
group by s_store_name, s_store_id	cs_item_rev
order by s_store_name,	from catalog_sales
s_store_id, sun_sales, mon_sales, tue_sales, we	,item
d_sales,thu_sales,fri_sales,sat_sales	,date_dim
limit 100;	JOIN (select dl.d_date
	from date_dim d1
end query 15 in stream 0 using template	JOIN date_dim d2
query43.tpl	ON dl.d_week_seq =
start query 16 in stream 0 using	d2.d_week_seq
template query27.tpl and seed 331218716	where d2.d_date = '1998-05-29')
with results as	sub
(select i item id,	ON date dim.d date = sub.d date

```
where cs item sk = i item sk
                                                              ) ssi on item.i manufact id =
 and cs_sold_date_sk = d_date_sk
                                                     ssi.i manufact id
 group by i item id),
                                                      where ss item sk
                                                      i item sk
 ws items as
 (select i item id item id
                                                       and
                                                               ss_sold_date_sk
       , sum (ws_ext_sales_price)
                                                      d_date_sk
ws item_rev
                                                           d_year
                                                                                         = 2001
                                                       and
                                                                                         = 6
  from web sales
                                                       and
                                                               d moy
     ,item
                                                      and
                                                               ss addr sk
      ,date dim
                                                      ca address_sk
     JOIN (select dl.d date
                                                             ca_gmt_offset
                                                      and
                                                                                         = -7
           from date_dim d1
                                                       group by item.i manufact id),
               JOIN date dim d2
                                                       cs as (
                   ON d1.d week seq =
                                                      select.
d2.d week seq
           where d2.d_date = '1998-05-29')
                                                      item.i_manufact_id,sum(cs_ext_sales_price)
                                                      total sales
         ON date dim.d date = sub.d date
                                                       from
where ws_item_sk = i_item_sk
                                                              catalog sales,
 and ws sold date sk = \overline{d} date sk
                                                              date dim,
 group by i_item_id)
                                                               customer address,
 select ss items.item id
                                                               item
       ,ss item rev
                                                              left semi join (
                                                                               select
                                                      i manufact id
,ss item rev/(ss item rev+cs item rev+ws it
em \overline{rev})/3 * 100 \overline{s}s \overline{dev}
                                                                               from item
       ,cs item rev
                                                                               where i category in
                                                      ('Books')
,cs_item_rev/(ss_item_rev+cs_item_rev+ws_it
                                                              ) csi on item.i_manufact_id =
em \overline{rev})/3 * 100 cs dev
                                                      csi.i manufact id
       ,ws item rev
                                                       where cs item sk
                                                      i item_sk
,ws_item_rev/(ss_item_rev+cs_item_rev+ws_it
                                                      and
                                                               cs_sold_date_sk
em \overline{rev})/3 * 100 ws dev
                                                      d date sk
                                                       and
                                                              d_year
                                                                                         = 2001
,(ss_item_rev+cs_item_rev+ws_item_rev)/3
                                                       and
                                                               d moy
                                                                                         = 6
                                                               cs bill_addr_sk
average
                                                       and
 from ss items, cs items, ws items
                                                      ca address sk
where ss items.item id=cs items.item id
                                                      and
                                                              ca gmt offset
   and ss_items.item_id=ws_items.item_id
                                                      group by item.i_manufact_id),
   and ss_item_rev between 0.9 *
                                                       ws as (
cs item rev and 1.1 * cs item rev
                                                       select
   and ss item rev between 0.9 *
ws item \overline{rev} and 1.1 * ws item \overline{rev}
                                                      item.i manufact id, sum(ws ext sales price)
   and cs item rev between 0.9 ^{\star}
                                                      total \overline{\text{sales}}
ss_item_rev and 1.1 * ss_item_rev
                                                       from
   and cs item rev between 0.9 *
                                                              web sales,
ws item \overline{rev} and 1.1 * ws item \overline{rev}
                                                              date dim,
   and ws_item_rev between 0.\overline{9} *
                                                               customer address,
ss item rev and 1.1 * ss item rev
                                                               item
   and ws_item_rev between 0.9 *
                                                              left semi join (
cs item rev and 1.1 * cs item rev
                                                                               select
order by item_id
                                                      i manufact id
         ,ss_item_rev
                                                                               from item
 limit 100;
                                                                               where i category in
                                                      ('Books')
                                                              ) wsi on item.i manufact id =
-- end query 19 in stream 0 using template
                                                      wsi.i manufact id
query58.tpl
-- start query 22 in stream 0 using
                                                       where ws_item_sk
template query33.tpl and seed 248487088
                                                      i item sk
with ss as (
                                                       and
                                                               ws sold date sk
select
                                                      d date sk
                                                                                         = 2001
                                                       and
                                                               d_year
item.i_manufact_id,sum(ss_ext_sales_price)
                                                       and
                                                               d moy
                                                               ws bill addr sk
                                                       and
                                                      ca address sk
 from
                                                              ca_gmt_offset
        store sales,
                                                       and
        date \overline{\text{dim}},
                                                       group by item.i_manufact_id)
         customer address,
                                                       select i_manufact_id ,sum(total_sales)
                                                      total_sales
                                                       from (select * from ss
        left semi join (
                                                              union all
                         select
i manufact id
                                                              select * from cs
                         from item
                                                              union all
                                                              select * from ws) tmp1
                         where i_category in
('Books')
                                                       group by i_manufact_id
                                                       order by total sales
```

```
limit 100;
                                                     'exportiunivamalg #9','scholaramalgamalg
-- end query 22 in stream 0 using template
                                                     #9'))
query33.tpl
                                                                      i category in
                                                     ('Women','Music','Men')
-- start query 24 in stream 0 using
template query62.tpl and seed 800775315
                                                                  and i class in
                                                     ('accessories','classical','fragrances','pa
select
  substr(w warehouse name, 1, 20)
                                                     nts')
 ,sm type
                                                                   and i brand in ('amalgimporto
                                                     #1','edu packscholar #1','exportiimporto
 ,web name
  ,sum(case when (ws_ship_date_sk -
                                                     #1',
ws_sold_date_sk <= 30 ) then 1 else 0 end)
                                                     'importoamalg #1')))
as "30 davs"
, sum(case when (ws ship date sk -
ws_sold_date_sk > 30) and
                                                     group by i manager id, d moy),
                 (ws ship date sk -
ws sold date sk \leq 60) then 1 else 0 end )
                                                     cte2 as
as "31-60 days"
                                                     (
 ,sum(case when (ws ship date sk -
                                                     select.
ws_sold_date_sk > 60) and
                                                       i manager id,
                 (ws_ship_date_sk -
                                                       avg(sum_sales) avg_monthly_sales
ws sold date sk \leq 90) then 1 else 0 end)
                                                     from ctel
as "61-90 days"
                                                      aroup by
 ,sum(case when (ws_ship_date_sk -
                                                       i manager id)
ws_sold_date_sk > 90) and
                 (ws ship date sk -
                                                      select
ws sold date sk \leq 120) then 1 else 0 end)
                                                      cte1.i_manager_id
as "91-120 days"
                                                       ,cte1.sum_sales
 ,sum(case when (ws_ship_date_sk -
                                                       ,cte2.avg_monthly_sales
ws sold date sk > 1\overline{20}) then 1 else 0 end)
                                                     from cte1
as ">120 days"
                                                         cross join cte2
from
                                                     where
  web_sales
                                                       (cte1.i_manager_id = cte2.i_manager_id
 ,warehouse
 ,ship_mode
                                                       (ctel.i\_manager\_id is NULL and
 ,web_site
                                                     cte2.i_manager_id is NULL))
                                                       and case when avg_monthly_sales > 0 then
  ,date dim
                                                     abs (sum sales - avg monthly sales) /
    d month seq between 1201 and 1201 + 11
                                                     avg monthly sales else null end > 0.1
and ws_ship_date_sk = d_date_sk
                                                     order by ctel.i manager id
                      = w_warehouse_sk
                                                             ,avg_monthly_sales
and ws_warehouse_sk
                      = sm_ship_mode_sk
and ws_ship_mode_sk
                                                              sum sales
and ws web site sk
                      = web_site_sk
                                                     limit 100;
group by
                                                     -- end query 27 in stream 0 using template
  substr(w_warehouse_name,1,20)
 ,sm_type
                                                     query63.tpl
  ,web name
                                                      -- start query 28 in stream 0 using
order by substr(w warehouse name, 1, 20)
                                                     template query69.tpl and seed 1390437346
        ,sm type
                                                     select.
       ,web name
                                                       cd gender,
limit 100;
                                                       cd marital status,
                                                       cd education status,
                                                       count(*) cnt1,
-- end query 24 in stream 0 using template
querv62.tpl
                                                       cd_purchase_estimate,
-- start query 27 in stream 0 using
                                                       count(*) cnt2,
template query63.tpl and seed 812633773
                                                       cd credit rating,
with ctel as
                                                       count(*) cnt3
(select i_manager_id
                                                      from
             ,sum(ss sales price) sum sales
                                                       customer c, customer address
                                                     ca, customer_demographics
  left semi join
      from item
          ,store sales
          ,date_dim
                                                              (select ss_customer_sk as
          ,store
                                                     customer_sk
      where ss_item_sk = i_item_sk
                                                               from store_sales,date_dim
        and ss sold date sk = d date sk
                                                               where store_sales.ss_sold_date_sk
        and ss store_sk = s_store_sk
                                                     = date_dim.d date sk
        and d_month_seq in
                                                               and d_year = 2004 and
                                                                     d_moy between 3 and 3+2) ss
(1178, 1178+1, \overline{1}178+2, \overline{1}178+3, 1178+4, 1178+5, 11
78+6,1178+7,1178+8,1178+9,1178+10,1178+11)
                                                           on c.c_customer_sk = ss.customer_sk
and (( i_category in
('Books','Children','Electronics')
                                                       left outer join
                                                                (select ws_bill_customer_sk as
              and i class in
                                                     customer_sk
('personal', 'portable', 'refernece', 'self-
                                                                 from web sales, date dim
help')
                                                                 where web_sales.ws_sold_date_sk
              and i_brand in
                                                     = date_dim.d_date_sk
                                                                 and d_year = 2004 and
('scholaramalgamalg #14','scholaramalgamalg
```

d_moy between 3 and 3+2)	where i_category
WS	in ('Jewelry')
on c.c_customer_sk = ws.customer_sk) ics on item.i_item_id =
left outer join (select cs ship customer sk as	ics.i_item_id where cs item sk =
customer_sk	i_item_sk
from catalog_sales,date_dim	and cs_sold_date_sk =
where	d_date_sk
<pre>catalog_sales.cs_sold_date_sk = date dim.d date sk</pre>	and $d_year = 2002$ and $d_year = 10$
and d year = 2004 and	and cs_bill_addr_sk =
<pre>d_moy between 3 and 3+2)</pre>	ca_address_sk
CS	and ca_gmt_offset = -5
on c.c_customer_sk = cs.customer_sk where	group by i_item_id), ws as (
c.c_current_addr_sk = ca.ca_address_sk	select
and	
<pre>ca_state in ('AL','VA','GA') and cd demo sk = c.c current cdemo sk and</pre>	item.i_item_id,sum(ws_ext_sales_price)
ws.customer sk is NULL and	total_sales from
cs.customer_sk is NULL	web_sales,
group by cd_gender,	date_dim,
cd_marital_status,	customer_address,
<pre>cd_education_status, cd purchase estimate,</pre>	item left semi join (
cd_credit_rating	select i_item_id
order by cd_gender,	from item
cd_marital_status,	where i_category i ('Jewelry')
cd_education_status, cd purchase estimate,) iws on item.i item id =
cd_credit_rating	iws.i_item_id
limit 100;	where ws_item_sk =
end query 28 in stream 0 using template	i_item_sk and ws sold date sk =
query69.tpl	and ws_sold_date_sk = d date sk
start query 29 in stream 0 using	$\frac{1}{2}$ and $\frac{1}{2}$
template query60.tpl and seed 374071684	and $d_{moy} = 10$
with ss as (select	and ws_bill_addr_sk = ca address sk
SCICCO	and ca gmt offset = -5
<pre>item.i_item_id,sum(ss_ext_sales_price)</pre>	group by i_item_id)
total_sales	select
from store sales,	i_item_id ,sum(total sales) total sales
date dim,	from (select * from ss
customer_address,	union all
item	select * from cs
left semi join (select i item id	union all select * from ws) tmp1
from item	group by i item id
where i_category	order by i_item_id
<pre>in ('Jewelry')</pre>	,total_sales limit 100;
iss.i item id	TIMEC 100,
where ss_item_sk =	end query 29 in stream 0 using template
i_item_sk	query60.tpl
and ss_sold_date_sk = d date sk	start query 30 in stream 0 using template query59.tpl and seed 1976435349
and d year $= 2002$	with wss as
and $d_{moy} = 10$	(select d_week_seq,
and ss_addr_sk =	ss_store_sk,
<pre>ca_address_sk and ca gmt offset = -5</pre>	<pre>sum(case when (d_day_name='Sunday' then ss_sales_price else null end)</pre>
group by i item id),	sun sales,
cs as (sum(case when (d_day_name='Monday'
select	then ss_sales_price else null end)
<pre>item.i item id, sum(cs ext sales price)</pre>	mon_sales, sum(case when
total_sales	(d_day_name='Tuesday') then ss_sales_price
from	else null end) tue_sales,
catalog_sales,	sum(case when
<pre>date_dim, customer address,</pre>	<pre>(d_day_name='Wednesday') then ss_sales_price else null end) wed_sales,</pre>
item	sum(case when
left semi join ((d_day_name='Thursday') then ss_sales_pric
select i_item_id from item	else null end) thu_sales,
TTOM TOTAL	

```
sum(case when (d day name='Friday')
                                                              and ss sold date sk = d date sk
                                                             and d date between cast('1999-05-
then ss sales price else null end)
                                                      25' as timestamp)
fri sales,
                                                                     and (cast('1999-05-25' as
        sum(case when
(d_day_name='Saturday') then ss_sales_price
                                                     timestamp) + interval 30 days)
else null end) sat sales
                                                         group by
 from store_sales,date_dim
                                                             i item id
where d date sk = ss sold date sk
                                                              ,i item desc
                                                              ,i_category
group by d week seq, ss store sk
                                                              ,i_class
 select
                                                              ,i_current_price ),
s_store_name1,s_store_id1,d_week_seq1
                                                      cte2 as
                                                      ( select
,sun sales1/sun sales2,mon sales1/mon sales
                                                          i class
                                                          , sum (itemrevenue) as sumitemrevenue
                                                        from cte1
,tue sales1/tue sales1,wed sales1/wed sales
                                                        group by
2, thu sales1/thu sales2
                                                         i class)
                                                      select
,fri sales1/fri sales2,sat sales1/sat sales
                                                         i item desc
                                                         ,i_category
                                                         ,i_class
                                                         ,i_current price
 (select s store name
                                                         ,itemrevenue
s_store_name1,wss.d_week_seq d_week_seq1
        ,s store id s store id1, sun sales
                                                         ,revenueratio
sun sales1
                                                      from (
        ,mon sales mon sales1, tue sales
                                                      select
tue_sales1
                                                         cte1.i_item_id
        ,wed sales wed sales1,thu sales
                                                         ,cte1.i_item_desc
thu sales1
                                                        ,ctel.i category
                                                         ,cte1.i_class
,cte1.i_current_price
        ,fri sales fri sales1,sat sales
sat sales1
  from wss, store, date_dim d
                                                         ,cte1.itemrevenue
  where d.d_week_seq = wss.d_week_seq and
    ss store sk = s store sk and
                                                      ,ctel.itemrevenue*100/cte2.sumitemrevenue
        d_{month\_seq} between 1\overline{1}89 and 1189 +
                                                      as revenueratio
                                                      from ctel
 (select s store name
                                                          cross join cte2
s store name2, wss.d week seq d week seq2
                                                      where
        ,s_store_id s_store_id2,sun_sales
                                                        ctel.i class = cte2.i class or
                                                      (ctel.i_class is NULL and cte2.i_class is
sun sales2
                                                     NULL)
        ,mon sales mon sales2, tue sales
tue sales2
                                                     ) v1
        ,wed sales wed sales2,thu sales
                                                     order by
thu sales2
                                                        i category
        ,fri_sales fri_sales2,sat_sales
                                                        ,i_class
                                                        ,i_item_id
,i_item_desc
sat sales2
 from wss, store, date dim d
 where d.d_week_seq = wss.d_week_seq and
    ss_store_sk = s_store_sk and
                                                         ,revenueratio
        d_month_seq between 1189+ 12 and
1189 + 23) x
                                                      -- end query 32 in stream 0 using template
where s store id1=s store id2
                                                     query98.tpl
                                                      -- start query 36 in stream 0 using
  and d_week_seq1=d_week_seq2-52
order by
                                                      template query28.tpl and seed 24799953
s store name1,s store id1,d week seq1
limit 100;
                                                      from (select avg(ss list price) B1 LP
                                                                  ,count(ss_list_price) B1_CNT
-- end query 30 in stream 0 using template
                                                                  ,count(distinct ss_list_price)
                                                      B1 CNTD
-- start query 32 in stream 0 using
                                                            from store sales
                                                            where ss\_quantity between 0 and 5
template query98.tpl and seed 1900673199
                                                              and (ss\_list\_price\ between\ 47\ and
with ctel as
( select
                                                      47 + 10
    i item id
                                                                   or ss coupon amt between 11713
   ,i_item desc
                                                     and 11713+1000
                                                                   or ss_wholesale_cost between
    ,i_category
                                                      55 and 55+20)) B1
    ,i_class
    ,i_current_price
                                                      cross join
    , sum(ss_ext_sales_price) as itemrevenue
                                                           (select avg(ss_list_price) B2_LP
                                                                  ,count(ss_list_price) B2_CNT
    from store_sales
        ,item
                                                                  ,count(distinct ss list price)
                                                      B2 CNTD
         ,date dim
                                                            from store sales
        ss_item_sk = i_item_sk
                                                            where ss_quantity between 6 and 10
       and i_category in ('Shoes',
                                                              and (ss_list_price between 93 and
'Music', 'Home')
                                                      93+10
```

```
or ss_coupon amt between 7733 and
                                                           ,sum(jul_sales) as jul sales
7733+1000
                                                           ,sum(aug_sales) as aug_sales
                                                           ,sum(sep_sales) as sep sales
          or ss wholesale cost between 43
and 43+20)) B2
                                                           ,sum(oct sales) as oct sales
cross join
                                                           ,sum(nov_sales) as nov_sales
                                                           ,sum(dec_sales) as dec sales
    (select avg(ss_list_price) B3_LP
           ,count(ss_list_price) B3_CNT
                                                           ,sum(jan_sales/w_warehouse_sq_ft)
            ,count(distinct ss list price)
                                                   as jan sales per sq foot
                                                          ____,sum(feb sales/w warehouse sq ft)
B3 CNTD
                                                   as feb_sales_per_sq_foot
      from store sales
      where ss_quantity between 11 and 15
                                                           , sum (mar_sales/w_warehouse_sq_ft)
        and (ss_list_price between 32 and
                                                   as mar_sales_per_sq_foot
                                                   , sum(apr_sales/w_warehouse_sq_ft)
as apr_sales_per_sq_foot
          or ss\_coupon\_amt between 11517
and 11517+1000
                                                           ,sum(may_sales/w_warehouse_sq_ft)
         or ss_wholesale_cost between 26
                                                   as may_sales_per_sq_foot
and 26+20)) B3
                                                           ,sum(jun sales/w warehouse sq ft)
                                                   cross join
     (select avg(ss_list_price) B4_LP
           ,count(ss_list_price) B4_CNT
                                                    as jul_sales_per_sq_foot
            ,count(distinct ss_list_price)
                                                           ,sum(aug_sales/w_warehouse_sq_ft)
                                                    as aug_sales_per_sq foot
B4 CNTD
                                                           ,sum(sep_sales/w_warehouse_sq_ft)
      from store_sales
                                                    as sep_sales_per_sq_foot
      where ss\_quantity between 16 and 20
        and (ss = 1) list price between 147 and
                                                           ,sum(oct sales/w warehouse sq ft)
                                                   or ss_coupon_amt between 509 and
509+1000
                                                   as nov_sales_per_sq_foot
          or ss_wholesale_cost between 78
                                                           ,sum(dec_sales/w_warehouse_sq_ft)
and 78+20)) B4
                                                   as dec sales per sq foot
                                                          ,sum(jan_net) as jan_net
,sum(feb_net) as feb_net
cross join
     (select avg(ss list price) B5 LP
           ,count(ss_list_price) B5_CNT
                                                           ,sum(mar_net) as mar_net
            ,count(distinct ss_list_price)
                                                           ,sum(apr_net) as apr net
B5 CNTD
                                                           ,sum(may_net) as may_net
      from store_sales
                                                           ,sum(jun_net) as jun_net
      where ss\_quantity between 21 and 25
                                                           ,sum(jul net) as jul net
       and (ss list price between 16 and
                                                           ,sum(aug net) as aug net
                                                           ,sum(sep_net) as sep_net
          or ss\_coupon\_amt between 2401 and
                                                           ,sum(oct_net) as oct_net
2401+1000
                                                           ,sum(nov_net) as nov_net
         or ss_wholesale_cost between 32
                                                           ,sum(dec_net) as dec_net
and 32+20)) B5
                                                    from (
cross join
                                                       (select
     (select avg(ss_list_price) B6_LP
                                                          w warehouse name
           ,count(ss_list_price) B6 CNT
                                                           ,w_warehouse_sq_ft
            ,count(distinct ss list price)
                                                           ,w city
B6 CNTD
                                                           ,w county
      from store sales
                                                           ,w_state
      where ss\_quantity between 26 and 30
                                                           ,w_country
                                                           ,concat('MSC', ',', 'USPS') as
        and (ss\_list\_price\ between\ 11\ and
11+10
                                                    ship carriers
         or ss_coupon amt between 916 and
                                                          ,d_year as year
                                                           , sum(case when d moy = 1)
916+1000
          or ss wholesale cost between 6
                                                                  then ws sales price*
and 6+20)) B6
                                                    ws quantity else 0 end) as jan sales
limit 100;
                                                          , sum (case when d_{moy} = 2
                                                                 then ws_sales_price*
-- end query 36 in stream 0 using template
                                                   ws quantity else 0 end) as feb sales
                                                           ,sum(case when d moy =
 - start query 39 in stream 0 using
                                                                  then ws sales price*
template query66.tpl and seed 1688498284
                                                   ws quantity else 0 end) as mar sales
                                                           , sum (case when d_{moy} = 4
select
         w warehouse name
                                                                  then ws_sales_price*
                                                   ws quantity else 0 end) as apr sales
       ,w_warehouse_sq_ft
       ,w_city
                                                           , sum (case when d moy = 5
                                                                  then ws_sales_price*
       ,w_county
                                                   ws_quantity else 0 end) as may_sales
       ,w_state
       ,w country
                                                          ,sum(case when d_moy = 6
        ,ship_carriers
                                                   then ws_sales_price* ws_quantity else 0 end) as jun_sales
        ,year
       ,sum(jan_sales) as jan_sales
                                                           , sum (case when d moy = 7
       , sum (feb sales) as feb sales
                                                                  then ws sales price*
       , sum (mar sales) as mar sales
                                                   ws quantity else 0 end) as jul sales
                                                          , sum (case when d moy = 8
       ,sum(apr_sales) as apr_sales
                                                                  then ws_sales_price*
       , sum (may_sales) as may_sales
       ,sum(jun sales) as jun sales
                                                   ws quantity else 0 end) as aug sales
```

,sum(case when d moy = 9	group by
then ws sales price*	w warehouse name
ws quantity else 0 end) as sep sales	,w warehouse sq ft
, sum (case when d moy = 10	,w city
then ws sales price*	,w county
ws quantity else 0 end) as oct sales	,w state
,sum(case when d_moy = 11	,w_country
then ws_sales_price*	,d_year
ws_quantity else 0 end) as nov_sales)
<pre>, sum(case when d_moy = 12</pre>	union all
then ws_sales_price*	(select
ws_quantity else 0 end) as dec_sales	w_warehouse_name
<pre>, sum (case when d_moy = 1</pre>	,w_warehouse_sq_ft
then	,w_city
ws_net_paid_inc_ship_tax * ws_quantity else	,w_county
0 end) as jan_net	,w_state
<pre>, sum(case when d_moy = 2</pre>	<pre>,w_country ,concat('MSC', ',', 'USPS') as</pre>
ws net paid inc ship tax * ws quantity else	ship carriers
0 end) as feb net	,d year as year
, sum(case when d moy = 3	, sum(case when d moy = 1
then	then cs_ext_sales_price
ws_net_paid_inc_ship_tax * ws_quantity else	cs quantity else 0 end) as jan sales
0 end) as mar net	sum(case when d moy = 2
sum(case when d moy = 4	then cs ext sales price
then	cs quantity else 0 end) as feb sales
ws_net_paid_inc_ship_tax * ws_quantity else	,sum(case when d_moy = 3
0 end) as apr_net	then cs_ext_sales_price
<pre>, sum(case when d_moy = 5</pre>	cs_quantity else 0 end) as mar_sales
then	<pre>, sum(case when d_moy = 4</pre>
ws_net_paid_inc_ship_tax * ws_quantity else	then cs_ext_sales_price
0 end) as may_net	cs_quantity else 0 end) as apr_sales
,sum(case when d_moy = 6	sum(case when d_moy = 5
then	then cs_ext_sales_price
ws_net_paid_inc_ship_tax * ws_quantity else	cs_quantity else 0 end) as may_sales
0 end) as jun_net	, sum(case when d_moy = 6
<pre>, sum(case when d_moy = 7</pre>	then cs_ext_sales_price
ws net paid inc ship tax * ws quantity else	<pre>cs_quantity else 0 end) as jun_sales , sum(case when d moy = 7</pre>
0 end) as jul net	then cs_ext_sales_price
, sum(case when d moy = 8	cs quantity else 0 end) as jul sales
then	sum(case when d moy = 8
ws net paid inc ship tax * ws quantity else	then cs ext sales price
0 end) as aug net	cs quantity else 0 end) as aug sales
sum(case when d_moy = 9	<pre>, sum(case when d_moy = 9</pre>
then	then cs_ext_sales_price
ws_net_paid_inc_ship_tax * ws_quantity else	cs_quantity else 0 end) as sep_sales
0 end) as sep_net	$, sum(case when d_moy = 10)$
<pre>, sum(case when d_moy = 10</pre>	then cs_ext_sales_price
then	cs_quantity else 0 end) as oct_sales
ws_net_paid_inc_ship_tax * ws_quantity else	<pre>, sum(case when d_moy = 11</pre>
0 end) as oct_net	then cs_ext_sales_price
<pre>, sum(case when d_moy = 11</pre>	<pre>cs_quantity else 0 end) as nov_sales , sum(case when d moy = 12</pre>
then	then cs ext sales price
<pre>ws_net_paid_inc_ship_tax * ws_quantity else 0 end) as nov net</pre>	cs quantity else 0 end) as dec sales
, sum(case when d moy = 12	, sum (case when d moy = 1
then	then cs net profit *
ws net paid inc ship tax * ws quantity else	cs quantity else 0 end) as jan net
0 end) as dec net	sum(case when d moy = 2
from	then cs net profit *
web sales	cs quantity else 0 end) as feb net
,warehouse	, sum(case when d moy = 3
,date dim	then cs net profit *
,time_dim	cs_quantity else 0 end) as mar_net
,ship_mode	<pre>, sum(case when d_moy = 4</pre>
where	then cs_net_profit *
ws_warehouse_sk =	cs_quantity else 0 end) as apr_net
w_warehouse_sk	, sum(case when d_moy = 5
and ws_sold_date_sk = d_date_sk	then cs_net_profit *
and ws_sold_time_sk = t_time_sk	<pre>cs_quantity else 0 end) as may_net ,sum(case when d moy = 6</pre>
<pre>and ws_ship_mode_sk = sm ship mode sk</pre>	, sum(case when a_moy = 6 then cs_net_profit *
and d year = 2002	cs quantity else 0 end) as jun net
and t time between 18036 and	, sum(case when d moy = 7
18036+28800	then cs net profit *
and em carrier in ('MSC' 'HSDS')	ce quantity also (and) as jul nat

```
, sum(case when d moy = 8)
                                                             from web sales,
               then cs_net_profit *
                                                      household demographics , time dim, web page
cs quantity else 0 end) as aug net
                                                             where ws sold time sk =
       , sum (case when d moy = 9
                                                      time dim.t time sk
then cs_net_profit * cs_quantity else 0 end) as sep_net
                                                               and ws_ship_hdemo_sk =
                                                      household demographics.hd demo sk
        , sum (case when d moy = 10
                                                               and ws_web_page_sk =
               then cs net profit *
                                                      web page.wp web page sk
cs quantity else 0 end) as oct net
                                                               and time dim.t hour between 18 and
        , sum (case when d moy = \overline{11}
                                                      18+1
               then cs_net_profit *
cs quantity else 0 end) as nov_net
                                                      household_demographics.hd_dep_count = 9
        , sum (case when d moy = \frac{1}{12}
                                                                and web page.wp char count between
                                                      5000 and 5200) pt
              then cs_net_profit *
cs quantity else 0 end) as dec net
                                                       order by am_pm_ratio
                                                        limit 100;
          catalog_sales
         ,warehouse
                                                      -- end query 40 in stream 0 using template
         ,date_dim
                                                      query90.tpl
                                                      -- start query 44 in stream 0 using
         ,time_dim
         ,ship_mode
                                                      template query92.tpl and seed 643980925
     where
                                                      select
                                                         sum(ws ext discount amt) as "Excess
            cs_warehouse_sk =
w warehouse sk
                                                      Discount Amount"
        and cs sold date sk = d date sk
        and cs_sold_time_sk = t_time_sk and cs_ship_mode_sk =
                                                          web sales
                                                         ,item
                                                         ,date_dim
sm_ship_mode_sk
        and d_year = 2002
                                                         join(
        and t time between 18036 and
                                                               SELECT
18036+28800
                                                                  ws item sk, 1.3 *
        and sm_carrier in ('MSC','USPS')
                                                      avg(ws_ext discount amt)
                                                      avg_ws_ext_discount_amt_130
     group by
        w warehouse name
                                                               FROM
        ,w_warehouse_sq_ft
                                                                  web sales
        \tt ,w\_city
                                                                  ,date_dim
                                                               WHERE
        ,w county
        ,w state
                                                                     d date between '1999-01-03'
        ,w country
                                                      and
       ,d_year
                                                                                     (cast('1999-
                                                      01-03' as timestamp) + interval 90 days)
     )
 ) x
                                                                and d_date_sk = ws_sold_date_sk
                                                                GROUP BY ws item sk
 group by
                                                            ) wsd on item. i item sk =
        w warehouse name
        ,w_warehouse_sq_ft
                                                      wsd.ws_item_sk
        ,w_city
                                                      where
        ,w county
                                                      i manufact id = 926
                                                      and i_item_sk = web_sales.ws_item_sk and d_date between '1999-01-03' and
        ,w state
        ,w country
                                                              (cast('1999-01-03' as timestamp) +
        ,ship carriers
                                                      interval 90 days)
       ,year
 order by w warehouse name
                                                      and d date sk = ws sold date sk
  limit 100;
                                                      and ws_ext_discount_amt >
                                                      avg_ws_ext_discount_amt_130
order by sum(ws_ext_discount_amt)
-- end query 39 in stream 0 using template
                                                       limit 100;
 -- start query 40 in stream 0 using
template query90.tpl and seed 1949014749
                                                      -- end query 44 in stream 0 using template
select cast(amc as decimal(15,4))/cast(pmc
                                                      query92.tpl
as decimal(15,4)) am pm ratio
                                                       -- start query 45 in stream 0 using
                                                      template query3.tpl and seed 691662667
 from ( select count(*) amc
                                                      select dt.d_year
       from web sales,
                                                             ,item.i_brand_id brand_id
household_demographics , time_dim, web_page
       where ws_sold_time_sk =
                                                             ,item.i_brand brand
time dim.t time sk
                                                              , sum (ss net profit) sum agg
         and ws ship hdemo_sk =
                                                       from date dim dt
household demographics.hd demo sk
                                                            ,store_sales
         and ws_web_page_sk =
                                                            ,item
web_page.wp_web_page_sk
                                                       where dt.d_date_sk =
         and time dim.t hour between 11 and
                                                      store_sales.ss_sold_date_sk
                                                         and store_sales.ss_item_sk =
         and
                                                      item.i_item_sk
household demographics.hd dep count = 9
                                                         and item.i manufact id = 596
        and web_page.wp_char_count between
                                                         and dt.d_moy=12
                                                       group by dt.d_year
5000 and 5200) at
                                                            ,item.i brand
cross join
      ( select count(*) pmc
                                                            ,item.i brand id
```

```
order by dt.d year
                                                    -- end query 49 in stream 0 using template
        ,sum_agg desc
                                                    querv9.tpl
         ,brand id
                                                    -- start query 52 in stream 0 using
                                                    template query93.tpl and seed 1821797098
  limit 100;
                                                    select ss_customer_sk
                                                                ,sum(act_sales) sumsales
-- end query 45 in stream 0 using template
query3.tpl
                                                          from (select ss_item_sk
                                                                      ,ss_ticket number
 -- start query 49 in stream 0 using
                                                                      ,ss_customer sk
template query9.tpl and seed 937436805
select bucket1, bucket2, bucket3, bucket4,
                                                                       ,case when
bucket5
                                                    sr_return_quantity is not null then
from
                                                    (ss quantity-
                                                    sr return quantity) *ss sales price
(select case when count1 > 62316685 then
                                                    else (ss quantity*ss sales price) end
then1 else else1 end bucket1
from (
                                                    act sales
select count(*) count1,
                                                                 from store sales left outer
avg(ss_ext_sales_price) then1,
avg(ss_net_paid_inc_tax) else1
                                                    join store_returns on (sr_item_sk =
                                                    ss_item_sk
from store sales
where ss\_quantity between 1 and 20
                                                    and sr_ticket_number = ss_ticket_number)
) A1) B1
                                                                    ,reason
                                                                where sr_reason_sk =
CROSS JOIN
                                                    r reason sk
                                                                  and r reason desc = 'reason
(select case when count2 > 19045798 then
                                                    74') t
then2 else else2 end bucket2
                                                          group by ss customer sk
from (
                                                          order by sumsales, ss_customer_sk
select count(*) count2,
                                                     limit 100;
avg(ss ext sales price) then2,
avg(ss_net_paid_inc_tax) else2
from store sales
                                                    -- end query 52 in stream 0 using template
                                                    query93.tpl
                                                    -- start query 55 in stream 0 using
where ss_quantity between 21 and 40
) A2) B2
                                                    template query22.tpl and seed 635815297
                                                    with results as
CROSS JOIN
                                                    (select i_product_name
                                                                 ,i brand
(select case when count3 > 365541424 then
                                                                  ,i_class
then3 else else3 end bucket3
                                                                  ,i_category
from (
                                                                  , avg(inv_quantity_on_hand) qoh
select count(*) count3,
                                                            from inventory
avg(ss ext sales price) then3,
                                                               ,date dim
avg(ss_net_paid_inc_tax) else3
                                                               .item
from store sales
                                                               ,warehouse
where ss\_quantity between 41 and 60
                                                           where inv_date_sk=d_date_sk
) A3) B3
                                                                  and inv_item_sk=i_item_sk
                                                                  and inv warehouse sk =
CROSS JOIN
                                                    w_warehouse_sk
                                                                  and d month seq between 1199
(select case when count4 > 216357808 then
                                                    and 1199 + 11
then4 else else4 end bucket4
                                                           group by
from (
                                                    i product name, i brand, i class, i category),
select count(*) count4,
                                                    results_rollup as
avg(ss_ext_sales_price) then4,
avg(ss net paid inc tax) else4
                                                     select i product name, i brand, i class,
from store sales
                                                    i category, qoh from results
where ss quantity between 61 and 80
) A4) B4
                                                     union all
CROSS JOIN
                                                     select i product name, i brand, i class,
(select case when count5 > 184483884 then
                                                    null i category, sum (qoh) from results
then5 else else5 end bucket5
                                                     group by i_product_name,i_brand,i_class
from (
select count(*) count5,
                                                     union all
avg(ss_ext_sales_price) then5,
avg(ss_net_paid_inc_tax) else5
                                                     select i product name, i brand, null
from store sales
                                                    i_class, null i_category, sum(qoh) from
where ss_quantity between 81 and 100
                                                    results
) A5) B5
                                                     group by i_product_name, i_brand
CROSS JOIN
                                                     union all
                                                     select i product name, null i brand, null
reason
where r reason sk = 1
                                                    i class, null i category, sum(qoh) from
                                                    results
                                                     group by i_product_name
```

```
union all
                                                           --and cte1.d moy = cte2.d moy
                                                           and case when (avg_monthly_sales <>
select null i product name, null i brand,
                                                    0) then (abs(sum sales - avg monthly sales)
null i class, null i category, sum(qoh) from
                                                    / avg monthly sales) else null end > 0.1
results
                                                    order by sum_sales - avg_monthly_sales,
                                                    ctel.s_store_name
select i product name, i brand, i class,
                                                     limit 100;
i category, qoh
     from results rollup
                                                    -- end query 56 in stream 0 using template
      order by qoh, i_product_name,
                                                    query89.tpl
                                                    -- start query 59 in stream 0 using
i_brand, i_class, i_category
limit 100;
                                                    template query52.tpl and seed 223505300
                                                    select dt.d year
-- end query 55 in stream 0 using template
                                                           ,item.i brand id brand id
query22.tpl
                                                           ,item.i_brand brand
-- start query 56 in stream 0 using
                                                            ,sum(ss_ext_sales_price) ext_price
template query89.tpl and seed 2079706651
                                                     from date dim dt
                                                        ,store_sales
with ctel as
                                                         ,item
where dt.d date sk =
                                                    store_sales.ss_sold_date_sk
       d moy,
                                                       and store sales.ss item sk =
       sum(ss_sales_price) sum_sales
                                                    item.i_item_sk
from item, store_sales, date_dim, store
                                                        and item.i manager id = 1
where ss item sk = i item sk and
                                                        and dt.d moy=11
      ss sold date sk = d date sk and
                                                        and dt.d year=1999
      ss_store_sk = s_store_sk and
                                                     group by dt.d_year
      d_{year\ in}\ (1999) and
                                                           ,item.i_brand
        ((i category in
                                                            ,item.i brand id
('Books','Jewelry','Men') and
                                                     order by dt.d year
                                                           ,ext_price desc
         i class in
('history', 'birdal', 'pants')
                                                            ,brand id
                                                     limit 100 ;
      or (i_category in
('Music','Home','Shoes') and
                                                    -- end query 59 in stream 0 using template
i_class in
('pop','furniture','athletic')
                                                    query52.tpl
                                                    -- start query 60 in stream 0 using
       ))
                                                    template query50.tpl and seed 1718577076
group by i_category, i_class, i_brand,
                                                    select
                                                      s store name
        s_store_name, s_company_name,
d moy),
                                                      ,s_company_id
                                                      ,s_street number
cte2 as
                                                      ,s_street_name
                                                      ,s street type
                                                      ,s_suite_number
select
i_category, i_brand, s_store_name,
                                                      ,s_city
s company name,
                                                      ,s county
avg(sum sales) avg monthly sales
                                                      ,s state
from ctel
                                                      ,s_zip
                                                      , sum(case when (sr returned date sk -
group by
i_category, i_brand, s_store_name,
                                                    ss\_sold\_date\_sk \le 30 ) then 1 else 0 end)
s_company_name
                                                    as "30 days"
                                                     , sum(case when (sr returned date sk -
                                                    ss\_sold\_date\_sk > 30 and
select ctel.i_category, ctel.i_class,
ctel.i brand,
                                                                     (sr returned date sk -
      ctel.s store name,
                                                    ss sold date sk \leq 60) then 1 else 0 end )
ctel.s_company_name, ctel.d moy,
                                                    as "31-60 days"
      ctel.sum_sales,
                                                     ,sum(case when (sr_returned_date_sk -
cte2.avg_monthly_sales
                                                    ss_sold_date_sk > 60) and
from ctel cross join ctel
                                                                     (sr returned date sk -
                                                    ss sold date sk \leq 90) then 1 else 0 end)
       (ctel.i_category = cte2.i_category
or (ctel.i_category is NULL and
                                                    as "61-90 days"
                                                     ,sum(case when (sr_returned_date_sk -
cte2.i_category is NULL))
                                                    ss_sold_date_sk > 90) and
                                                                     (sr returned date sk -
                                                    ss_sold_date_sk \le 120) then 1 else 0 end)
       and (ctel.i brand = cte2.i brand
       or (ctel.i brand is NULL and
                                                    as "91-120 days"
cte2.i_brand is NULL))
                                                     ,sum(case when (sr_returned_date_sk -
                                                    ss_sold_date_sk > 1\overline{20}) then \overline{1} else 0 end)
                                                    as ">120 days"
       and (cte1.s_store_name =
                                                    from
cte2.s_store_name
       or (ctel.s_store_name is NULL and
                                                      store_sales
cte2.s_store_name is NULL))
                                                      ,store returns
       and (ctel.s company name =
                                                     ,store
                                                      ,date_dim d1
cte2.s_company_name
       or (ctel.s_company_name is NULL and
                                                      ,date_dim d2
cte2.s_company_name is NULL))
                                                    where
```

```
d2.d year = 1999
                                                              (i color = 'magenta' or i color =
and d2.d_moy = 10
and ss ticket number = sr ticket number
                                                      'goldenrod') and
                                                              (i units = 'Cup' or i units = 'Oz')
and ss\_item\_sk = sr\_item\_sk
and ss_sold_date_sk = d1.d_date_sk
                                                              (i size = 'economy' or i size =
and sr_returned_date_sk = d2.d_date_sk
                                                      'extra large')
and ss_customer_sk = sr_customer_sk
                                                              ) or
                                                              (i_category = 'Men' and
and ss store sk = s store sk
                                                              (i color = 'cyan' or i color =
group by
  s_store name
                                                      'antique') and
                                                              (i units = 'Dozen' or i units =
  ,s_company_id
 ,s_street_number
                                                      'Case') and
                                                              (i size = 'medium' or i size =
  ,s street name
  ,s street type
                                                      'petite')
  ,s_suite_number
                                                              ) or
  ,s_city
                                                              (i_category = 'Men' and
                                                              (i color = 'moccasin' or i color =
  ,s county
  ,s_state
                                                      'black') and
                                                              (i units = 'Box' or i units =
  s zip
                                                      'Pallet') and
order by s_store_name
                                                              (i_size = 'large' or i size =
        ,s_company_id
        ,s street number
                                                      'N/A')
        ,s_street_name
                                                              ))) or
        ,s street type
                                                              ((i_category = 'Women' and
(i color = 'azure' or i color =
        ,s suite number
        ,s city
                                                      'light') and
        ,s county
                                                              (i units = 'Gross' or i units =
        ,s_state
                                                      'Each') and
limit 100;
                                                              (i size = 'large' or i size =
                                                      'N/A')
-- end query 60 in stream 0 using template
                                                              (i_category = 'Women' and
query50.tpl
-- start query 61 in stream 0 using
                                                              (i color = 'mint' or i color =
                                                      'burnished') and
template query42.tpl and seed 709936855
                                                      (i_units = 'N/A' or i_units = 'Unknown') and
select dt.d_year
       ,item.i_category_id
        ,item.i category
                                                              (i size = 'economy' or i size =
        , sum(ss ext sales price)
                                                      'extra large')
from date dim dt
                                                              ) or
                                                              (i_category = 'Men' and
(i_color = 'floral' or i_color =
       ,store_sales
        .item
where dt.d date sk =
                                                      'midnight') and
store sales.ss sold date sk
                                                              (i_units = 'Pound' or i units =
       and store_sales.ss_item_sk =
                                                      'Ton') and
                                                              (i_size = 'medium' or i size =
item.i_item_sk
        and item.i manager id = 1
                                                      'petite')
        and dt.d moy=12
                                                              (i_category = 'Men' and (i_color = 'navy' or i_color =
       and dt.d_year=2000
 group by
               dt.d year
               ,item.i_category_id
                                                      'blue') and
               ,item.i_category
                                                              (i units = 'Bundle' or i units =
                sum(ss_ext_sales_price)
                                                      'Ounce') and
order by
                                                              (i_size = 'large' or i size =
desc, dt.d year
               ,item.i_category_id
                                                      'N/A')
               ,item.i category
limit 100;
                                                           group by i manufact) i2
                                                     ON i1.i manufact = i2.i manufact
-- end query 61 in stream 0 using template
                                                      where i1.i_manufact_id between 716 and
                                                      716+40
-- start query 62 in stream 0 using
                                                      and i2.item cnt > 0
template query41.tpl and seed 944250029
                                                      order by i_product_name
                                                       limit 100;
select distinct(i_product_name)
from item i1
 JOIN (select i manufact, count(*) as
                                                      -- end query 62 in stream 0 using template
item cnt
                                                     querv41.tpl
        from item
                                                      -- start query 64 in stream 0 using
                                                     template query12.tpl and seed 918962166
        where (
        ((i\_category = 'Women' and
                                                     with ctel as
        (i_color = 'spring' or i_color =
                                                     ( select
'hot.') and
                                                         i_item_id
        (i_units = 'Carton' or i_units =
                                                         ,i_item_desc
'Tbl') and
                                                         ,i category
        (i_size = 'large' or i_size =
                                                         ,i_class
'N/A')
                                                         \tt,i\_current\_price
                                                          , sum(ws_ext_sales_price) as itemrevenue
        (i category = 'Women' and
                                                          from web sales
```

```
,item
                                                         and d date between cast('2000-02-09' as
         ,date dim
                                                     timestamp)
    where
                                                            and (cast('2000-02-09' as
        ws item sk = i item sk
                                                     timestamp) + interval 30 days)
        and ws_sold_date_sk = d_date_sk
                                                         group by
        and i_category in ('Jewelry',
                                                            i_item_id
'Men', 'Books')
                                                             ,i_item_desc
        and d date between cast('2002-06-
                                                             ,i category
                                                             ,i_class
11' as timestamp)
               and (cast('2002-06-11' as
                                                             ,i_current_price ),
timestamp) + interval 30 days)
                                                     cte2 as
                                                     ( select
    group by
        i item id
                                                         i class
        ,i item desc
                                                         , sum (itemrevenue) as sumitemrevenue
                                                       from cte1
        ,i_category
        ,i class
                                                       group by
        ,i_current_price ),
                                                        i class)
                                                       select i_item desc
cte2 as
( select
                                                        ,i_category
    i class
                                                        ,i_class
    ___,sum(itemrevenue) as sumitemrevenue
                                                        ,i_current_price
  from cte1
                                                        ,itemrevenue
                                                        ,revenueratio
  group by
    i class)
                                                     from (
 select i item desc
                                                     select
   ,i category
                                                       ctel.i item id
                                                        ,ctel.i_item_desc
   ,i class
   ,i_current_price
                                                        ,cte1.i_category
   ,itemrevenue
                                                        ,ctel.i_class
                                                        ,ctel.i current price
   ,revenueratio
from (
                                                        ,ctel.itemrevenue
select
  ctel.i_item_id
                                                     ,ctel.itemrevenue*100/cte2.sumitemrevenue
   ,cte1.i_item_desc
                                                     as revenueratio
   ,cte1.i_category
                                                     from ctel
   ,cte1.i_class
,cte1.i_current_price
                                                         cross join cte2
                                                     where
   ,ctel.itemrevenue
                                                        (cte1.i class = cte2.i class
                                                        or
                                                        (ctel.i_class is NULL and cte2.i_class
,cte1.itemrevenue*100/cte2.sumitemrevenue
as revenueratio
                                                     is NULL))
from cte1
                                                     ) v1
     cross join cte2
                                                     order by
where
                                                        i category
  (cte1.i_class = cte2.i class
                                                        ,i_class
   or
                                                        ,i_item_id
                                                        ,i_item_desc
   (ctel.i class is NULL and cte2.i class
is NULL))
                                                        , revenueratio
) v1
                                                      limit. 100:
order by
  i_category
                                                     -- end query 65 in stream 0 using template
   ,i_class
,i_item_id
                                                     query20.tpl
                                                     -- start query 66 in stream 0 using
   ,i_item desc
                                                     template query88.tpl and seed 1924183468
   ,revenueratio
                                                     select
limit 100;
                                                     from
                                                      (select count(*) h8 30 to 9
-- end query 64 in stream 0 using template
                                                      from store_sales, household_demographics ,
query12.tpl
                                                     time dim, store
-- start query 65 in stream 0 using
                                                      where ss sold time sk = time dim.t time sk
template query20.tpl and seed 711739272
                                                         and ss hdemo sk =
with ctel as
                                                     household_demographics.hd demo sk
( select
                                                          and ss\_store\_sk = s\_store\_sk
    i_item_id
                                                          and time_dim.t_hour = 8
                                                          and time dim.t minute >= 30
    ,i_item_desc
    ,i_category
                                                          and
    ,i_class
                                                     ((household demographics.hd dep count = 1
                                                     and
    ,i_current_price
    ,sum(cs_ext_sales_price) as itemrevenue
                                                     household_demographics.hd_vehicle_count<=1+
    {\tt from\ catalog\_sales}
         ,item
                                                     (household demographics.hd dep count = 4
         ,date_dim
    where
        cs item sk = i item sk
                                                     household demographics.hd vehicle count<=4+
    and i_category in ('Jewelry', 'Music',
                                                     2) or
'Men')
    and cs_sold_date_sk = d_date_sk
                                                     (household demographics.hd dep count = 2
```

```
household demographics.hd vehicle count<=2+
                                                    (household demographics.hd dep count = 2
                                                    and
                                                    household demographics.hd vehicle count <= 2+
    and store.s store name = 'ese') s1
cross join
                                                    2))
 (select count(*) h9 to 9 30
                                                         and store.s_store_name = 'ese') s4
 from store sales, household demographics ,
                                                    cross join
                                                     (select count(*) h10 30 to 11
time dim, store
where ss sold time sk = time dim.t time sk
                                                    from store sales, household demographics,
    and ss hdemo sk =
                                                    time_dim, store
household_demographics.hd_demo_sk
                                                    where ss_sold_time_sk = time_dim.t_time_sk
     and ss_store_sk = s_store_sk
                                                         and ss_hdemo_sk =
     and time dim.t hour = 9
                                                    household demographics.hd demo sk
                                                         and ss store_sk = s_store_sk
     and time dim.t minute < 30
     and
                                                         and time \dimt hour = 10
((household_demographics.hd_dep_count = 1
                                                         and time_dim.t_minute >= 30
household demographics.hd vehicle count<=1+
                                                    ((household demographics.hd dep count = 1
2) or
                                                    and
                                                   household demographics.hd vehicle count<=1+
(household_demographics.hd_dep_count = 4
household demographics.hd vehicle_count<=4+
                                                    (household demographics.hd dep count = 4
                                                   and
2) or
                                                    household demographics.hd vehicle count<=4+
(household demographics.hd dep count = 2
                                                   2) or
household_demographics.hd_vehicle_count<=2+
                                                    (household demographics.hd dep count = 2
                                                    and
                                                    household demographics.hd vehicle count<=2+
    and store.s store name = 'ese') s2
cross join
                                                   2))
 (select count(*) h9 30 to 10
                                                         and store.s_store_name = 'ese') s5
 from store_sales, household_demographics,
                                                   cross join
                                                     (select count(*) h11_to_11_30
time dim, store
where ss sold time sk = time dim.t time sk
                                                    from store sales, household demographics,
    and ss_hdemo_sk =
                                                    time dim, store
                                                    where ss sold time sk = time dim.t time sk
household demographics.hd demo sk
    and ss store sk = s store sk
                                                         and ss hdemo sk =
     and time_dim.t_hour = 9
                                                    household demographics.hd demo sk
                                                        and ss store_sk = s_store_sk
     and time dim.t minute >= 30
                                                         and time_dim.t_hour = 11
     and
((household demographics.hd dep count = 1
                                                         and time dim.t minute < 30
                                                         and
household demographics.hd vehicle count<=1+
                                                    ((household demographics.hd dep count = 1
2) or
                                                    and
                                                    household_demographics.hd_vehicle_count<=1+
(household demographics.hd dep count = 4
household demographics.hd vehicle count<=4+
                                                    (household demographics.hd dep count = 4
2) or
                                                    and
                                                   household_demographics.hd_vehicle_count<=4+
(household demographics.hd dep count = 2
                                                    2) or
and
household demographics.hd vehicle_count<=2+
                                                    (household demographics.hd dep count = 2
                                                    and
    and store.s store name = 'ese') s3
                                                   household demographics.hd vehicle count<=2+
cross join
                                                    2))
 (select count(*) h10_to_10_30
                                                         and store.s_store_name = 'ese') s6
 from store sales, household demographics ,
                                                    cross join
                                                     (select count(*) h11 30 to 12
time dim, store
where ss sold time sk = time dim.t time sk
                                                    from store sales, household demographics,
    and ss hdemo sk =
                                                    time_dim, store
household_demographics.hd_demo_sk
                                                    where ss_sold_time_sk = time_dim.t_time_sk
     and ss_store_sk = s_store_sk
                                                         and ss_hdemo_sk =
     and time dim.t hour = 10
                                                    household demographics.hd demo sk
                                                        and ss store sk = s_store_sk
     and time dim.t minute < 30
                                                         and time dim.t hour = 11
     and
((household_demographics.hd_dep_count = 1
                                                         and time_dim.t_minute >= 30
                                                         and
                                                    ((household demographics.hd_dep_count = 1
household demographics.hd vehicle count<=1+
2) or
                                                    and
                                                   household demographics.hd vehicle count<=1+
(household demographics.hd dep count = 4
household demographics.hd vehicle count<=4+
                                                    (household demographics.hd dep count = 4
2) or
                                                   and
```

```
household demographics.hd vehicle count<=4+
                                                                             ss item sk as
2) or
                                                    sold item sk,
                                                                              ss sold time sk as
(household demographics.hd dep count = 2
                                                    {\tt time \ sk}
                                                                      from store sales, date dim
household demographics.hd vehicle count<=2+
                                                                      where d_date_sk =
                                                    ss sold date sk
     and store.s store name = 'ese') s7
                                                                        and d moy=12
                                                                        and d year=1998
cross join
 (select count(*) h12 to 12 30
                                                                      ) as tmp, time dim
 from store_sales, household_demographics ,
                                                     where
time_dim, store
                                                        sold item sk = i item sk
where ss sold time sk = time dim.t time sk
                                                        and i manager id=1
                                                       and time sk = t_time_sk
     and ss hdemo sk =
                                                       and (t_meal_time = 'breakfast' or
household demographics.hd demo sk
                                                    t meal time = 'dinner')
     and ss\_store\_sk = s\_store\_sk
     and time \dimt hour = 12
                                                     group by i brand,
     and time_dim.t_minute < 30
                                                     i brand id, t hour, t minute
                                                     order by ext_price desc, i_brand_id
     and
((household demographics.hd dep count = 1
household demographics.hd vehicle count<=1+
                                                    -- end query 72 in stream 0 using template
2) or
                                                    query71.tpl
                                                     -- start query 73 in stream 0 using
(household demographics.hd dep count = 4
                                                    template query34.tpl and seed 1451328249
                                                    select c last name
                                                           ,c_first name
household demographics.hd vehicle count<=4+
2) or
                                                            ,c_salutation
                                                            ,c preferred cust flag
(household demographics.hd dep count = 2
                                                            ,ss ticket number
                                                            ,cnt from
                                                        (select ss_ticket number
household demographics.hd vehicle count<=2+
                                                               ,ss_customer_sk
     and store.s store name = 'ese') s8
                                                               ,count(*) cnt
                                                        from
                                                    store_sales,date_dim,store,household_demogr
-- end query 66 in stream 0 using template
                                                    aphics
query88.tpl
                                                        where store sales.ss sold date sk =
 - start query 72 in stream 0 using
                                                     date dim.d date sk
template query71.tpl and seed 1436004490
                                                        and store_sales.ss_store_sk =
select i_brand_id brand_id, i_brand
                                                     store.s_store_sk
brand, t_hour, t_minute,
                                                        and store sales.ss hdemo sk =
       sum(ext price) ext price
                                                     household demographics.hd demo sk
from item, (select ws ext sales price as
                                                        and (date dim.d dom between 1 and 3 or
                                                     date dim.d dom between 25 and 28)
ext price,
                        ws_sold_date_sk as
                                                        and
sold date sk,
                                                     (household demographics.hd buy potential =
                                                     '1001-5000' or
                        ws item sk as
sold item sk,
                        ws sold time sk as
                                                    household demographics.hd buy potential =
time sk
                                                     '5001-100<del>0</del>0')
                 from web sales, date dim
                                                        and
                 where d_date_sk =
                                                    household demographics.hd vehicle count > 0
ws sold date sk
                                                        and (case when
                   and d moy=12
                                                    household demographics.hd vehicle count > 0
                   and d year=1998
                                                            then
                 union all
                                                     household demographics.hd dep count/
                                                    household_demographics.hd_vehicle_count
                 select cs_ext_sales_price
as ext price,
                                                            else null
                         cs sold date sk as
                                                            end) > 1.2
                                                         and date dim.d year in
sold date sk,
                        cs item sk as
                                                     (1999,1999+1,1999+2)
sold_item_sk,
                                                        and store.s_county in ('Sierra
                         cs sold time sk as
                                                    County','Lunenburg County','Jackson
                                                    County', 'Harmon County',
time sk
                 from
                                                    County','Pipestone County','Pennington
County','Perry County')
catalog_sales,date dim
                 where d_date_sk =
cs sold date sk
                                                        group by
                   and d_moy=12
                                                    ss ticket number,ss customer sk)
                   and d year=1998
                                                    dn, customer
                 union all
                                                        where ss\_customer\_sk = c\_customer\_sk
                 select ss ext sales price
                                                          and cnt between 15 and 20
as ext price,
                                                        order by
                                                    c last name,c_first_name,c_salutation,c_pre
                        ss sold date sk as
                                                     ferred_cust_flag desc;
sold_date_sk,
```

```
-- end guery 73 in stream 0 using template
                                                         date dim,
querv34.tpl
                                                         web page
-- start query 78 in stream 0 using
                                                     where wr returned date sk = d date sk
template query77.tpl and seed 1879081522
                                                          and d date between cast('2002-08-24'
with ss as
                                                    as timestamp)
                                                                      and (cast('2002-08-24' as
 (select s_store_sk,
                                                    timestamp) + interval 30 days)
         sum(ss_ext_sales_price) as sales,
         sum(ss net profit) as profit
                                                          and wr web page sk = wp web page sk
                                                     group by wp web page sk)
from store sales,
     date \overline{\text{dim}},
     store
                                                     results as
where ss sold date sk = d date sk
                                                     (select channel
                                                           , id
      and d date between cast('2002-08-24'
                                                           , sum(sales) as sales
as timestamp)
                  and (cast('2002-08-24') as
                                                            , sum(returnz) as returnz
timestamp) + interval 30 days)
                                                            , sum(profit) as profit
     and ss store sk = s store sk
group by s_store_sk)
                                                     (select 'store channel' as channel
                                                            , ss.s_store_sk as id
                                                            , sales
 sr as
                                                            , coalesce(returnz, 0) as returnz
 (select s_store_sk,
        sum(sr return amt) as returnz,
                                                            , (profit -
                                                   coalesce(profit_loss,0)) as profit
         sum(sr_net_loss) as profit_loss
 from store returns,
                                                    from ss left join sr
     date dim,
                                                            on ss.s store sk = sr.s store sk
     store
                                                    union all
where sr returned date sk = d date sk
                                                    select 'catalog channel' as channel
       and d date between cast ('2002-08-24'
                                                           , cs_call_center_sk as id
                                                           , sales
as timestamp)
                  and (cast('2002-08-24' as
timestamp) + interval 30 days)
                                                            , (profit - profit loss) as profit
     and sr store sk = s store_sk
                                                     from cs
group by s_store_sk),
                                                          cross join cr
                                                     union all
 cs as
 (select cs call center sk,
                                                     select 'web channel' as channel
        sum(cs_ext_sales_price) as sales,
sum(cs_net_profit) as profit
                                                           , ws.wp_web_page_sk as id
                                                            , sales
 from catalog sales,
                                                            , coalesce(returnz, 0) returnz
     date dim
                                                            , (profit ·
 where cs sold date sk = d date sk
                                                   coalesce(profit_loss,0)) as profit
      and d date between cast('2002-08-24'
                                                    from ws left join wr
as timestamp)
                                                            on ws.wp web page sk =
                  and (cast('2002-08-24' as
                                                    wr.wp_web_page_sk
timestamp) + interval 30 days)
                                                    ) x
group by cs_call_center_sk
                                                    group by channel, id )
),
cr as
                                                     select channel
                                                         , id
 (select
        sum(cr_return_amount) as returnz,
                                                           , sales
                                                          , returnz
        sum(cr net loss) as profit loss
 from catalog_returns,
                                                           , profit
     date dim
                                                    from (
where cr returned date sk = d date sk
                                                    select channel, id, sales, returnz, profit
       and d_date between cast('2002-08-24'
                                                    from results
as timestamp)
                                                    union
                  and (cast('2002-08-24' as
                                                     select channel, NULL AS id, sum(sales) as
timestamp) + interval 30 days)
                                                    sales, sum(returnz) as returnz, sum(profit)
                                                   as profit from results group by channel
ws as
                                                    union
 ( select wp web page sk,
                                                     select NULL AS channel, NULL AS id,
       sum(ws ext sales price) as sales,
                                                    sum(sales) as sales, sum(returnz) as
        sum(ws_net_profit) as profit
                                                    returnz, sum(profit) as profit from
 from web_sales,
                                                    results
      date dim,
                                                    ) foo
                                                   order by channel, id
      web page
                                                     limit 100;
where ws sold date sk = d date sk
      and d date between cast('2002-08-24'
                                                    -- end query 78 in stream 0 using template
as timestamp)
                 and (cast('2002-08-24' as
                                                   query77.tpl
timestamp) + interval 30 days)
                                                    -- start query 79 in stream 0 using
                                                    template query73.tpl and seed 413577677
      and ws_web_page_sk = wp_web_page_sk
group by wp_web_page_sk),
                                                    select c last name
                                                          ,c_first name
 wr as
 (select wp web page sk,
                                                          ,c salutation
        sum(wr_return_amt) as returnz,
                                                           ,c_preferred_cust flag
        sum(wr_net_loss) as profit_loss
                                                           ,ss_ticket_number
 from web returns,
                                                           ,cnt from
```

```
(select ss ticket number
                                                    group by i brand, i brand id
          ,ss customer sk
                                                    order by ext_price desc, i_brand_id
          ,count(*) cnt
                                                    limit 100;
store sales, date dim, store, household demogr
                                                    -- end query 82 in stream 0 using template
aphics
                                                    query55.tpl
                                                    -- start query 83 in stream 0 using
    where store_sales.ss_sold_date_sk =
date dim.d date sk
                                                    template query56.tpl and seed 1152645577
   and store sales.ss store sk =
                                                    with ss as (
store.s_store_sk
                                                    select
   and store_sales.ss_hdemo_sk =
                                                    item.i_item_id, sum(ss_ext_sales_price)
household_demographics.hd_demo_sk
                                                    total_sales
    and date dim.d dom between 1 and 2
                                                    from
                                                           store sales,
(household_demographics.hd_buy_potential =
                                                           date \overline{\text{dim}},
'501-1000' or
                                                            customer address,
household demographics.hd buy potential =
                                                            left semi join (
'5001-100<del>0</del>0')
                                                                            select i item_id
   and
                                                                            from item
household_demographics.hd_vehicle_count > 0
                                                                            where i_color in
  and case when
                                                    ('maroon','powder','lawn')
household demographics.hd vehicle count > 0
                                                     ) ssi on item.i_item_id =
                                                    ssi.i item id
                                                    where ss_item_sk
household demographics.hd dep count/
                                                    i item sk
household demographics.hd vehicle count
                                                            ss_sold_date_sk
                                                    and
else null end > 1
                                                    d_date_sk
                                                         d_year
d_moy
    and date dim.d year in
                                                    and
                                                                                     = 2000
                                                                                     = 1
(1999, 1999+1, 1999+2)
                                                    and
   and store.s county in ('Lea
                                                    and
                                                            ss addr sk
County','West Feliciana Parish','Nowata
                                                    ca address sk
County', 'Jackson County')
                                                    and ca_gmt_offset
                                                                                     = -5
                                                    group by i_item_id),
    group by
ss ticket_number,ss_customer_sk)
                                                    cs as (
                                                    select.
dj,customer
    where ss_customer_sk = c_customer_sk
                                                    item.i item id, sum(cs ext sales price)
     and cnt between 1 and \overline{5}
                                                    total_sales
    order by cnt desc;
                                                           catalog_sales,
-- end query 79 in stream 0 using template
                                                           date_dim,
query73.tpl
                                                            customer address,
-- start query 80 in stream 0 using
                                                             item
template query84.tpl and seed 1842474049
                                                            left semi join (
                                                                            `select i item_id
select c_customer_id as customer_id
      ,concat(c_last_name, ', ',
                                                                            from item
coalesce(c first name,'')) as customername
                                                                            where i color in
                                                    ('maroon','powder','lawn')
from customer
                                                           ) csi on item.i item id =
     , customer\_address
     ,customer demographics
                                                    csi.i item id
                                                    where cs_item_sk
     ,household demographics
     ,income_band
                                                    i item sk
     ,store returns
                                                    and
                                                            cs_sold_date_sk
                      = 'Mount Zion'
 where ca_city
                                                    d date sk
   and c_current_addr_sk = ca_address sk
                                                          d_year
d_moy
                                                                                     = 2000
                                                    and
  and ib_lower_bound >= 50749 and ib_upper_bound <= 50749 + 50000 and ib_income_band_sk =
                                                    and
                                                    and
                                                            cs bill addr sk
                                                    ca_address_sk
hd_income_band_sk
                                                    and
                                                           ca_gmt_offset
                                                                                     = -5
   and cd demo sk = c current cdemo sk
                                                    group by i item id),
   and hd demo sk = c current hdemo sk
                                                    ws as (
   and sr_cdemo_sk = cd_demo_sk
                                                    select
 order by c_customer_id
                                                    item.i_item_id, sum(ws_ext_sales_price)
  limit 100;
                                                    total_sales
-- end query 80 in stream 0 using template
                                                           web sales,
query84.tpl
                                                           date dim,
-- start query 82 in stream 0 using
                                                            customer_address,
template query55.tpl and seed 1117454508
                                                             item
left semi join (
                                                                            select i item_id
 from date_dim, store_sales, item
                                                                            from item
 where d date sk = ss sold date sk
                                                                            where i\_color in
       and ss item sk = i item sk
                                                    ('maroon','powder','lawn')
       and i_manager_id=48
                                                           ) wsi on item.i_item_id =
       and d_moy=11
                                                    wsi.i_item_id
       and d year=2001
```

```
,sat sales sat_sales1
where ws_item_sk
                                                      from wswscs,date_dim
i_item_sk
 and
        ws sold date sk
                                                      where date dim.d week seq =
                                                     wswscs.d week seq and
d date sk
      d_year
and
                                  = 2000
                                                            \bar{d} \text{ year} = 1998) \text{ y,}
                                                      (select wswscs.d_week_seq d_week_seq2
and
        d moy
                                  = 1
       ws_bill_addr_sk
and
                                                            sun_sales sun_sales,
ca address sk
                                                             ,mon sales mon sales2
                                                             , tue sales tue sales2
and ca gmt offset
group by i_item_id)
  select i_item_id ,sum(total_sales)
                                                             ,wed_sales wed_sales2
                                                             ,thu_sales thu_sales2
total_sales
                                                             fri_sales fri_sales2
 from (select * from ss
                                                             ,sat sales sat sales2
       union all
                                                      from wswscs
        select * from cs
                                                          ,date dim
        union all
                                                      where date_dim.d_week_seq =
        select * from ws) tmp1
                                                     wswscs.d_week_seq and
                                                            \frac{1}{2} d year = 1998+1) z
group by i item id
order by total_sales
                                                     where d week seq1=d week seq2-53
 limit 100;
                                                     order by d week seq1;
-- end query 83 in stream 0 using template
                                                     -- end query 84 in stream 0 using template
querv56.tpl
                                                    query2.tpl
-- start query 84 in stream 0 using
                                                     -- start query 85 in stream 0 using
template query2.tpl and seed 1528114170
                                                     template query26.tpl and seed 1427200905
with wscs as
                                                     select i item id,
 (select sold date sk
                                                            avg(cs quantity) agg1,
        ,sales_price
                                                             avg(cs_list_price) agg2,
 from (select ws sold date sk sold date sk
                                                             avg(cs coupon amt) agg3,
              ,ws ext sales price
                                                             avg(cs_sales_price) agg4
sales price
                                                     from catalog_sales, customer_demographics,
       from web sales) x
                                                     date dim, item, promotion
        union all
                                                     where cs\_sold\_date\_sk = d\_date\_sk and
       (select cs sold date sk sold date sk
                                                            cs_item_sk = i_item_sk and
              ,cs ext sales price
                                                            cs_bill_cdemo_sk = cd_demo_sk and
                                                            cs_promo_sk = p_promo_sk and
cd_gender = 'M' and
sales_price
       from catalog sales)),
wswscs as
                                                            cd marital status = 'D' and
                                                            cd education status = 'Advanced
 (select d week seq,
                                                    Degree' and
       sum(case when (d_day_name='Sunday')
                                                    (p_channel_email = 'N' or
p_channel_event = 'N') and
then sales_price else null end) sun_sales,
       sum(case when (d_day_name='Monday')
then sales price else null end) mon sales,
                                                          d_year = 2000
        sum(case when
                                                      group by i item id
(d_day_name='Tuesday') then sales_price
                                                     order by i item id
                                                      limit 100;
else null end) tue_sales,
       sum(case when
(d day name='Wednesday') then sales price
                                                     -- end query 85 in stream 0 using template
else null end) wed sales,
                                                     querv26.tpl
                                                     -- start query 86 in stream 0 using
        sum(case when
(d_day_name='Thursday') then sales_price
                                                     template query40.tpl and seed 600490395
else null end) thu sales,
                                                     select
       sum(case when (d day name='Friday')
                                                       w_state
then sales_price else null end) fri_sales,
                                                       ,i_item_id
                                                       ,sum(case when d_date < cast ('2000-04-
        sum(case when
                                                     27' as timestamp)
(d day name='Saturday') then sales price
else null end) sat sales
                                                                    then cs sales price -
                                                     coalesce(cr_refunded_cash,0) else 0 end) as
from wscs
    ,date dim
                                                     sales before
where d date sk = sold date sk
                                                       , sum(case when d date >= cast ('2000-04-
group by d week seq)
                                                     27' as timestamp)
select d week seq1
                                                                    then cs_sales_price -
       , round(sun_sales1/sun_sales2,2)
                                                     coalesce(cr_refunded_cash,0) else 0 end) as
       , round(mon_sales1/mon_sales2,2)
                                                     sales_after
       ,round(tue_sales1/tue_sales2,2)
       round (wed sales1/wed sales2,2)
                                                      catalog sales left outer join
       , round(thu_sales1/thu_sales2,2)
, round(fri_sales1/fri_sales2,2)
                                                    catalog returns on
                                                           (cs_order_number = cr order number
       ,round(sat_sales1/sat_sales2,2)
                                                            and cs_item_sk = cr_item_sk)
                                                      ,warehouse
 from
                                                      ,item
 (select wswscs.d_week_seq d_week_seq1
        ,sun_sales sun_sales1
                                                       ,date dim
        ,mon sales mon_sales1
                                                      where
        ,tue sales tue sales1
                                                        i current price between 0.99 and 1.49
                                                                      = cs_item_sk
                                                     and i_item sk
        , wed_sales wed_sales1
                                                                             = w_warehouse_sk
                                                     and cs_warehouse_sk
        ,thu_sales thu_sales1
                                                                             = d_date_sk
        fri sales fri sales1
                                                     and cs sold date sk
```

```
and d date between (cast ('2000-04-27' as
                                                     -- start query 89 in stream 0 using
                                                     template query79.tpl and seed 2112737383
timestamp) - interval 30 days)
                 and (cast ('2000-04-27' as
                                                     select
timestamp) + interval 30 days)
group by
                                                     c last name, c first name, substr(s city, 1, 30
   w_state,i_item_id
                                                     ),ss_ticket_number,amt,profit
order by w_state,i_item_id
                                                       from
  limit 100;
                                                         (select ss ticket number
                                                                ,ss customer sk
-- end query 86 in stream 0 using template
                                                                ,store.s_city
query40.tpl
                                                                ,sum(ss_coupon_amt) amt
-- start query 88 in stream 0 using
                                                                ,sum(ss_net_profit) profit
template query53.tpl and seed 1796782974
with ctel as
                                                     store sales, date dim, store, household demogr
                                                     aphics
select i manufact id,
                                                         where store_sales.ss_sold_date_sk =
                                                      date dim.d date sk
sum(ss sales price) sum sales
                                                         and store_sales.ss_store_sk =
from item, store_sales, date_dim, store
                                                      store.s_store_sk
where ss_item_sk = i_item_sk and ss_sold_date_sk = d_date_sk and
                                                         and store sales.ss hdemo sk =
                                                     household_demographics.hd_demo_sk
ss store_sk = s_store_sk and
                                                         and
d_month_seq in
                                                      (household demographics.hd dep count = 3 or
(1198, 1198+1, 1198+2, 1198+3, 1198+4, 1198+5, 11
                                                     household demographics.hd vehicle count >
98+6,1198+7,1198+8,1198+9,1198+10,1198+11)
and
                                                          and date dim.d dow = 1
                                                         and date dim.d year in
((i category in
('Books','Children','Electronics') and
                                                      (2000, 2000+1, 2000+2)
i class in
                                                         and store.s_number_employees between
('personal', 'portable', 'reference', 'self-
                                                      200 and 295
help') and
                                                         aroup by
i brand in ('scholaramalgamalg
                                                      ss ticket number,ss_customer_sk,ss_addr_sk,
#14','scholaramalgamalg #7',
                                                      store.s_city) ms,customer
                'exportiunivamalg
                                                         where ss_customer_sk = c_customer_sk
#9','scholaramalgamalg #9'))
                                                      order by
or(i_category in ('Women','Music','Men')
                                                      c_last_name,c_first_name,substr(s_city,1,30
and
                                                      ), profit
i class in
                                                      limit 100;
( accessories', 'classical', 'fragrances', 'pa
nts') and
                                                      -- end query 89 in stream 0 using template
i_brand in ('amalgimporto #1','edu
                                                     query79.tpl
packscholar #1', 'exportiimporto #1',
                                                      -- start query 96 in stream 0 using
                'importoamalg #1')))
                                                      template query83.tpl and seed 593789178
group by i manufact id, d qoy
                                                     with sr items as
                                                       (select i_item_id item_id,
) ,
cte2 as
                                                             sum(sr_return_quantity) sr_item_qty
                                                       from store returns,
select
                                                           item,
                                                            date_dim
 i manufact id.
                                                            left semi join
  avg(sum sales) avg quarterly sales
from ctel
                                                                       (select d date
 group by
                                                                       from date dim
  i_manufact_id)
                                                            left semi join
                                                                       (select d_week_seq
select
                                                                      from date dim
 ctel.i manufact id
                                                               where d date in ('1999-06-
                                                      14','1999-08-26','1999-11-06')
  ,ctel.sum sales
  ,cte2.avg_quarterly_sales
                                                           ) d3 on date_dim.d_week_seq =
from ctel
                                                      d3.d_week_seq
     cross join cte2
                                                           ) d\overline{2} on date dim.d date = d2.d date
                                                       where sr item sk = i item sk
                                                      and sr_returned_date_sk = d_date_sk group by i_item_id),
  (cte1.i manufact id = cte2.i manufact id
  (ctel.i_manufact_id is NULL and
                                                       cr items as
                                                       (select i item id item id,
cte2.i manufact id is NULL))
  and case when avg_quarterly_sales > 0
                                                             sum(cr return quantity) cr item qty
then abs (sum_sales - avg_quarterly_sales) / avg_quarterly_sales
                                                      from catalog_returns,
                                                           item,
       else \overline{\text{null}} end > 0.1
                                                            date_dim
order by avg_quarterly_sales,
                                                            left semi join
                                                                       (select d date
        sum sales,
         ctel.i manufact id
                                                                        from date dim
limit 100;
                                                            left semi join
                                                                       (select d week seq
-- end query 88 in stream 0 using template
                                                                      from date_dim
                                                               where d_date in ('1999-06-
query53.tpl
                                                     14','1999-08-26','1999-11-06')
```

```
) d3 on date dim.d_week_seq =
                                                               (p channel dmail = 'Y' or
                                                         and
                                                      p_channel_email = 'Y' or p_channel_tv =
d3.d_week_seq
      ) d\overline{2} on date dim.d date = d2.d date
                                                       'Y')
 where cr item sk = i item sk
                                                         and
                                                                s gmt offset = -7
 and cr_returned_date_sk = d_date_sk
                                                          and
                                                               d_year = 1999
                                                               d_moy = 12) promotional_sales
 group by i_item_id),
                                                          and
 wr items as
                                                       cross join
 (select i item id item id,
                                                         (select sum(ss ext sales price) total
        sum (wr return quantity) wr item qty
                                                          from store sales
 from web returns,
                                                               ,store
                                                               ,date_dim
      item,
      date_dim
                                                               ,customer
      left semi join
                                                               ,customer_address
                 (select d date
                                                               ,item
                                                          where ss_sold_date_sk = d_date_sk
                  from date dim
      left semi join
                                                          and ss_store_sk = s_store_sk
                                                                ss_customer_sk= c_customer_sk
                 (select d week seq
                                                          and
                                                                ca_address_sk = c_current_addr_sk
ss_item_sk = i_item_sk
ca_gmt_offset = -7
         from date_dim
where d_date in ('1999-06-
                                                          and
                                                          and
14','1999-08-26','1999-11-06')
                                                          and
                                                                i_category = 'Electronics'
      ) d3 on date_dim.d_week_seq =
                                                          and
                                                                s_{gmt_offset} = -7
d3.d week seq
                                                          and
                                                                d_year = 1999
      ) d2 on date_dim.d_date = d2.d_date
                                                          and
                                                               d moy = 12) all sales
 where wr item sk = i item sk
                                                          and
 and wr_returned_date_sk = d_date_sk
                                                      order by promotions, total
group by i_item_id)
select sr_items.item_id
                                                       limit 100;
                                                       -- end query 97 in stream 0 using template
       ,sr_item_qty
                                                       query61.tpl
                                                       -- start query 99 in stream 0 using
,sr_item_qty/(sr_item_qty+cr_item_qty+wr_it
em qty)/3.0 \times 100 sr dev
                                                       template query76.tpl and seed 945056756
       ,cr item qty
                                                       select channel, col_name, d_year, d_qoy,
                                                       i category, COUNT(*) sales_cnt,
,cr_item_qty/(sr_item_qty+cr_item_qty+wr_it
                                                       SUM(ext_sales_price) sales_amt FROM (
                                                               SELECT 'store' as channel,
em \overline{q}ty)/\overline{3}.0 * 100 cr \overline{d}ev
                                                       'ss_hdemo_sk' col_name, d_year, d_qoy,
       ,wr_item_qty
                                                       i_category, ss_ext_sales_price
,wr item qty/(sr item qty+cr item qty+wr it
                                                       ext_sales_price
em \overline{q}ty)/\overline{3.0} * 10\overline{0} wr \overline{d}ev
                                                                FROM store sales, item, date dim
                                                                WHERE ss_hdemo_sk IS NULL
                                                                  AND ss_sold_date_sk=d_date_sk
,(sr_item_qty+cr_item_qty+wr_item_qty)/3.0
average
                                                                  AND ss_item_sk=i_item_sk
from sr_items
                                                               UNION ALL
     ,cr_items
                                                               SELECT 'web' as channel,
                                                       'ws web page sk' col name, d_year, d_qoy,
     ,wr items
 where sr\_items.item\_id=cr\_items.item\_id
                                                      i_category, ws_ext_sales_price
   and sr_items.item_id=wr_items.item_id
                                                      ext sales price
 order by sr_items.item_id
                                                                FROM web sales, item, date dim
                                                                WHERE ws_web_page_sk IS NULL
         ,sr_item_qty
  limit 100;
                                                                  AND ws_sold_date_sk=d_date_sk
                                                                  AND ws_item_sk=i_item_sk
-- end query 96 in stream 0 using template
                                                               UNION ALL
                                                               SELECT 'catalog' as channel,
query83.tpl
-- start query 97 in stream 0 using
                                                       'cs_ship_addr_sk' col_name, d_year, d_qoy,
template query61.tpl and seed 1770420976
                                                      i_category, cs_ext_sales_price
select promotions, total, cast (promotions as
                                                      ext sales price
decimal(15,4))/cast(total as
                                                                FROM catalog sales, item, date dim
decimal(15,4))*100
                                                                WHERE cs_ship_addr_sk IS NULL
                                                                  AND cs_sold_date_sk=d_date_sk
                                                      AND cs_item_sk=i_item_sk) foo GROUP BY channel, col name, d year, d qoy,
  (select sum(ss_ext_sales_price)
promotions
   from store_sales
                                                       i category
        ,store
                                                      ORDER BY channel, col_name, d_year, d_qoy,
        ,promotion
                                                       i_category
        ,date dim
                                                       limit 100;
        ,customer
                                                       -- end query 99 in stream 0 using template
        ,customer address
         ,item
                                                      query76.tpl
   where ss_sold_date_sk = d_date_sk
        ss_store_sk = s_store_sk
   and
         ss\_promo\_sk = p\_promo\_sk
   and
                                                      E.3 Hive 0.13 Queries:
         ss_customer_sk= c_customer_sk
   and
         ca address sk = c current addr sk
   and
                                                       -- start query 1 in stream 0 using template
        ss_item_sk = i_item_sk
ca_gmt_offset = -7
   and
                                                      query96.tpl and seed 550831069
   and
```

i_category = 'Electronics'

and

select count(*) count1

from store sales

```
,household demographics
                                                    where inv1.i item sk = inv2.i item sk
                                                       and inv1.w_warehouse_sk =
    ,time_dim, store
where ss sold time sk = time dim.t time sk
                                                     inv2.w warehouse sk
                                                      and \overline{i}nv1.d moy=2
   and ss hdemo sk =
household demographics.hd demo sk
                                                      and inv2.d moy=2+1
    and ss_store_sk = s_store_sk
                                                    order by
                                                     inv1.w_warehouse_sk,inv1.i_item_sk,inv1.d_m
    and time_dim.t_hour = 15
    and time dim.t minute >= 30
                                                     oy, inv1.mean, inv1.cov
    and household demographics.hd dep count
                                                             ,inv2.d moy,inv2.mean, inv2.cov
    and store.s_store_name = 'ese'
                                                    with inv as
--order by count(*)
                                                     (select
order by count1
                                                     w warehouse name, w warehouse sk, i item sk, d
limit 100;
                                                     _moy
-- end query 1 in stream 0 using template
                                                            ,stdev,mean, case mean when 0 then
query96.tpl
                                                     null else stdev/mean end cov
-- start query 2 in stream 0 using template
                                                     from(select
query7.tpl and seed 997258328
                                                     w warehouse name, w warehouse sk, i item sk, d
select i item id,
                                                    moy
        avg(ss_quantity) agg1,
        avg(ss_list_price) agg2,
                                                     ,stddev_samp(inv_quantity_on_hand)
        avg(ss coupon amt) agg3,
                                                     stdev,avg(inv_quantity_on_hand) mean
                                                           from inventory
        avg(ss_sales_price) agg4
from store sales, customer demographics,
                                                               ,item
                                                               ,warehouse
date dim, item, promotion
 where ss sold date sk = d date sk and
                                                               ,date dim
       ss item sk = i item sk and
                                                           where inv item sk = i item sk
       ss\_cdemo\_sk = cd\_demo\_sk and
                                                             and inv_warehouse_sk =
       ss_promo_sk = p_promo_sk and
cd_gender = 'M' and
                                                     w warehouse sk
                                                             and inv_date_sk = d_date_sk
       cd marital status = 'W' and
                                                             and d year =\overline{2000}
       cd_education_status = '2 yr Degree'
                                                           aroup by
and
                                                     w_warehouse_name,w_warehouse_sk,i_item_sk,d
       (p_channel_email = 'N' or
                                                     _moy) foo
p channel event = 'N') and
                                                     where case mean when 0 then 0 else
       d_year = 1999
                                                     stdev/mean end > 1)
 group by i_item_id
                                                     select.
 order by i_item_id
                                                     inv1.w warehouse sk,inv1.i item sk,inv1.d m
  limit 100;
                                                     oy, inv1.mean, inv1.cov
                                                     ,inv2.w_warehouse_sk,inv2.i_item_sk,inv2.d
-- end query 2 in stream 0 using template
query7.tpl
                                                     moy, inv2.mean, inv2.cov
-- start query 5 in stream 0 using template
                                                     from inv inv1, inv inv2
query39.tpl and seed 1420791654
                                                     where inv1.i item sk = inv2.i item sk
                                                      and inv1.w warehouse_sk =
with inv as
(select
                                                     inv2.w_warehouse_sk
                                                      and \overline{i}nv1.d moy=2
w warehouse name, w warehouse sk,i item sk,d
                                                      and inv2.d moy=2+1
_moy
       ,stdev,mean, case mean when 0 then
                                                      and inv1.cov > 1.5
null else stdev/mean end cov
                                                     order by
from(select
                                                     inv1.w_warehouse_sk,inv1.i_item_sk,inv1.d_m
w warehouse name, w warehouse sk,i item sk,d
                                                    oy, inv1.mean, inv1.cov
                                                            ,inv2.d_moy,inv2.mean, inv2.cov
moy
,stddev samp(inv quantity on hand)
stdev, avg (inv quantity on hand) mean
                                                     -- end query 5 in stream 0 using template
      from inventory
                                                    query39.tpl
                                                     -- start query 7 in stream 0 using template
          ,item
          ,warehouse
                                                     query32.tpl and seed 944563352
                                                     select sum(cs.cs ext discount amt) as
          ,date dim
      where inv item sk = i item sk
                                                     excess discount amount
       and inv_warehouse_sk =
                                                    from
w_warehouse_sk
                                                        catalog_sales cs
        and inv_date_sk = d_date_sk
                                                        join date dim dd on dd.d date sk =
        and d year =2000
                                                     cs.cs sold date sk
      group by
                                                        join item i on i.i item sk =
                                                     cs.cs item_sk
w warehouse name, w warehouse sk, i item sk, d
_moy) foo
                                                       join
where case mean when 0 then 0 else
stdev/mean end > 1)
                                                              select
                                                                 cs2.cs_item_sk as cs_item_sk,
inv1.w warehouse sk,inv1.i item sk,inv1.d m
                                                                 1.3 7
oy, inv1.mean, inv1.cov
                                                     avg(cs2.cs ext discount amt) as
                                                     tmp_cs_ext_discount amt
,inv2.w_warehouse_sk,inv2.i_item_sk,inv2.d_
                                                              from
moy,inv2.mean, inv2.cov
                                                                 catalog sales cs2
from inv inv1, inv inv2
```

```
join date dim dd2 on
                                                            sum(case when
dd2.d date sk = cs2.cs sold date sk
                                                     (d day name='Wednesday') then
                                                     ss sales price else null end) wed sales,
         where
         dd2.d date between '2000-01-16'
                                                             sum(case when
         and date_add(cast('2000-01-16' as
                                                     (d day name='Thursday') then ss sales price
date), 90)
                                                     else null end) thu sales,
                                                            sum(case when (d day name='Friday')
         group by cs2.cs item sk
      ) tmp on tmp.cs item sk = i.i item sk
                                                     then ss sales price else null end)
                                                     fri sales,
where
                                                             sum(case when
i.i_manufact_id = 353
                                                     (d_day_name='Saturday') then ss_sales_price
and dd.d_date between '2000-01-16'
                                                     else null end) sat_sales
and date add(cast('2000-01-16' as date),
                                                      from date dim, store sales, store
                                                      where d date sk = ss sold date sk and
                                                            s_store_sk = ss_store_sk and
and cs.cs ext discount amt >
tmp.tmp_cs_ext_discount_amt
                                                            s_gmt_offset = -8 and
                                                            d_year = 1998
 limit 100;
                                                     group by s_store_name, s_store_id order by s_store_name,
-- end query 7 in stream 0 using template
querv32.tpl
                                                     s_store_id,sun_sales,mon_sales,tue_sales,we
 - start query 14 in stream 0 using
                                                     d_sales,thu_sales,fri_sales,sat_sales
template query21.tpl and seed 614834996
                                                       __limit 100;
select *
from(select w_warehouse_name
                                                     -- end query 15 in stream 0 using template
            ,i_item id
                                                     query43.tpl
            , sum(case when (cast(d date as
                                                      -- start query 16 in stream 0 using
date) < cast ('1998-06-27' as date))
                                                     template query27.tpl and seed 331218716
                        then
                                                     select i_item_id,
                                                             s_state,
inv_quantity_on_hand
                                                             --grouping(s_state) g_state,
                       else 0 end) as
inv before
                                                             GROUPING__ID g_state,
                                                             avg(ss quantity) agg1,
            , sum (case when (cast(d date as
date) >= cast ('1998-06-27' as date))
                                                             avg(ss_list_price) agg2,
                                                             avg(ss_coupon_amt) agg3,
                       then
inv_quantity_on_hand
                                                             avg(ss_sales_price) agg4
                       else 0 end) as
                                                      from store_sales, customer_demographics,
inv after
                                                     date dim, store, item
   from inventory
                                                      where ss sold date sk = d date sk and
                                                            ss item sk = i item sk and
      ,warehouse
       ,item
                                                            ss\_store\_sk = s\_store\_sk and
                                                            ss_cdemo_sk = cd_demo_sk and
cd_gender = 'F' and
       ,date_dim
   where i current price between 0.99 and
                                                            cd_marital_status = 'W' and
                                                            cd education status = '4 yr Degree'
     and i item sk
                             = inv item sk
     and inv_warehouse_sk
                                                     and
                                                            d_year = 1999 and
w_warehouse_sk
     and inv date sk = d date sk
                                                            s state in ('OH','IL', 'LA', 'GA',
                                                     'CO', 'AL')
     and d_date between date_sub(cast
                                                      group by i_item_id, s_state with ROLLUP order by i_item_id
('1998-06-\overline{27}') as date), 30)
                    and date add(cast
('1998-06-27' as date), 30)
                                                             ,s_state
  group by w warehouse name, i item id) x
                                                        limit 100;
where (case when inv_before > \overline{0}
                                                     -- end query 16 in stream 0 using template
             then inv_after / inv_before
             else null
                                                     query27.tpl
             end) between 2.0/3.0 and
                                                     -- start query 19 in stream 0 using
                                                     template query58.tpl and seed 1844319395
3.0/2.0
                                                     with ss_items as
order by w_warehouse_name
        ,i_item_id
                                                      (select i_item_id item_id
  limit 10\overline{0};
                                                             ,sum(ss_ext_sales_price)
                                                     ss item rev
-- end query 14 in stream 0 using template
                                                      from store sales
                                                          JOIN item ON ss_item_sk = i_item_sk
query21.tpl
-- start query 15 in stream 0 using
                                                          JOIN date_dim dd0 ON ss_sold_date_sk =
template query43.tpl and seed 959608359
                                                     dd0.d date sk
select s_store_name, s store id,
                                                         JOIN
       sum(case when (d_day_name='Sunday')
                                                             (select ddl.d date
then ss_sales_price else null end)
                                                             from date_dim dd1
        sum(case when (d_day_name='Monday')
                                                                 JOIN date dim dd2 ON
                                                     ddl.d_week_seq = dd2.d_week_seq
where dd2.d_date = '1998-05-29') v1
then ss_sales_price else null end)
mon sales,
                                                     ON dd0.d date = v1.\overline{d} date
       sum(case when
(d day name='Tuesday') then ss sales price
                                                     group by i_item_id),
else null end) tue_sales,
                                                     cs items as
                                                      (select i_item_id item_id
```

```
, sum(cs_ext_sales_price)
                                                    select distinct i1.i manufact id,
cs item rev
 from catalog sales
                                                    ss ext sales price
     JOIN item ON cs item sk = i item sk
     JOIN date dim dd0 ON cs sold date sk
                                                           store sales,
= dd0.d_date_sk
                                                           date dim,
    JOIN
                                                             customer address,
        (select dd1.d date
                                                             item i1.
                                                             item i2
        where
                                                             i1.i_manufact_id =
dd1.d_week_seq = dd2.d_week_seq
                                                    i2.i_manufact_id
        where dd2.d date = '1998-05-29') v1
                                                            i2.i category in ('Books')
                                                    and
ON dd0.d date = v1.d date
                                                    and
                                                            ss item_sk
                                                    i1.i\_item\_sk
group by i_item_id),
ws items as
                                                    and
                                                           ss_sold_date_sk
(select i item id item id
                                                    d date sk
       , sum (ws_ext_sales_price)
                                                    and
                                                            d year
                                                                                     = 2001
                                                                                     = 6
ws item rev
                                                    and
                                                             d mov
from web_sales
                                                    and
                                                             ss addr sk
                                                   ca_address sk
     JOIN item ON ws_item_sk = i_item_sk
    JOIN date_dim dd0 ON ws_sold_date_sk =
                                                    and ca_gmt_offset
dd0.d_date_sk
                                                    ) v1
     JOIN
                                                    group by i_manufact_id),
        (select dd1.d date
                                                    cs as (
                                                    select
        from date dim ddl
            JOIN date_dim dd2 ON
                                                    i_manufact_id,sum(cs_ext_sales_price)
dd1.d_week_seq = dd2.d_week_seq
                                                    total_sales
        where dd2.d date = '1998-05-29') v1
                                                    from
ON dd0.d date = v1.\overline{d} date
group by i_item_id)
                                                    select distinct il.i manufact id,
  select ss_items.item_id
                                                    cs_ext_sales_price
      ,ss item rev
                                                    from
                                                           catalog sales,
,ss_item_rev/(ss_item_rev+cs_item_rev+ws_it
                                                           date dim.
em rev)/\overline{3} * 100 ss dev
                                                             customer address,
      ,cs item rev
                                                             item i1,
                                                             item i2
,cs item rev/(ss_item_rev+cs_item_rev+ws_it
                                                    where
                                                    i1.i_manufact_id =
i2.i_manufact_id
em_rev)/3 * 100 cs_dev
       ,ws item rev
                                                    and
                                                           i2.i category in ('Books')
,ws item rev/(ss item rev+cs item rev+ws it
                                                    and
                                                            cs item sk
em rev)/\overline{3} * 100 ws dev
                                                    i1.i_item_sk
                                                    and
                                                            cs_sold_date_sk
,(ss item rev+cs item rev+ws item rev)/3
                                                    d date sk
                                                           d_year
                                                    and
                                                                                     = 2001
average
 from ss_items,cs_items,ws_items
                                                    and
                                                                                     = 6
                                                            d mov
where ss items.item id=cs items.item id
                                                            cs bill addr sk
                                                    and
   and ss_items.item_id=ws_items.item_id
                                                    ca address sk
   and ss item rev between 0.9 *
                                                    and
                                                           ca_gmt_offset
cs_item_rev and 1.1 * cs_item rev
                                                    ) v2
   and ss_item_rev between 0.\overline{9} *
                                                    group by i_manufact_id),
ws item rev and 1.1 * ws item rev
                                                    ws as (
   and cs item rev between 0.9 *
                                                    select
ss item rev and 1.1 * ss item rev
  and cs_item_rev between 0.9 *
                                                    i_manufact_id, sum (ws_ext_sales_price)
ws item \overline{rev} and 1.1 * ws item \overline{rev}
                                                    total_sales
   and ws item rev between 0.9 *
                                                    from
ss item rev and 1.1 * ss item rev
                                                    (
   and ws item rev between 0.9 *
                                                    select distinct il.i manufact id,
cs item rev and 1.1 * cs_item_rev
                                                    ws_ext_sales_price
order by item_id
                                                    from
         ,ss item rev
                                                           web sales,
  limit 100;
                                                           date dim,
                                                             customer_address,
-- end query 19 in stream 0 using template
                                                             item il.
query58.tpl
                                                             item i2
-- start query 22 in stream 0 using
                                                    where
template query33.tpl and seed 248487088
                                                             i1.i_manufact_id =
                                                    i2.i_manufact_id
with ss as (
select
                                                    and
                                                          i2.i category in ('Books')
                                                    and
                                                             ws item sk
i manufact_id,sum(ss_ext_sales_price)
                                                    i1.i_item_sk
                                                    and ws_sold_date_sk
total_sales
                                                    d_date_sk
from
```

```
d year
                                   = 2001
                                                                    ,avg(sum(ss sales price)) over
and
                                                      (partition by i_manager_id)
and
         d moy
                                   = 6
         ws bill addr sk
                                                      avg monthly sales
 and
                                                            from \overline{i}tem
ca address sk
                                                                ,store_sales
and
         ca_gmt_offset
                                                                ,date_dim
 ) v3
 group by i_manufact_id)
                                                                 ,store
  select i manufact id , sum(total sales)
                                                            where ss item sk = i item sk
                                                              and ss sold date sk = d date sk
total sales
 \operatorname{from}^- (select * from ss
                                                              and ss_store_sk = s_store sk
        union all
                                                              and d_month_seq in
        select * from cs
                                                      (1178, 1178+1, 1178+2, 1178+3, 1178+4, 1178+5, 11
        union all
                                                      78+6,1178+7,1178+8,1178+9,1178+10,1178+11)
                                                      and (( i_category in
('Books','Children','Electronics')
        select * from ws) tmp1
 group by i_manufact_id
 order by total_sales
                                                                    and i class in
  limit 100;
                                                      ('personal', 'portable', 'refernece', 'self-
                                                      help')
-- end query 22 in stream 0 using template
                                                                     and i brand in
query33.tpl
                                                      ('scholaramalgamalg #14','scholaramalgamalg
 - start query 24 in stream 0 using
template query62.tpl and seed 800775315
                                                      'exportiunivamalg #9','scholaramalgamalg
select
   substr(w warehouse name, 1, 20)
                                                      #9'))
w warehouse name
                                                                         i category in
                                                                  or(
 ,sm type
                                                      ('Women','Music','Men')
  ,web name
                                                                    and i class in
                                                      ('accessories','classical','fragrances','pa\\
  ,sum(case when (ws_ship_date_sk -
ws sold date sk <= 30 ) then 1 else 0 end)
                                                      nts')
                                                                     and i brand in ('amalgimporto
as 30 davs
 , sum (case when (ws ship date sk -
                                                      #1','edu packscholar #1','exportiimporto
ws_sold_date_sk > 30) and
                                                      #1',
                  (ws_ship_date_sk -
ws_sold_date_sk \le 60) then 1 else 0 end )
                                                      'importoamalg #1')))
as 31 60 days
                                                      group by i_manager_id, d_moy) tmp1
, sum(case when (ws_ship_date_sk -
ws_sold_date_sk > 60) and
                                                      where case when avg_monthly_sales > 0 then abs (sum_sales - avg_monthly_sales) /
                  (ws ship date sk -
                                                      avg monthly sales else null end > 0.1
ws sold date sk \leq 90) then 1 else 0 end)
                                                      order by i manager id
as 61 90 days
                                                              ,avg_monthly_sales
  ,sum(case when (ws_ship_date_sk -
                                                               sum_sales,
ws_sold_date_sk > 90) and
                                                       limit 100;
                  (ws ship date sk -
ws sold date sk \leq 120) then 1 else 0 end)
                                                      -- end query 27 in stream 0 using template
as 91 120 days
                                                      query63.tpl
                                                      -- start query 28 in stream 0 using
  , sum(case when (ws_ship_date_sk -
ws sold date sk > 120) then 1 else 0 end)
                                                      template query69.tpl and seed 1390437346
as above120 days
                                                      select
                                                        cd gender,
from
  web sales
                                                        cd marital status,
  ,warehouse
                                                        cd education status,
  , ship mode
                                                        count(*) cnt1,
  ,web site
                                                        cd purchase estimate,
                                                        count(*) cnt2,
  ,date dim
where
                                                        cd credit rating,
    d month seq between 1201 and 1201 + 11
                                                        count(*) cnt3
and ws ship date sk = d date sk
                                                       from
                      = w_warehouse_sk
and ws_warehouse_sk
                                                        customer c
and ws ship mode sk
                      = sm_ship_mode_sk
                                                        JOIN customer_address ca ON
and ws web site sk
                       = web site sk
                                                      c.c current addr sk = ca.ca address sk
                                                        JOIN customer demographics ON cd_demo_sk
group \overline{b}y
                                                      = c.c_current_cdemo_sk
LEFT SEMI JOIN
  substr(w warehouse name, 1, 20)
  ,sm_type
                                                        (select ss customer sk
  ,web name
order by substr(w warehouse name, 1,20)
                                                                 from store sales, date dim
                                                                where --c.c_customer_sk =
        ,sm type
       ,web name
                                                      ss_customer_sk and
  limit 100;
                                                                       ss_sold_date_sk = d_date_sk
                                                      and
-- end query 24 in stream 0 using template
                                                                       d year = 2004 and
                                                                       d moy between 3 and 3+2)
query62.tpl
-- start query 27 in stream 0 using
                                                      ssdd
template query63.tpl and seed 812633773
                                                        ON c.c customer sk = ssdd.ss customer sk
select *
                                                        LEFT OUTER JOIN
from (select i manager id
                                                         (select ws bill customer sk
              ,sum(ss_sales_price) sum_sales
                                                                   from web_sales,date_dim
```

```
where --c.c customer_sk =
                                                    select distinct il.i item id,
ws_bill_customer_sk and
                                                    cs ext sales price
                  ws sold date sk =
                                                     from
                                                            catalog sales,
d date sk and
                  d year = 2004 and
                                                            date dim,
                  d moy between 3 and 3+2)
                                                             customer_address,
wsdd
                                                              item i1,
 ON c.c customer sk =
                                                              item i2
wsdd.ws bill customer sk
                                                     where
 LEFT OUTER JOIN
                                                             i1.i_item_id = i2.i_item_id
i2.i_category in ('Jewelry')
    (select cs_ship_customer sk
                                                     and
            from catalog_sales,date_dim
                                                     and
                                                             cs item sk
                                                    i1.i_item_sk
            where --c.c customer sk =
cs_ship_customer_sk and
                                                    and cs_sold_date_sk
                  cs sold date sk =
                                                    d date sk
                                                                                      = 2002
d_date_sk and
                                                     and
                                                             d year
                  d year = 2004 and
                                                                                      = 10
                                                     and
                                                             d moy
                  d moy between 3 and 3+2)
                                                     and
                                                             cs_bill_addr_sk
csdd --)
                                                    ca address sk
                                                                                      = -5
 ON c.c customer sk =
                                                     and
                                                             ca gmt offset
csdd.cs_ship_customer_sk
                                                     ) v2
                                                     group by i item id),
where
                                                     ws as (
 ca state in ('AL','VA','GA') and
                                                     select
  wsdd.ws bill customer sk is null and
                                                               i item id, sum (ws ext sales price)
 csdd.cs ship customer sk is null
                                                    total sales
group by cd_gender,
                                                     from
          cd_marital_status,
                                                      (
          cd education status,
                                                     select distinct il.i item id,
          cd purchase estimate,
                                                    ws ext sales price
          cd credit rating
                                                     from
order by cd gender,
                                                            web sales,
          cd_marital_status,
                                                            date_dim,
          cd education status,
                                                             customer address,
          cd purchase estimate,
                                                             item i1,
          cd_credit_rating
                                                             item i2
   limit. 100;
                                                     where
                                                             i1.i item id = i2.i item id
-- end query 28 in stream 0 using template
                                                             i2.i category in ('Jewelry')
                                                     and
                                                             ws_item_sk
query69.tpl
                                                     and
-- start query 29 in stream 0 using
                                                    i1.i_item_sk
template query60.tpl and seed 374071684
                                                     and
                                                             ws_sold_date_sk
with ss as (
                                                    d date_sk
                                                           d_year
                                                                                      = 2002
select
                                                     and
                                                                                      = 10
          i_item_id,sum(ss_ext_sales_price)
                                                     and
                                                             d mov
total_sales
                                                     and
                                                             ws_bill_addr_sk
                                                    ca address sk
from
                                                    and
                                                             ca gmt offset
select distinct i1.i_item_id,
                                                     ) v3
ss ext sales price
                                                     group by i item id)
                                                      select
       store sales,
                                                      i item id
                                                    ,sum(total_sales) total sales
       date dim,
         customer address,
                                                     from (select * from s\bar{s}
                                                            union all
         item il.
         item i2
                                                            select * from cs
                                                            union all
where
         i1.i_item_id = i2.i_item id
                                                            select * from ws) tmp1
                                                     group by i_item_id
order by i item id
and
         i2.i_category in ('Jewelry')
        ss item sk
 and
i1.i item s\bar{k}
                                                         ,total sales
and
        ss_sold_date_sk
                                                       limit 100;
d_date_sk
      d_year
and
                                 = 2002
                                                    -- end query 29 in stream 0 using template
                                 = 10
and
        d moy
                                                    querv60.tpl
                                                    -- start query 30 in stream 0 using
        ss addr sk
and
ca address_sk
                                                    template query59.tpl and seed 1976435349
       ca_gmt_offset
                                  = -5
and
                                                    with wss as
) v1
                                                     (select d_week_seq,
                                                            ss_store_sk,
sum(case when (d_day_name='Sunday')
group by i_item_id),
cs as (
select.
                                                    then ss sales price else null end)
          i item id, sum(cs ext sales price)
                                                    sun_sales,
total sales
                                                           sum(case when (d day name='Monday')
                                                    then ss_sales_price else null end)
from
 (
                                                    mon_sales,
```

sum(case when	from
(d_day_name='Tuesday') then ss_sales_price	(
else null end) tue_sales,	select i_item_id
sum(case when	,i item desc
(d day name='Wednesday') then	,i category
ss sales price else null end) wed sales,	,i class
sum(case when	,i current price
(d day name='Thursday') then ss sales price	, sum (ss ext sales price) as
else null end) thu_sales,	itemrevenue
<pre>sum(case when (d_day_name='Friday')</pre>	/
then ss_sales_price else null end)	sum(ss_ext_sales_price)*100/sum(sum(ss_ext_
fri_sales,	_sales_price)) over
sum(case when	(partition by i_class) as
(d_day_name='Saturday') then ss_sales_price	revenueratio
else null end) sat sales	from
from store sales, date dim	store sales
where d date sk = ss sold date sk	,item
group by d week seq,ss store sk	,date dim
)	where
,	
select	ss_item_sk = i_item_sk
s_store_name1,s_store_id1,d_week_seq1	and i_category in ('Shoes',
	'Music', 'Home')
sun_sales1/sun_sales2,mon_sales1/mon_sales,	and ss_sold_date_sk = d_date_sk
2	and d date between cast('1999-05-
	25' as date)
,tue sales1/tue sales1,wed sales1/wed sales	and
2,thu sales1/thu sales2	date add(cast('1999-05-25' as date), 30)
2, 0114_041001, 0114_041001	group by
fri calcal/fri calcal cat calcal/cat calca	
,fri_sales1/fri_sales2,sat_sales1/sat_sales	i_item_id
2	,i_item_desc
from	,i_category
(select s_store_name	,i_class
s_store_name1,wss.d_week_seq d_week_seq1	,i_current_price
s_store_id s_store_id1,sun_sales,	order by
sun sales1	i category
,mon sales mon sales1,tue sales	,i class
tue sales1	,i item id
, wed sales wed sales1, thu sales	,i item desc
thu sales1	,revenueratio
,fri_sales fri_sales1,sat_sales) z;
sat_sales1	
from wss,store,date_dim d	end query 32 in stream 0 using template
where d.d_week_seq = wss.d_week_seq and	query98.tpl
ss_store_sk = s_store_sk and	start query 36 in stream 0 using
d_month_seq between 1189 and 1189 +	template query28.tpl and seed 24799953
11) y,	select *
(select s store name	from (select avg(ss list price) B1 LP
s store name2, wss.d week seq d week seq2	,count(ss list price) B1 CNT
s_store_id s_store_id2,sun_sales	<pre>,count(distinct ss list price)</pre>
sun sales2	
-	B1_CNTD
,mon_sales mon_sales2,tue_sales	from store_sales
tue_sales2	where ss_quantity between 0 and 5
<pre>,wed_sales wed_sales2,thu_sales</pre>	and (ss_list_price between 47 and
thu_sales2	47+10
fri sales fri sales2,sat sales,	or ss coupon amt between 11713
sat sales2	and 11713+1000
from wss,store,date dim d	or ss wholesale cost between
where d.d week seq = wss.d week seq and	55 and 55+20)) B1,
ss store sk = s store sk and	(select avg(ss list price) B2 LP
d_month_seq between 1189+ 12 and	<pre>,count(ss_list_price) B2_CNT</pre>
1189 + 23) x	<pre>, count (distinct ss_list_price)</pre>
where s_store_idl=s_store_id2	B2_CNTD
and d_week_seq1=d_week_seq2-52	from store_sales
order by	where ss quantity between 6 and 10
s store namel,s store idl,d week seql	and (ss list price between 93 and
limit 100;	93+10
	or ss coupon amt between 7733 and
end query 30 in stream 0 using template	7733+1000
query59.tpl	or ss_wholesale_cost between 43
start query 32 in stream 0 using	and 43+20)) B2,
template query98.tpl and seed 1900673199	(select avg(ss_list_price) B3_LP
select i_item_desc	<pre>,count(ss_list_price) B3_CNT</pre>
,i_category	<pre>, count (distinct ss_list_price)</pre>
,i class	B3 CNTD
,i current price	from store sales
,itemrevenue	where ss quantity between 11 and 15
, revenueratio	13_400.010, 200,0011 11 0110 10
,	

```
and (ss_list_price between 32 and
                                                            , sum (may sales/w warehouse sq ft)
32+10
                                                     as may_sales_per_sq_foot
          or ss coupon amt between 11517
                                                            ,sum(jun sales/w warehouse sq ft)
and 11517+1000
                                                     as jun sales per sq foot
         or ss_wholesale_cost between 26
                                                    ___,sum(jul_sales/w_warehouse_sq_ft) as jul_sales_per_sq_foot
and 26+20)) B3,
     (select avg(ss list price) B4 LP
                                                            ,sum(aug_sales/w_warehouse_sq_ft)
           , count(ss_list_price) B4 CNT
                                                     as aug sales per sq foot
                                                            ,sum(sep sales/w warehouse sq ft)
            ,count(distinct ss list price)
B4 CNTD
                                                     as sep_sales_per_sq_foot
      from store_sales
                                                            ,sum(oct_sales/w_warehouse_sq_ft)
      where ss\_quantity between 16 and 20
                                                     as oct_sales_per_sq_foot
                                                            ,sum(nov_sales/w_warehouse_sq ft)
        and (ss list price between 147 and
                                                     as nov_sales_per_sq_foot
                                                            ,sum(dec_sales/w_warehouse_sq_ft)
          or ss\_coupon\_amt between 509 and
509+1000
                                                     as dec_sales_per_sq_foot
          or ss_wholesale_cost between 78
                                                            ,sum(jan net) as jan net
                                                            ,sum(feb_net) as feb_net
,sum(mar_net) as mar_net
and 78+20)) B4,
     (select avg(ss_list_price) B5_LP
            ,count(ss_list_price) B5_CNT
                                                            ,sum(apr_net) as apr_net
            ,count(distinct ss_list_price)
                                                            ,sum(may_net) as may_net
B5 CNTD
                                                            ,sum(jun net) as jun net
                                                            ,sum(jul_net) as jul_net
      from store_sales
      where ss\_quantity between 21 and 25
                                                            , sum (aug net) as aug net
                                                            ,sum(sep_net) as sep_net
        and (ss list price between 16 and
                                                            ,sum(oct net) as oct net
                                                            , sum(nov_net) as nov net
          or ss coupon amt between 2401 and
2401+1000
                                                            ,sum(dec_net) as dec_net
          or ss_wholesale_cost between 32
                                                      from (
and 32+20)) B5,
                                                        select
     (select avg(ss list price) B6 LP
                                                            w warehouse name
            , count(ss_list_price) B6_CNT
                                                            ,w_warehouse_sq ft
            , count(distinct ss_list_price)
                                                            ,w_city
B6 CNTD
                                                            ,w_county
                                                            ,w_state
      from store sales
      where ss\_quantity between 26 and 30
                                                            ,w_country
                                                            concat('MSC', ',', 'USPS') as
        and (ss list price between 11 and
11+10
                                                     ship_carriers
          or ss coupon amt between 916 and
                                                           ,d year as year
                                                            or ss\_wholesale\_cost between 6 and 6+20)) B6
                                                                   then ws_sales_price*
                                                     ws quantity else 0 end) as jan sales
limit 100;
                                                           ,sum(case when d_moy = 2
                                                                    then ws sales price*
                                                     ws_quantity else 0 end) as feb sales
-- end query 36 in stream 0 using template
query28.tpl
                                                            , sum(case when d_moy = 3)
 - start query 39 in stream 0 using
                                                                    then ws sales price*
template query66.tpl and seed 1688498284
                                                     ws quantity else 0 end) as mar sales
                                                            , sum(case when d moy = \frac{1}{4}
select.
                                                                   then ws_sales_price*
         w warehouse name
                                                     ws_quantity else 0 end) as apr_sales
       ,w_warehouse_sq_ft
                                                            , sum(case when d_moy = 5
       ,w city
       ,w_county
                                                                   then ws_sales_price*
                                                     ws quantity else 0 end) as may_sales
       ,w_state
       ,w country
                                                            , sum(case when d moy = 6
        , ship carriers
                                                                   then ws sales price*
                                                     ws quantity else 0 end) as jun sales
        ,year
       ,sum(jan_sales) as jan sales
                                                            , sum (case when d_{moy} = 7
       ,sum(feb_sales) as feb_sales
                                                                    then ws_sales_price*
       ,sum(mar sales) as mar sales
                                                     ws quantity else 0 end) as jul sales
       ,sum(apr sales) as apr sales
                                                            , sum (case when d moy = 8
                                                    then ws_sales_price* ws_quantity else 0 end) as aug_sales
       ,sum(may_sales) as may_sales
       ,sum(jun_sales) as jun_sales
       ,sum(jul_sales) as jul_sales
                                                           , sum(case when d_moy = 9
       ,sum(aug sales) as aug sales
                                                                    then ws sales price*
                                                     ws quantity else 0 end) as sep_sales
       ,sum(sep sales) as sep sales
        sum(oct_sales) as oct_sales
                                                            , sum (case when d moy = 10
        , sum(nov_sales) as nov_sales
                                                                   then ws_sales_price*
       ,sum(dec_sales) as dec_sales
                                                     ws_quantity else 0 end) as oct_sales
                                                            , sum (case when d moy = \overline{11}
        , sum(jan_sales/w_warehouse_sq_ft)
as jan_sales_per_sq_foot
                                                                    then ws_sales_price*
                                                    ws quantity else 0 end) as nov sales
       ,sum(feb_sales/w_warehouse_sq_ft)
as feb_sales_per_sq_foot
                                                           , sum (case when d moy = 12
       , sum (mar_sales/w_warehouse_sq_ft)
                                                                   then ws sales price*
as mar_sales_per_sq_foot
                                                    ws_quantity else 0 end) as dec_sales
                                                            , sum(case when d_moy = 1)
        , sum(apr_sales/w_warehouse_sq_ft)
```

as apr_sales_per_sq_foot

then	,w_county
ws_net_paid_inc_ship_tax * ws_quantity else	,w_state
0 end) as jan_net	,w_country
<pre>, sum (case when d_moy = 2</pre>	<pre>,concat('MSC', ',', 'USPS') as</pre>
then	ship_carriers
ws_net_paid_inc_ship_tax * ws_quantity else	,d_year as year
0 end) as feb_net	<pre>,sum(case when d_moy = 1</pre>
$, sum(case when d_moy = 3)$	then cs_ext_sales_price*
then	cs_quantity else 0 end) as jan_sales
ws_net_paid_inc_ship_tax * ws_quantity else	<pre>, sum(case when d_moy = 2</pre>
0 end) as mar_net	then cs_ext_sales_price*
<pre>, sum (case when d_moy = 4</pre>	cs_quantity else 0 end) as feb_sales
then	$, sum(case when d_moy = 3)$
ws_net_paid_inc_ship_tax * ws_quantity else	then cs_ext_sales_price*
0 end) as apr_net	cs_quantity else 0 end) as mar_sales
<pre>, sum(case when d_moy = 5</pre>	<pre>, sum(case when d_moy = 4</pre>
then	then cs_ext_sales_price*
ws_net_paid_inc_ship_tax * ws_quantity else	cs_quantity else 0 end) as apr_sales
0 end) as may_net	$, sum(case when d_moy = 5)$
<pre>, sum(case when d_moy = 6</pre>	then cs_ext_sales_price*
then	cs_quantity else 0 end) as may_sales
ws_net_paid_inc_ship_tax * ws_quantity else	<pre>, sum(case when d_moy = 6</pre>
0 end) as jun_net	then cs_ext_sales_price*
, sum(case when d moy = 7)	cs quantity else 0 end) as jun sales
then	, sum (case when d moy = 7
ws net paid inc ship tax * ws quantity else	then cs ext sales price*
0 end) as jul net	cs quantity else 0 end) as jul sales
sum(case when d moy = 8	,sum(case when d moy = 8
then	then cs ext sales price*
ws net paid inc ship tax * ws quantity else	cs quantity else 0 end) as aug sales
0 end) as aug net	, sum(case when d moy = 9
sum(case when d moy = 9	then cs ext sales price*
then	cs quantity else 0 end) as sep sales
ws net paid inc ship tax * ws quantity else	, sum(case when d moy = 10
0 end) as sep net	then cs ext sales price*
sum(case when d moy = 10	cs quantity else 0 end) as oct sales
then	sum(case when d moy = 11
ws net paid inc ship tax * ws quantity else	then cs ext sales price*
0 end) as oct net	cs quantity else 0 end) as nov sales
, sum(case when d moy = 11	sum(case when d moy = 12
then	then cs ext sales price*
ws net paid inc ship tax * ws quantity else	cs quantity else 0 end) as dec sales
0 end) as nov net	sum(case when d moy = 1
sum(case when d moy = 12	then cs net profit *
then	cs quantity else 0 end) as jan net
ws net paid inc ship tax * ws quantity else	, sum(case when d moy = 2
0 end) as dec net	then cs net profit *
from	cs quantity else 0 end) as feb net
web sales	sum(case when d moy = 3
,warehouse	then cs_net_profit *
, date dim	cs quantity else 0 end) as mar net
, time dim	, sum(case when d moy = 4
_	then cs net profit *
<pre>,ship_mode where</pre>	cs quantity else 0 end) as apr net
ws warehouse sk =	sum(case when d moy = 5
ws_warehouse_sk - w warehouse sk	then cs net profit *
and ws sold date sk = d date sk	cs quantity else 0 end) as may net
and ws_sold_date_sk = d_date_sk and ws_sold_time_sk = t_time_sk	-
	<pre>, sum(case when d_moy = 6</pre>
and ws_ship_mode_sk =	then cs_net_profit *
sm_ship_mode_sk	cs_quantity else 0 end) as jun_net
and d_y ear = 2002	, sum (case when d_moy = 7
and t_time between 18036 and	then cs_net_profit *
18036+28800	cs_quantity else 0 end) as jul_net
and sm_carrier in ('MSC','USPS')	, sum (case when d_moy = 8
group by	then cs_net_profit *
w_warehouse_name	cs_quantity else 0 end) as aug_net
<pre>, w_warehouse_sq_ft</pre>	, sum (case when $d_{moy} = 9$
,w_city	then cs_net_profit *
,w_county	cs_quantity else 0 end) as sep_net
,w_state	, sum (case when $d_{moy} = 10$
,w_country	then cs_net_profit *
,d_year	cs_quantity else 0 end) as oct_net
union all	<pre>, sum(case when d_moy = 11</pre>
select	then cs_net_profit *
w_warehouse_name	cs_quantity else 0 end) as nov_net
<pre>, w_warehouse_sq_ft</pre>	$, sum(case when d_moy = 12)$
,w city	

```
then cs net profit *
                                                    limit 100;
cs quantity else 0 end) as dec net
                                                   -- end query 40 in stream 0 using template
     from
         catalog sales
                                                   query90.tpl
                                                   -- start query 44 in stream 0 using
         ,warehouse
                                                   template query92.tpl and seed 643980925
         ,date_dim
        ,time_dim
                                                   select
        , ship mode
                                                      sum(ws.ws ext discount amt)
     where
                                                   Excess Discount Amount
           cs_warehouse sk =
                                                   from
w_warehouse_sk
                                                       web_sales ws join item i1 on
        and cs_sold_date_sk = d_date_sk
                                                   i1.i_item_sk = ws.ws_item_sk
        and cs sold time sk = t time sk
                                                       join date dim dd on dd.d date sk =
                                                   ws.ws sold date_sk
       and cs_ship_mode_sk =
sm ship mode sk
                                                       join
        and \overline{d}_year = 2002
                                                         (
        and t_{\text{time}} between 18036 and
                                                          SELECT
18036+28800
                                                             ws item sk, 1.3 *
       and sm_carrier in ('MSC','USPS')
                                                   avg(ws_ext_discount_amt) as
     group by
                                                   avg ws ext discount amt
       w_warehouse_name
                                                          FROM
       ,w_warehouse_sq_ft
                                                            web sales join date dim on
       ,w_city
                                                   d_date_sk = ws_sold_date_sk
       ,w county
                                                          WHERE
       ,w state
                                                             d date between '1999-01-03'
       ,w country
                                                             and date add(cast('1999-01-03' as
                                                   date), 90)
       ,d_year
 ) x
                                                             group by ws_item_sk
                                                          ) tmp on tmp.ws_item_sk =
 group by
       w warehouse name
                                                   il.i item sk
       ,w warehouse sq ft
                                                   where
                                                   i1.i manufact id = 926
       ,w_city
                                                   and \overline{d}d.d_{date} between '1999-01-03'
       ,w_county
       ,w_state
                                                   and date_add(cast('1999-01-03' as date),
       ,w_country
                                                   and ws.ws_ext_discount_amt >
tmp.avg_ws_ext_discount_amt
       ,ship_carriers
       ,year
 order by w warehouse name
                                                   order by Excess Discount Amount
   limit 10\overline{0};
                                                     limit 100;
-- end query 39 in stream 0 using template
                                                   -- end query 44 in stream 0 using template
query66.tpl
                                                   query92.tpl
-- start query 40 in stream 0 using
                                                    -- start query 45 in stream 0 using
template query90.tpl and seed 1949014749
                                                   template query3.tpl and seed 691662667
select cast(amc as decimal(15,4))/cast(pmc
                                                   select dt.d year
as decimal(15,4)) am_pm_ratio
                                                          ,item.i_brand_id brand_id
from ( select count(*) amc
                                                          ,item.i brand brand
       from web sales,
                                                          ,sum(ss_net_profit) sum_agg
from date_dim dt
       where ws sold time sk =
                                                         ,store sales
time_dim.t_time_sk
                                                         ,item
        and ws ship hdemo sk =
                                                    where dt.d date sk =
household_demographics.hd_demo_sk
                                                   store sales.ss sold date sk
                                                      and store_sales.ss_item_sk =
         and ws_web_page_sk =
web page.wp web page sk
                                                   item.i item sk
         and time dim.t hour between 11 and
                                                      and item.i manufact id = 596
                                                      and dt.d moy=12
                                                    group by dt.d_year
         and
household_demographics.hd_dep_count = 9
                                                        ,item.i_brand
         and web_page.wp_char_count between
                                                         ,item.i brand id
5000 and 5200) at,
                                                    order by dt.d year
      ( select count(*) pmc
                                                            ,sum agg desc
       from web_sales,
                                                             ,brand_id
household_demographics , time_dim, web_page
                                                     limit 100;
       where ws sold time sk =
time_dim.t time sk
                                                   -- end query 45 in stream 0 using template
        and ws_ship_hdemo_sk =
                                                   query3.tpl
                                                   -- start query 49 in stream 0 using
household_demographics.hd_demo_sk
        and ws_web_page_sk =
                                                   template query9.tpl and seed 937436805
with temp1 as
                                                     select 'bucket1' bucket, count(*) cnt,
18 + 1
                                                   avg(ss ext sales price) avg amt,
                                                   avg(ss net paid_inc_tax) avg_paid
household demographics.hd dep count = 9
         and web_page.wp_char_count between
                                                     from store sales
5000 and 5200) pt
                                                     where ss_quantity between 1 and 20
 order by am_pm_ratio
```

```
temp2 as
                                                   else (ss quantity*ss sales price) end
  select 'bucket2' bucket, count(*) cnt,
                                                   act_sales
avg(ss ext sales price) avg amt,
                                                                from store sales left outer
                                                   join store_returns on (sr item sk =
avg(ss_net_paid_inc_tax) avg_paid
                                                   ss_item_sk
 from store_sales
 where ss \overline{quantity} between 21 and 40
                                                   and sr ticket number = ss ticket number)
temp3 as
                                                                  ,reason
                                                               where sr_reason_sk =
 select 'bucket3' bucket, count(*) cnt,
                                                   r_reason_sk
avg(ss_ext_sales_price) avg_amt,
                                                                 and r reason desc = 'reason
avg(ss net paid inc tax) avg paid
                                                   74') t
 from store sales
                                                         group by ss customer sk
 where ss quantity between 41 and 60
                                                         order by sumsales, ss_customer_sk
                                                    limit 100;
temp4 as
                                                   -- end query 52 in stream 0 using template
 select 'bucket4' bucket, count(*) cnt,
                                                   query93.tpl
                                                   -- start query 55 in stream 0 using
avg(ss_ext_sales_price) avg_amt,
avg(ss_net_paid_inc_tax) avg_paid
                                                   template query22.tpl and seed 635815297
                                                   select i_product_name
 from store sales
 where ss quantity between 61 and 80
                                                                ,i_brand
                                                                ,i_class
temp5 as
                                                                ,i category
                                                                 ,avg(inv quantity on hand) qoh
(
 select 'bucket5' bucket, count(*) cnt,
                                                          from inventory
avg(ss_ext_sales_price) avg_amt,
                                                              ,date_dim
avg(ss_net_paid_inc_tax) avg_paid
                                                              ,item
 from store sales
                                                              ,warehouse
                                                          where inv_date_sk=d_date_sk and inv_item_sk=i_item_sk
 where ss quantity between 81 and 100
                                                                 and inv_warehouse_sk =
                                                   w warehouse sk
     case when (temp1.bucket = 'bucket1'
                                                                 and d month seq between 1199
and temp1.cnt > 62316685)
                                                   and 1199 + 11
           then temp1.avg amt
                                                        group by i_product_name
            else temp1.avg paid
                                                                          ,i brand
            end bucket1
                                                                           ,i class
                                                                           ,i category WITH
      case when (temp2.bucket = 'bucket2'
and temp2.cnt > 19045798)
                                                   ROLLUP
            then temp2.avg amt
                                                   order by qoh, i product name, i brand,
                                                   i_class, i_category limit 100;
            else temp2.avg paid
            end bucket2 ,
      case when (temp3.bucket = 'bucket3'
and temp3.cnt > 365541424)
                                                   -- end query 55 in stream 0 using template
            then temp3.avg amt
                                                   query22.tpl
                                                    -- start query 56 in stream 0 using
            else temp3.avg_paid
            end bucket3 ,
                                                   template query89.tpl and seed 2079706651
      case when (temp4.bucket = 'bucket4'
                                                   select
and temp4.cnt > 216357808)
                                                   from(
                                                   select i_category, i_class, i_brand,
            then temp4.avg amt
            else temp4.avg_paid
                                                          s_store_name, s_company_name,
            end bucket4
                                                          d moy,
       case when (temp5.bucket = 'bucket5'
                                                          sum(ss sales price) sum sales,
and temp5.cnt > 184483884)
                                                           avg(sum(ss sales price)) over
           then temp5.avg amt
                                                            (partition by i_category, i_brand,
            else temp5.avg_paid
                                                   s_store_name, s_company_name)
            end bucket5
                                                            avg_monthly_sales
                                                    from item, store sales, date dim, store
                                                   where ss item sk = i item sk and
from temp1, temp2, temp3, temp4, temp5
                                                         ss_sold_date_sk = d_date_sk and
ss_store_sk = s_store_sk and
                                                         d_year in (1999) and
-- end query 49 in stream 0 using template
                                                            ((i category in
-- start query 52 in stream 0 using
                                                    ('Books','Jewelry','Men') and
template query93.tpl and seed 1821797098
                                                             i class in
                                                    ('history', 'birdal', 'pants')
select ss_customer_sk
           ,sum(act_sales) sumsales
      or (i category in
                                                    ('Music','Home','Shoes') and
                  ,ss_customer_sk
                                                            i_class in
                  ,case when
                                                    ('pop', 'furniture', 'athletic')
sr return quantity is not null then
                                                           ))
(ss quantity-
                                                   group by i_category, i_class, i_brand,
sr_return_quantity) *ss_sales_price
                                                           s_store_name, s_company_name,
```

```
where case when (avg monthly sales <> 0)
                                                       and d2.d moy = 10
then (abs(sum_sales - avg_monthly_sales) /
                                                       and ss_ticket_number = sr_ticket_number
avg monthly sales) else \overline{\text{null}} end > 0.1
                                                       and ss item sk = sr item sk
order by sum sales - avg monthly sales,
                                                       and ss sold date sk = d1.d date sk
s_store_name limit 100;
                                                       and sr\_returned\_date\_sk = d2.d\_date\_sk and ss\_customer\_sk = sr\_customer\_sk
                                                       and ss store sk = s store sk
-- end query 56 in stream 0 using template
                                                       aroup by
                                                          s store name
-- start query 59 in stream 0 using template query52.tpl and seed 223505300
                                                          ,s_company id
                                                          ,s_street_number
select dt.d_year
                                                          ,s_street_name
        ,item.i brand id brand id
                                                          ,s street type
        ,item.i brand brand
                                                          ,s suite number
        ,sum(ss_ext_sales_price) ext_price
                                                          ,s_city
                                                         ,s_county
 from date dim dt
     ,store_sales
                                                          ,s state
     ,item
                                                          ,s_zip
 where dt.d date sk =
                                                       order by s_store_name
store sales.ss sold date sk
                                                               ,s company id
                                                                ,s_street number
    and store_sales.ss_item_sk =
item.i_item sk
                                                                ,s street name
    and item.i_manager_id = 1
                                                                ,s_street_type
    and dt.d moy=11
                                                                ,s suite number
                                                                ,s_city
    and dt.d year=1999
 group by dt.d year
                                                                ,s_county
       ,item.i brand
                                                                ,s_state
        ,item.i_brand_id
                                                                ,s zip
 order by dt.d year
                                                         limit 100;
        ,ext_price desc
        ,brand id
                                                        -- end query 60 in stream 0 using template
 limit 100 ;
                                                       query50.tpl
                                                        -- start query 61 in stream 0 using
-- end query 59 in stream 0 using template
                                                       template query42.tpl and seed 709936855
query52.tpl
                                                       select dt.d year
-- start query 60 in stream 0 using
                                                               ,item.i_category_id
,item.i_category
template query50.tpl and seed 1718577076
                                                               , sum (ss ext sales price)
select
   s store name
                                                        sum ss ext sales price
  ,s_company id
                                                        from date dim dt
  ,s_street_number
                                                               ,store_sales
  ,s_street_name
                                                                ,item
                                                        where dt.d date sk =
  ,s street type
  ,s_suite number
                                                        store sales.ss sold date sk
  ,s_city
                                                               and store_sales.ss_item_sk =
  ,s_county
                                                        item.i_item_sk
  ,s_state
                                                               and item.i manager id = 1
  ,s zip
                                                               and dt.d moy=12
                                                               and dt.d_year=2000
  ,sum(case when (sr_returned_date_sk -
ss sold date sk \leq 30 ) then \overline{1} else 0 end)
                                                        group by
                                                                       dt.d year
as 30_days
                                                                        ,item.i_category_id
                                                                        ,item.i_category
  --sum(ss_ext_sales_price)
 , sum (case when (sr returned date sk -
ss_sold_date_sk > 30 and
                                                        order by
                   (sr_returned_date_sk -
                                                       desc,dt.d_year
ss sold date sk <= 60) then 1 else 0 end )
                                                                         --4 desc, dt.d year
as 31 60 days
                                                                         sum ss ext sales price
 , sum(case when (sr returned date sk -
                                                       desc, dt.d year
ss_sold_date_sk > 60) and
                                                                        ,item.i_category_id
                   (sr_returned_date_sk -
                                                                        ,item.i_category
ss sold date sk \leq 90) then 1 else 0 end)
                                                         limit 100;
as 61 90 days
,sum(case when (sr_returned_date_sk -
ss_sold_date_sk > 90) and
                                                        -- end query 61 in stream 0 using template
                                                       query42.tpl
                   (sr returned date sk -
                                                        -- start query 62 in stream 0 using
ss sold date sk \leq 120) then \overline{1} else 0 end)
                                                       template query41.tpl and seed 944250029
as 91 120 days
                                                       select distinct(i product name)
  ,sum(case when (sr_returned_date_sk -
                                                        from item i1
ss_sold_date_sk > 120) then 1 else 0 end)
                                                        where i_manufact_id between 716 and 716+40
as above120_days
                                                           and exists (
                                                                select tmp.i manufact from
from
   store_sales
  ,store_returns
                                                                select i manufact
                                                                from item
  ,store
  ,date dim d1
                                                                --(i_manufact = i1.i_manufact and ((i_category = 'Women' and
  ,date_dim d2
where
    d2.d year = 1999
```

```
,i_category
        (i color = 'spring' or i color =
'hot') and
                                                           ,i_class
        (i units = 'Carton' or i units =
                                                           ,i current price
'Tbl') and
                                                           ,itemrevenue
        (i_size = 'large' or i size =
                                                            ,revenueratio
'N/A')
                                                     from (
        ) or
                                                     select i_item_id
                                                           ,i_item desc
        (i category = 'Women' and
                                                           ,i_category
        (i color = 'magenta' or i color =
'goldenrod') and
                                                            ,i_class
        (i_units = 'Cup' or i_units = 'Oz')
                                                            ,i_current_price
                                                            , sum(ws_ext_sales_price) as
        (i size = 'economy' or i size =
                                                     itemrevenue
'extra large')
                                                     ,sum(ws ext sales price) *100/sum(sum(ws ext
        ) or
                                                     _sales_price)) over
        (i_category = 'Men' and
        (i color = 'cyan' or i color =
                                                                (partition by i class) as
'antique') and
    (i_units = 'Dozen' or i_units =
                                                     revenueratio
                                                     from
'Case') and
                                                             web sales
        (i_size = 'medium' or i size =
                                                             ,item
'petite')
                                                             ,date dim
        ) or
                                                     where
        (i_category = 'Men' and
(i_color = 'moccasin' or i_color =
                                                             ws_item_sk = i_item_sk
                                                             and i category in ('Jewelry',
'black') and
                                                      'Men', 'Books')
        (i units = 'Box' or i units =
                                                             and ws sold date sk = d date sk
'Pallet') and
                                                             and d_date between cast('2002-06-
        (i_size = 'large' or i size =
                                                     11' as date)
'N/A')
                                                     date add(cast('2002-06-11' as date), 30)
        ))
        --)
                                                     group by
        or
                                                             i item_id
       --(i_manufact = i1.i_manufact and
                                                             ,i_item_desc
        ((i_category = 'Women' and (i_color = 'azure' or i_color =
                                                              ,i_category
                                                              ,i_class
'light') and
                                                              ,i_current_price
        (i units = 'Gross' or i units =
                                                     order by
'Each') and
                                                             i category
        (i_size = 'large' or i size =
                                                             ,i_class
'N/A')
                                                             ,i_item_id
                                                              ,i item desc
        (i category = 'Women' and
                                                              ,revenueratio
        (i color = 'mint' or i color =
                                                     ) v1
'burnished') and
                                                       limit 100;
        (i_units = 'N/A' or i_units =
'Unknown') and
                                                      -- end query 64 in stream 0 using template
       (i size = 'economy' or i_size =
                                                     query12.tpl
                                                      -- start query 65 in stream 0 using
'extra large')
                                                     template query20.tpl and seed 711739272
        ) or
        (i_category = 'Men' and
                                                     select v1.i_item_desc
        (i_color = 'floral' or i color =
                                                           ,v1.i_category
,v1.i_class
'midnight') and
        (i_units = 'Pound' or i_units =
                                                             ,v1.i_current_price
'Ton') and
                                                             ,v1.itemrevenue
        (i size = 'medium' or i size =
                                                             ,v1.revenueratio
'petite')
                                                     from
        ) or
        (i_category = 'Men' and
(i_color = 'navy' or i_color =
                                                      select i_item_id, i_item_desc
                                                            ,i category
                                                             ,i class
'blue') and
        (i_units = 'Bundle' or i units =
                                                             ,i current price
'Ounce') and
                                                             , \verb"sum" (cs_ext_sales_price) as
        (i_size = 'large' or i_size =
                                                     itemrevenue
'N/A')
                                                     ,sum(cs_ext_sales price)*100/sum(sum(cs ext
                                                     _sales_price)) over
        --)
        ) tmp where tmp.i_manufact =
                                                               (partition by i_class) as
il.i manufact )
                                                     revenueratio
                                                      from catalog_sales
order by i_product_name
  limit 10\overline{0};
                                                          ,item
                                                          ,date dim
-- end query 62 in stream 0 using template
                                                      where cs \overline{i}tem sk = i item sk
                                                        and i_category in ('Jewelry', 'Music',
query41.tpl
-- start query 64 in stream 0 using
                                                      'Men')
template query12.tpl and seed 918962166
                                                        and cs_sold_date_sk = d_date_sk
select i_item_desc
```

```
and d date between cast('2000-02-09' as
                                                     where ss sold time sk = time dim.t time sk
date)
                                                          and ss hdemo sk =
                  and date add(cast('2000-
                                                    household demographics.hd demo sk
02-09' as date), 30)
                                                          and ss store sk = s store sk
                                                          and time_dim.t_hour = 9
 group by i_item_id
         ,i_item_desc
                                                          and time_dim.t_minute >= 30
                                                          and
         ,i_category
         ,i class
                                                     ((household demographics.hd dep count = 1
         ,i current price
order by i_category
                                                    household demographics.hd vehicle count<=1+
        ,i_class
         ,i_item_id
         ,i item desc
                                                     (household demographics.hd dep count = 4
         ,revenueratio
                                                    household_demographics.hd_vehicle_count<=4+
) v1
  limit 100
                                                     (household demographics.hd dep count = 2
-- end query 65 in stream 0 using template
                                                    and
query20.tpl
                                                    household demographics.hd vehicle count <= 2+
 - start query 66 in stream 0 using
template query88.tpl and seed 1924183468
                                                          and store.s store name = 'ese') s3,
                                                      (select count(*) h10_to_10_30
select *
                                                     from store sales, household demographics ,
 (select count(*) h8 30 to 9
                                                     time_dim, store
 from store sales, household demographics,
                                                     where ss sold time sk = time dim.t time sk
                                                         and \overline{ss} hdemo s\overline{k} =
time dim, store
where ss_sold_time_sk = time_dim.t_time_sk
                                                    household_demographics.hd_demo_sk
     and ss hdemo sk =
                                                          and ss_store_sk = s_store_sk
                                                          and time_dim_t hour = 10
household demographics.hd demo sk
     and ss_store_sk = s_store_sk
                                                          and time_dim.t_minute < 30
     and time dim.t hour = 8
                                                         and
     and time_dim.t_minute >= 30
                                                     ((household_demographics.hd_dep_count = 1
((household demographics.hd dep count = 1
                                                    household demographics.hd vehicle count<=1+
household demographics.hd vehicle count<=1+
                                                     (household demographics.hd dep count = 4
(household demographics.hd dep count = 4
                                                    household demographics.hd vehicle count<=4+
                                                     2) or
household demographics.hd vehicle count<=4+
                                                     (household demographics.hd dep count = 2
                                                     and
(household demographics.hd dep count = 2
                                                    household demographics.hd vehicle count<=2+
and
                                                          and store.s_store_name = 'ese') s4,
household demographics.hd vehicle count<=2+
                                                      (select count(*) h10 \overline{30} to 11
2))
 and store.s_store_name = 'ese') s1,
(select count(*) h9_to_9_30
                                                     from store_sales, household demographics ,
                                                     time dim, store
from store sales, household demographics,
                                                     where ss_sold_time_sk = time_dim.t_time_sk
time dim, store
                                                         and ss hdemo sk =
where ss sold time sk = time dim.t time sk
                                                    household demographics.hd demo sk
     and ss_hdemo_sk =
                                                          and ss_store_sk = s_store_sk
household demographics.hd demo sk
                                                          and time dim.t hour = 10
     and ss store sk = s store sk
                                                          and time dim.t minute >= 30
     and time dim.t hour = 9
     and time_dim.t_minute < 30
                                                     ((household_demographics.hd_dep_count = 1
     and
((household demographics.hd dep count = 1
                                                     household demographics.hd vehicle count<=1+
                                                     2) or
household demographics.hd vehicle count<=1+
                                                     (household_demographics.hd_dep_count = 4
(household demographics.hd dep count = 4
                                                     household demographics.hd vehicle count<=4+
                                                    2) or
household demographics.hd_vehicle_count<=4+
                                                     (household_demographics.hd_dep_count = 2
2) or
(household demographics.hd dep count = 2
                                                    household demographics.hd vehicle count<=2+
and
                                                     2))
                                                          and store.s_store_name = 'ese') s5,
household demographics.hd vehicle count<=2+
                                                      (select count(*) h11 to 11 30
     and store.s store name = 'ese') s2,
                                                     from store sales, household demographics,
 (select count(*) h9_30_to_10
                                                     time_dim, store
 from store_sales, household_demographics ,
                                                     where ss_sold_time_sk = time_dim.t_time_sk
time dim, store
```

```
and ss hdemo sk =
                                                    select i brand id brand id, i brand
household demographics.hd demo sk
                                                    brand, t_hour, t_minute,
     and ss store sk = s store sk
                                                            sum(ext price) ext price
     and time dim.t hour = 11
                                                     from item, (select ws ext sales price as
     and time_dim.t_minute < 30
                                                    ext price,
     and
                                                                              ws sold date sk as
((household demographics.hd dep count = 1
                                                     sold date sk,
                                                                              ws item sk as
household demographics.hd vehicle count<=1+
                                                    sold item sk,
                                                                             ws sold time sk as
                                                     time_sk
(household demographics.hd dep count = 4
                                                                      from web sales, date dim
                                                                      where d date sk = \frac{1}{2}
household demographics.hd vehicle count<=4+
                                                    ws_sold_date sk
                                                                        and d_{moy}=12
2) or
                                                                        and d year=1998
(household demographics.hd dep count = 2
                                                                      union all
                                                                      select cs ext sales price
household demographics.hd vehicle count<=2+
                                                    as ext price,
                                                                             cs sold date sk as
     and store.s_store_name = 'ese') s6,
                                                     sold date sk,
 (select count(*) h11 \overline{30} to 12
                                                                             cs item sk as
from store sales, household demographics,
                                                     sold item sk,
time dim, store
                                                                              cs sold time sk as
where ss sold time sk = time dim.t time sk
                                                     time sk
     and ss hdemo sk =
                                                                      from
\verb|household_demographics.hd| demo sk|
                                                    catalog sales, date dim
     and ss_store_sk = s_store_sk
                                                                      \overline{where} d date sk =
     and time_dim.t_hour = 11
                                                     cs sold date sk
     and time dim.t minute >= 30
                                                                        and d moy=12
                                                                        and d year=1998
((household demographics.hd dep count = 1
                                                                      union all
                                                                      select ss_ext_sales_price
household_demographics.hd_vehicle_count<=1+
                                                     as ext price,
2) or
                                                                             ss sold date sk as
                                                     sold date sk,
(household demographics.hd dep count = 4
                                                                              ss item sk as
                                                     sold item sk,
household demographics.hd vehicle count<=4+
                                                                             ss sold time sk as
2) or
                                                     {\tt time\_sk}
                                                                      from store_sales,date_dim
(household demographics.hd dep count = 2
                                                                      where d date sk =
                                                     ss sold date sk
household demographics.hd vehicle count<=2+
                                                                        and d moy=12
                                                                        and d_year=1998
2))
     and store.s_store_name = 'ese') s7,
                                                                      ) as tmp,time_dim
 (select count(*) h12_to_12_30
                                                     where
 from store sales, household demographics,
                                                       sold item sk = i item sk
time dim, store
                                                        and i_manager_id=1
where ss sold time sk = time dim.t time sk
                                                       and time sk = t time sk
                                                       and (t_meal_time = 'breakfast' or
     and ss hdemo sk =
                                                     t_meal_time = 'dinner')
household demographics.hd demo sk
     and ss store_sk = s_store_sk
                                                     group by i brand,
     and time_dim.t_hour = 12
                                                     i_brand_id,t_hour,t_minute
     and time dim.t minute < 30
                                                     order by ext price desc, i brand id
((household demographics.hd dep count = 1
                                                     -- end query 72 in stream 0 using template
household demographics.hd vehicle count<=1+
                                                     query71.tpl
                                                     -- start query 73 in stream 0 using
                                                     template query34.tpl and seed 1451328249
(household demographics.hd dep count = 4
                                                     select c_last_name
                                                            ,c_first_name
household_demographics.hd_vehicle_count<=4+
                                                            ,c salutation
                                                            ,c preferred cust flag
                                                            ,ss_ticket_number
(household demographics.hd dep count = 2
                                                            ,cnt from
                                                        (select ss_ticket_number
household demographics.hd vehicle count<=2+
                                                               ,ss_customer sk
                                                               ,count(*) cnt
     and store.s_store_name = 'ese') s8
                                                        from
                                                     store sales, date dim, store, household demogr
                                                     aphics
-- end query 66 in stream 0 using template
                                                        where store sales.ss sold date sk =
                                                    date_dim.d_date_sk
query88.tpl
-- start query 72 in stream 0 using
                                                        and store_sales.ss_store_sk =
template query71.tpl and seed 1436004490
                                                    store.s store sk
```

```
and store sales.ss hdemo sk =
                                                         date dim
household demographics.hd demo sk
                                                     where cs_sold_date_sk = d_date_sk
    and (date dim.d dom between 1 and 3 or
                                                           and d date between cast('2002-08-24'
date dim.d dom between 25 and 28)
   and
                                                                      and date add(cast('2002-
                                                    08-24' as date), 30)
(household demographics.hd buy potential =
'1001-5000' or
                                                     group by cs call center sk
household demographics.hd buy potential =
                                                     cr as
'5001-100<del>0</del>0')
                                                     (select
                                                            sum(cr_return_amount) as returns,
   and
household demographics.hd vehicle count > 0
                                                            sum(cr_net_loss) as profit_loss
   and (case when
                                                     from catalog returns,
household demographics.hd vehicle count > 0
                                                        date dim
                                                     where cr returned date sk = d date sk
       then
household_demographics.hd_dep_count/
                                                           and d_date between cast('2002-08-24'
household demographics.hd vehicle count
       else null
                                                                       and date add(cast('2002-
       end) > 1.2
                                                    08-24' as date), 30)
    and date dim.d year in
(1999,1999+1,1999+2)
                                                     ws as
    and (
                                                     ( select wp web page sk,
                                                            sum(ws_ext_sales_price) as sales,
    store.s_county = 'Sierra County'
    or store.s_county = 'Lunenburg County'
                                                            sum(ws_net_profit) as profit
    or store.s county = 'Jackson County'
                                                     from web sales,
    or store.s_county = 'Harmon County'
or store.s_county = 'Mesa County'
                                                          date dim,
                                                          web page
    or store.s_county = 'Pipestone County'
                                                     where ws_sold_date_sk = d_date_sk
    or store.s_county = 'Pennington County'
                                                           and d_date between cast('2002-08-24'
    or store.s county = 'Perry County')
                                                    as date)
                                                                      and date add(cast('2002-
    aroup by
ss ticket number,ss_customer_sk)
                                                    08-24' as date), 30)
dn,customer
                                                           and ws_web_page_sk = wp_web_page_sk
    where ss_customer_sk = c_customer_sk
                                                     group by wp_web_page_sk),
     and cnt between 15 and 20
                                                     wr as
    order by
                                                     (select wp_web_page_sk,
                                                            sum (wr return amt) as returns,
c last name, c first name, c salutation, c pre
ferred cust flag desc;
                                                            sum(wr net loss) as profit loss
                                                     from web returns,
-- end query 73 in stream 0 using template
                                                          date dim,
                                                          web_page
query34.tpl
-- start query 78 in stream 0 using
                                                     where wr_returned_date_sk = d_date_sk
template query77.tpl and seed 1879081522
                                                           and d date between cast('2002-08-24'
with ss as
                                                    as date)
 (select s_store_sk,
                                                                      and date_add(cast('2002-
                                                    08-24' as date), 30)
         sum(ss_ext_sales_price) as sales,
         sum(ss net profit) as profit
                                                           and wr web page sk = wp web page sk
                                                     group by wp_web_page_sk)
from store sales,
     date dim,
                                                      select channel
     store
                                                            , id
where ss_sold_date_sk = d_date_sk
                                                            , sum(sales) as sales
      and d date between cast('2002-08-24'
                                                            , sum(returns) as returns
                                                             , sum(profit) as profit
                  and date_add(cast('2002-
                                                     from
08-24' as date), 30)
                                                     (select 'store channel' as channel
       and ss store sk = s store sk
                                                            , ss.s store sk as id
group by s_store_sk)
                                                            , sales
                                                            , coalesce(returns, 0) as returns
 sr as
                                                             , (profit -
                                                    coalesce(profit_loss,0)) as profit
from ss left join sr
 (select s store sk,
         sum (sr return amt) as returns,
         sum(sr net loss) as profit loss
                                                           on ss.s_store_sk = sr.s_store_sk
 from store returns,
                                                     union all
      date dim,
                                                     select 'catalog channel' as channel
                                                           , cs_call_center_sk as id
                                                            , sales
where sr returned date sk = d date sk
      and d date between cast('2002-08-24'
                                                            , returns
                                                             , (profit - profit_loss) as profit
                  and date_add(cast('2002-
                                                     from cs
08-24' as date), 30)
                                                           , cr
       and sr_store_sk = s_store_sk
                                                     union all
                                                     select 'web channel' as channel
group by s_store_sk),
 cs as
                                                            , ws.wp web page sk as id
 (select cs call center sk,
                                                            , sales
                                                            , coalesce(returns, 0) returns
        sum(cs_ext_sales_price) as sales,
        sum(cs_net_profit) as profit
                                                             , (profit -
 from catalog_sales,
                                                    coalesce(profit loss,0)) as profit
```

```
from ws left join wr
                                                      and ib income band sk =
       on ws.wp_web_page_sk =
                                                   hd_income_band_sk
wr.wp web page sk
                                                       and cd demo sk = c current cdemo sk
                                                       and hd demo sk = c current hdemo sk
) x
                                                     and sr_cdemo_sk = cd_demo_sk
order by customer_id --
 group by channel, id WITH ROLLUP
order by channel
        ,id
                                                       limit 100;
  limit 100;
                                                    -- end query 80 in stream 0 using template
-- end query 78 in stream 0 using template
                                                    query84.tpl
                                                    -- start query 82 in stream 0 using
query77.tpl
-- start query 79 in stream 0 using
                                                    template query55.tpl and seed 1117454508
template query73.tpl and seed 413577677
                                                    select i brand id brand id, i brand brand,
select c last name
                                                           sum(ss ext sales price) ext price
       ,c_first_name
                                                     from date_dim, store_sales, item
       ,c salutation
                                                     where d_date_sk = ss_sold_date_sk
       ,c preferred cust flag
                                                           and ss item sk = i item sk
       ,ss_ticket_number
                                                            and i_manager_id=48
                                                           and d moy=11
                                                           and d_year=2001
   (select ss_ticket_number
        ,ss_customer_sk
                                                     group by i_brand, i_brand_id
          ,count(*) cnt
                                                     order by ext_price desc, i_brand_id
                                                     limit 100 ;
   from
store_sales,date_dim,store,household_demogr
                                                    -- end query 82 in stream 0 using template
   where store sales.ss sold date sk =
                                                    query55.tpl
date dim.d date sk
                                                    -- start query 83 in stream 0 using
    and store_sales.ss_store_sk =
                                                    template query56.tpl and seed 1152645577
                                                    with ss as (
store.s store sk
   and store sales.ss hdemo sk =
                                                    select i item id, sum(ss ext sales price)
household demographics.hd demo sk
                                                    total sales
   and date dim.d dom between 1 and 2
                                                    from
(household_demographics.hd_buy_potential =
                                                     select distinct
                                                    i1.i_item_id,ss_ext_sales_price
'501-1000' or
household demographics.hd buy potential =
                                                            store sales,
'5001-10000')
                                                           date dim,
                                                             customer address,
household demographics.hd vehicle count > 0
                                                             item i1,
   and case when
                                                             item i2
household demographics.hd vehicle count > 0
                                                     where i1.i_item_id = i2.i_item_id
                                                     and i2.i color in
                                                    ('maroon','powder','lawn')
household demographics.hd dep count/
                                                     and ss item_sk
household_demographics.hd_vehicle_count
                                                    i1.i_item_sk
else null end > 1
                                                     and
                                                             ss sold date sk
                                                    d_date_sk
   and date dim.d year in
(1999, 1999+1, 1999+2)
                                                     and d_year
                                                                                     = 2000
    and store.s_county in ('Lea
                                                                                     = 1
                                                    and
                                                             d moy
County','West Feliciana Parish','Nowata
                                                    and
                                                            ss addr sk
County','Jackson County')
                                                    ca address sk
                                                            ca_gmt_offset
                                                                                      = -5
                                                    and
  group by
ss ticket_number,ss_customer_sk)
                                                    ) v1
dj,customer
                                                     group by i item id),
    where ss customer sk = c customer sk
                                                    cs as (
     and cnt between 1 and \overline{5}
                                                    select i item id, sum(cs ext sales price)
    order by cnt desc;
                                                    total_sales
                                                     from
-- end query 79 in stream 0 using template
                                                     select distinct
-- start query 80 in stream 0 using
                                                   il.i item id, cs ext sales price
template query84.tpl and seed 1842474049
                                                    from
select c_customer_id as customer_id
                                                            catalog_sales,
       ,concat(c_last_name, ', ' ,
                                                           date dim,
c first name) as customername
                                                             customer address,
 from customer
                                                             item il.
    ,customer_address
                                                             item i2
     ,customer_demographics
                                                     where i1.i_item_id = i2.i_item_id
                                                    and i2.i_color in ('maroon','powder','lawn')
     ,household demographics
    ,income_band
     ,store returns
                                                     and
                                                           cs item sk
 where ca city
                      = 'Mount Zion'
                                                    il.i item sk
   and c current addr_sk = ca_address_sk
                                                    and cs sold date sk
   and ib_lower_bound >= 50749
and ib_upper_bound <= 50749 + 50000
                                                   d_date_sk
                                                                                      = 2000
                                                    and
                                                            d_year
                                                     and
                                                                                      = 1
                                                             d moy
```

```
cs_bill_addr_sk
and
                                                            sum(case when
ca address_sk
                                                    (d day name='Saturday') then sales price
and
       ca gmt offset
                                 = -5
                                                     else null end) sat sales
) v2
                                                     from wscs
group by i_item_id),
                                                         ,date dim
                                                      where d_{date_sk} = sold_{date_sk}
ws as (
select i item id, sum(ws ext sales price)
                                                      group by d week seq)
total sales
                                                      select d week seq1
                                                            round(sun sales1/sun sales2,2)
                                                            , round(mon_sales1/mon_sales2,2)
                                                            ,round(tue_sales1/tue_sales2,2)
select distinct
i1.i_item_id,ws_ext_sales_price
                                                            , round(wed_sales1/wed_sales2,2)
                                                            , round(thu sales1/thu sales2,2)
                                                            round(fri sales1/fri sales2,2)
       web sales,
       date_dim,
                                                            ,round(sat sales1/sat sales2,2)
        customer address,
                                                      {\tt from}
                                                      (select wswscs.d week seq d week seq1
         item i1,
                                                             ,sun_sales sun_sales1
         item i2
where i1.i item_id = i2.i_item_id
                                                             ,mon sales mon sales1
and i2.i_color in
                                                             ,tue_sales tue_sales1
('maroon','powder','lawn')
                                                             ,wed sales wed sales1
                                                             ,thu_sales thu sales1
      ws item sk
                                                             fri_sales fri_sales1
i1.i_item_sk
and ws_sold_date_sk
                                                             ,sat sales sat sales1
d date sk
                                                       from wswscs, date dim
and
        d year
                                  = 2000
                                                       where date dim.d week seq =
                                                     wswscs.d_week_seq and
        d moy
                                  = 1
and
        ws_bill_addr_sk
                                                            d_{year} = 1998) y,
and
ca address sk
                                                      (select wswscs.d_week_seq d_week_seq2
                                  = -5
                                                            sun sales sun sales2
and
       ca gmt offset
                                                             , mon_sales mon_sales2
, tue_sales tue_sales2
) v3
group by i_item id)
 select i_item_id ,sum(total_sales)
                                                             ,wed_sales wed_sales2
total_sales
                                                             ,thu sales thu sales2
from (select * from ss
                                                             ,fri_sales fri_sales2
        union all
                                                             ,sat_sales sat_sales2
        select * from cs
                                                       from wswscs
        union all
                                                          ,date dim
        select * from ws) tmp1
                                                       where date dim.d week seq =
group by i item id
                                                     {\tt wswscs.d\_week\_seq\ and}
                                                            d_{year} = 1998+1) z
order by total_sales
  limit 100;
                                                      where d_week_seq1=d_week_seq2-53
                                                     order by d week seq1;
-- end query 83 in stream 0 using template
                                                     -- end query 84 in stream 0 using template
querv56.tpl
-- start query 84 in stream 0 using
                                                     query2.tpl
template query2.tpl and seed 1528114170
                                                     -- start query 85 in stream 0 using
                                                     template query26.tpl and seed 1427200905
with wscs as
 (select sold date sk
                                                     select i_item_id,
                                                             avg(cs quantity) agg1,
        , sales price
  from (select ws_sold_date_sk sold_date_sk
                                                             avg(cs_list_price) agg2,
             ,ws_ext_sales_price
                                                             avg(cs coupon amt) agg3,
sales price
                                                             avg(cs_sales_price) agg4
                                                     from catalog_sales, customer_demographics,
        from web sales
        union all
                                                     date_dim, item, promotion
        select cs sold date sk sold date sk
                                                      where cs sold date sk = d date sk and
                                                            cs item sk = i item sk and
              ,cs ext sales price
                                                            cs_bill_cdemo_sk = cd_demo_sk and
sales_price
                                                            cs_promo_sk = p_promo_sk and
cd gender = 'M' and
       from catalog sales) x ),
wswscs as
                                                            cd marital status = 'D' and
 (select d week seq,
        sum(case when (d day name='Sunday')
                                                           cd education status = 'Advanced
then sales_price else null end) sun sales,
                                                     Degree' and
                                                    (p_channel_email = 'N' or
p_channel_event = 'N') and
        sum(case when (d_day_name='Monday')
then sales price else null end) mon sales,
                                                          d_year = 2000
       sum(case when
                                                     group by i_item_id
order by i_item_id
(d_day_name='Tuesday') then sales_price
else null end) tue_sales,
        sum(case when
                                                      limit 100;
(d day name='Wednesday') then sales price
else null end) wed sales,
                                                     -- end query 85 in stream 0 using template
        sum(case when
                                                     query26.tpl
(d day name='Thursday') then sales price
                                                     -- start query 86 in stream 0 using
else null end) thu sales,
                                                     template query40.tpl and seed 600490395
        sum(case when (d day name='Friday')
                                                     select
then sales_price else null end) fri_sales,
                                                       w_state
                                                       ,i_item_id
```

```
,sum(case when (cast(d date as date) <
                                                    -- end query 88 in stream 0 using template
cast ('2000-04-27' as date))
                                                    query53.tpl
               then cs sales price -
                                                     -- start query 89 in stream 0 using
coalesce(cr refunded cash, 0) else 0 end) as
                                                     template query79.tpl and seed 2112737383
sales before
                                                     select
 , sum(case when (cast(d date as date) >=
cast ('2000-04-27' as date))
                                                     c_last_name, c_first_name, substr(s_city, 1, 30
               then cs sales price -
                                                     ) s city part ,ss ticket number,amt,profit
coalesce(cr refunded cash, 0) else 0 end) as
sales after
                                                        (select ss_ticket_number
from
                                                               ,ss_customer_sk
   catalog sales left outer join
                                                               ,store.s city
catalog returns on
                                                               , sum (ss coupon amt) amt
       cs order number = cr order number
                                                               ,sum(ss net profit) profit
        and cs item sk = cr item sk)
                                                         from
  ,warehouse
                                                     store sales, date dim, store, household demogr
  ,item
                                                     aphics
                                                     where store_sales.ss_sold_date_sk =
date_dim.d_date_sk
  ,date dim
 where
    i_current_price between 0.99 and 1.49
                                                        and store_sales.ss_store_sk =
 and i item sk
                      = cs_item_sk
                                                     store.s_store_sk
 and cs_warehouse sk
                        = w warehouse_sk
                                                        and store sales.ss hdemo sk =
                      = d date_sk
 and cs_sold_date_sk
                                                     household_demographics.hd_demo_sk
 and d_date between date_sub(cast ('2000-
04-27' as date), 30)
                                                     (household demographics.hd dep count = 3 or
               and date add(cast ('2000-
                                                     household demographics.hd vehicle count >
04-27' as date), 30)
                                                         and date\_dim.d\_dow = 1
group by
    w_state,i_item_id
                                                         and date_dim.d_year in
                                                     (2000, 2000+1, 2000+2)
 order by w state, i item id
  limit 100;
                                                        and store.s number employees between
                                                     200 and 295
-- end query 86 in stream 0 using template
                                                        group by
                                                     ss_ticket_number,ss_customer_sk,ss_addr_sk,
-- start query 88 in stream 0 using
                                                     store.s city) ms, customer
template query53.tpl and seed 1796782974
                                                        where ss_customer_sk = c_customer_sk
select * from
                                                      order by
(select i manufact id,
                                                     c last name, c first name, s city part,
sum(ss sales price) sum sales,
                                                    profit
avg(sum(ss_sales_price)) over (partition by
                                                      limit 100;
i_manufact_id) avg_quarterly_sales
from item, store_sales, date_dim, store
                                                     -- end query 89 in stream 0 using template
where ss item sk = i item sk and
                                                     query79.tpl
ss sold \overline{d}ate \overline{s}k = d \overline{d}ate \overline{s}k and
                                                     -- start query 96 in stream 0 using
ss store sk = s_store_sk and
                                                     template query83.tpl and seed 593789178
d_month_seq in
                                                     with sr_items as
(1198, 1198+1, 1198+2, 1198+3, 1198+4, 1198+5, 11
                                                     (select i item id item id,
98+6,1198+7,1198+8,1198+9,1198+10,1198+11)
                                                            sum(sr return quantity) sr item qty
                                                      {\tt from \ store\_returns}
and
                                                           \overline{\text{JOIN item ON sr\_item\_sk}} = i\_item\_sk
((i category in
('Books','Children','Electronics') and
                                                           JOIN date_dim dd0 ON
i class in
                                                     sr returned date sk = dd0.d date sk
('personal', 'portable', 'reference', 'self-
                                                          JOIN
help') and
                                                            (select dd1.d_date
i brand in ('scholaramalgamalg
                                                            from date dim dd1
#14','scholaramalgamalg #7',
                                                                 JOIN date dim dd2 ON
               'exportiunivamalg
                                                     dd1.d week seq = dd2.d week seq
                                                           where dd2.d date in ('1999-06-
#9','scholaramalgamalg #9'))
or(i_category in ('Women','Music','Men')
                                                     14','1999-08-26','1999-11-06')) v1 ON
                                                     dd0.d date = v1.d date
and
                                                     group by i item id),
i class in
( accessories', 'classical', 'fragrances', 'pa
                                                     cr items as
                                                      (select i_item_id item_id,
nts') and
i brand in ('amalgimporto #1','edu
                                                             sum(cr_return_quantity) cr_item_qty
from catalog returns
                                                           JOIN item ON cr item sk = i_item_sk
                                                           JOIN date_dim dd0 ON
group by i_manufact_id, d_qoy ) tmp1
where case when avg_quarterly_sales > 0
                                                     cr_returned_date_sk = dd0.d_date_sk
       then abs (sum_sales -
avg_quarterly_sales) / avg_quarterly_sales
        else null end > 0.1
                                                            (select dd1.d_date
                                                            from date dim dd1
                                                                  JOIN date_dim dd2 ON
order by avg_quarterly_sales,
        sum sales,
                                                     dd1.d week seq = dd2.d week seq
         i manufact id
                                                            where dd2.d date in ('1999-06-
limit 100;
                                                     14','1999-08-26','1999-11-06')) v1 ON
                                                     dd0.d date = v1.d_date
                                                     group by i_item_id),
```

```
and s_gmt offset = -7
wr items as
 (select i item id item id,
                                                         and d_year = 1999
        sum(wr return quantity) wr item qty
                                                         and
                                                               d moy = 12) promotional sales,
 from web returns
                                                        (select sum(ss ext sales price) total
      JOIN item ON wr_item_sk = i_item_sk
                                                         from store sales
                                                              ,store
      JOIN date_dim dd0 ON
                      = dd0.d_date_sk
                                                               ,date dim
wr_returned_date_sk
                                                               ,customer
      JOIN
        (select dd1.d date
                                                               ,customer address
        from date_dim dd1 JOIN date_dim dd2 ON
                                                               ,item
                                                         where ss_sold_date_sk = d_date_sk
dd1.d week_seq = dd2.d_week_seq
                                                         and ss_store_sk = s_store_sk
        where dd2.d date in ('1999-06-
                                                                ss customer sk= c customer sk
                                                                ca_address_sk = c_current addr sk
14','1999-08-26','1999-11-06')) v1 ON
                                                         and
            = v1.d_date
dd0.d date
                                                                ss_item_sk = i_item_sk
                                                         and
 group by i_item_id)
                                                         and
                                                                ca gmt offset = -7
  select sr items.item id
                                                               i category = 'Electronics'
                                                         and
                                                               s_gmt_offset = -7
d_year = 1999
       ,sr item qty
                                                         and
                                                         and
,sr_item_qty/(sr_item_qty+cr_item_qty+wr_item_qty)/3.0 \times 100 sr_dev
                                                               d moy = 12) all sales
                                                         and
                                                      order by promotions, total
                                                       limit 100;
       ,cr item qty
,cr_item_qty/(sr_item_qty+cr_item_qty+wr_item_qty)/3.0 * 100 cr_dev
                                                      -- end query 97 in stream 0 using template
                                                      query61.tpl
       ,wr item qty
                                                        - start query 99 in stream 0 using
                                                      template query76.tpl and seed 945056756
,wr_item_qty/(sr_item_qty+cr_item_qty+wr_it
                                                      select channel, col_name, d_year, d_qoy,
em qty)/3.0 * 100 wr dev
                                                      i_category, COUNT(*) sales_cnt,
                                                      SUM(ext_sales_price) sales_amt FROM (
,(sr item qty+cr item qty+wr item qty)/3.0
                                                              SELECT 'store' as channel,
                                                      'ss hdemo sk' col_name, d_year, d_qoy,
average
 from sr_items
                                                      i_category, ss_ext_sales_price
     ,cr_items
                                                      ext sales price
     ,wr items
                                                                FROM store sales, item, date dim
                                                                WHERE ss_hdemo_sk IS NULL
AND ss_sold_date_sk=d_date_sk
where sr_items.item_id=cr_items.item_id and sr_items.item_id=wr_items.item_id
 order by sr items.item id
                                                                  AND ss item sk=i item sk
         ,sr item qty
                                                               UNION ALL
   limit 100;
                                                               SELECT 'web' as channel,
                                                      'ws_web_page_sk' col_name, d_year, d_qoy,
-- end query 96 in stream 0 using template
                                                      i category, ws ext sales price
                                                      ext_sales_price
 - start query 97 in stream 0 using
                                                                FROM web sales, item, date dim
template query61.tpl and seed 1770420976
                                                                WHERE ws_web_page_sk IS NULL
select promotions, total, cast (promotions as
                                                                  AND ws_sold_date_sk=d_date_sk
decimal(15,4))/cast(total as
                                                                  AND ws item sk=i item sk
decimal(15,4))*100
                                                               UNION ALL
                                                               SELECT 'catalog' as channel,
                                                      'cs ship addr sk' col name, d year, d qoy,
  (select sum(ss ext sales price)
promotions
                                                      i_category, cs_ext_sales_price
   from store sales
                                                      ext sales price
                                                                FROM catalog sales, item, date dim
        ,store
                                                                WHERE cs_ship_addr_sk IS NULL
AND cs_sold_date_sk=d_date_sk
        ,promotion
        ,date dim
        ,customer
                                                                  AND cs item sk=i item sk) foo
        ,customer address
                                                      GROUP BY channel, col name, d year, d qoy,
        ,item
                                                      i category
   where ss_sold_date_sk = d_date_sk
                                                      ORDER BY channel, col name, d year, d qoy,
        ss store sk = s store sk
                                                      i category
         sspromo_sk = p_promo_sk
                                                       limit 100;
         ss_customer_sk= c_customer_sk
   and
         ca_address_sk = c_current_addr_sk
                                                      -- end query 99 in stream 0 using template
   and
         ss_{item_sk} = i_{item_sk}
                                                      query76.tpl
   and
         ca gmt offset = -7
   and
         i category = 'Electronics'
   and
         (p channel dmail = 'Y' or
   and
p_channel_email = 'Y' or p_channel_tv =
'<u>Y</u>')
```

Appendix F: Load & Analayze Scripts:

This appendix contains all scripts used during the load phase of the benchmark.

F.1 Big SQL Load & Analyze scripts:

Load:

```
set schema $schema;
SET HADOOP PROPERTY 'dfs.blocksize'= 536870912;
SET HADOOP PROPERTY 'parquet.block.size' = 536870912;
load hadoop using file url '/HADOOPDS10000G_PARQ/call_center' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
call center overwrite WITH LOAD PROPERTIES ('num.map.tasks'='1');
load hadoop using file url '/HADOOPDS10000G PARQ/catalog page' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
catalog page overwrite WITH LOAD PROPERTIES ('num.map.tasks'='1');
load hadoop using file url '/HADOOPDS10000G PARQ/catalog returns' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
catalog returns overwrite WITH LOAD PROPERTIES ('num.map.tasks'='425');
load hadoop using file url '/HADOOPDS10000G PARQ/catalog sales' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
catalog sales overwrite WITH LOAD PROPERTIES ('num.map.tasks'='4250');
load hadoop using file url '/HADOOPDS10000G_PARQ/customer_demographics' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer demographics overwrite WITH LOAD PROPERTIES ('num.map.tasks'='1');
load hadoop using file url '/HADOOPDS10000G PARQ/date dim' with source properties
('field.delimiter'='|', 'ignore.extra.fields'='true') into table date dim overwrite
WITH LOAD PROPERTIES ('num.map.tasks'='1');
load hadoop using file url '/HADOOPDS10000G PARQ/household demographics' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
household demographics overwrite WITH LOAD PROPERTIES ('num.map.tasks'='1');
load hadoop using file url '/HADOOPDS10000G_PARQ/income_band' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
income band overwrite WITH LOAD PROPERTIES ('num.map.tasks'='1');
load hadoop using file url '/HADOOPDS10000G_PARQ/item' with source properties
('field.delimiter'='|', 'ignore.extra.fields'='true') into table item overwrite
WITH LOAD PROPERTIES ('num.map.tasks'='1');
load hadoop using file url '/HADOOPDS10000G PARQ/promotion' with source properties
('field.delimiter'='|', 'ignore.extra.fields'='true') into table promotion
overwrite WITH LOAD PROPERTIES ('num.map.tasks'='1');
load hadoop using file url '/HADOOPDS10000G PARQ/reason' with source properties
('field.delimiter'='|', 'ignore.extra.fields'='true') into table reason overwrite
WITH LOAD PROPERTIES ('num.map.tasks'='1');
```

```
load hadoop using file url '/HADOOPDS10000G_PARQ/ship_mode' with source properties
('field.delimiter'='|', 'ignore.extra.fields'='true') into table ship mode
overwrite WITH LOAD PROPERTIES ('num.map.tasks'='1');
load hadoop using file url '/HADOOPDS10000G_PARQ/store' with source properties
('field.delimiter'='|', 'ignore.extra.fields'='true') into table store overwrite
WITH LOAD PROPERTIES ('num.map.tasks'='1');
load hadoop using file url '/HADOOPDS10000G PARQ/store returns' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
store returns overwrite WITH LOAD PROPERTIES ('num.map.tasks'='700');
load hadoop using file url '/HADOOPDS10000G PARQ/store sales' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
store sales overwrite WITH LOAD PROPERTIES ('num.map.tasks'='5500');
load hadoop using file url '/HADOOPDS10000G_PARQ/time_dim' with source properties
('field.delimiter'='|', 'ignore.extra.fields'='true') into table time_dim overwrite
WITH LOAD PROPERTIES ('num.map.tasks'='1');
load hadoop using file url '/HADOOPDS10000G PARQ/warehouse/' with source properties
('field.delimiter'='|', 'ignore.extra.fields'='true') into table warehouse
overwrite WITH LOAD PROPERTIES ('num.map.tasks'='1');
load hadoop using file url '/HADOOPDS10000G PARQ/web page' with source properties
('field.delimiter'='|', 'ignore.extra.fields'='true') into table web page overwrite
WITH LOAD PROPERTIES ('num.map.tasks'='1');
load hadoop using file url '/HADOOPDS10000G_PARQ/web_returns' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
web returns overwrite WITH LOAD PROPERTIES ('num.map.tasks'='200');
load hadoop using file url '/HADOOPDS10000G PARQ/web sales' with source properties
('field.delimiter'='|', 'ignore.extra.fields'='true') into table web sales
overwrite WITH LOAD PROPERTIES ('num.map.tasks'='2000');
load hadoop using file url '/HADOOPDS10000G PARQ/web site' with source properties
('field.delimiter'='|', 'ignore.extra.fields'='true') into table web_site overwrite
WITH LOAD PROPERTIES ('num.map.tasks'='1');
set schema $schema;
SET HADOOP PROPERTY 'dfs.blocksize'= 536870912;
SET HADOOP PROPERTY 'parquet.block.size' = 536870912;
load hadoop using file url
'/HADOOPDS10000G PARQ/customer address/customer address 10 16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer address overwrite;
load hadoop using file url
'/HADOOPDS10000G PARQ/customer address/customer address 11 16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer address append;
load hadoop using file url
'/HADOOPDS10000G PARQ/customer address/customer address 12 16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer_address append ;
load hadoop using file url
```

'/HADOOPDS10000G PARQ/customer address/customer address 13 16.dat' with source

```
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer address append;
load hadoop using file url
'/HADOOPDS10000G PARQ/customer address/customer address 14 16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer address append;
load hadoop using file url
'/HADOOPDS10000G PARQ/customer address/customer address 15 16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer address append;
load hadoop using file url
'/HADOOPDS10000G PARQ/customer address/customer address 16 16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer address append;
load hadoop using file url
'/HADOOPDS10000G_PARQ/customer_address/customer_address_1_16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer address append;
load hadoop using file url
'/HADOOPDS10000G PARQ/customer address/customer address 2 16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer address append;
load hadoop using file url
'/HADOOPDS10000G PARQ/customer address/customer address 3 16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer address append;
load hadoop using file url
'/HADOOPDS10000G PARQ/customer address/customer address 4 16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer address append;
load hadoop using file url
'/HADOOPDS10000G PARQ/customer address/customer address 5 16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer_address append ;
load hadoop using file url
'/HADOOPDS10000G PARQ/customer address/customer address 6 16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer address append;
load hadoop using file url
'/HADOOPDS10000G_PARQ/customer_address/customer_address_7_16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer address append;
load hadoop using file url
'/HADOOPDS10000G PARQ/customer address/customer address 8 16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer_address append ;
load hadoop using file url
'/HADOOPDS10000G_PARQ/customer_address/customer_address_9_16.dat' with source
properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer address append;
set schema $schema;
SET HADOOP PROPERTY 'dfs.blocksize'= 536870912;
SET HADOOP PROPERTY 'parquet.block.size' = 536870912;
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 10 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer overwrite;
```

```
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 11 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer append;
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 12 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer append;
load hadoop using file url '/HADOOPDS10000G_PARQ/customer/customer_13_16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 14 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer append;
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 15 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer append;
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 16 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer append;
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 1 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer append;
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 2 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer append;
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 3 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer append;
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 4 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer append;
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 5 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer append;
load hadoop using file url '/HADOOPDS10000G_PARQ/customer/customer_6_16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer append;
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 7 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer append;
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 8 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
load hadoop using file url '/HADOOPDS10000G PARQ/customer/customer 9 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
customer append;
set schema $schema;
SET HADOOP PROPERTY 'dfs.blocksize' = 536870912;
SET HADOOP PROPERTY 'parquet.block.size' = 536870912;
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 10 16.dat'
with source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into
table inventory overwrite;
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 11 16.dat'
with source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into
table inventory append;
```

```
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 12 16.dat'
with source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into
table inventory append;
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 13 16.dat'
with source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into
table inventory append;
load hadoop using file url '/HADOOPDS10000G_PARQ/inventory/inventory_14_16.dat'
with source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into
table inventory append;
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 15 16.dat'
with source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into
table inventory append;
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 16 16.dat'
with source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into
table inventory append;
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 1 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
inventory append;
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 2 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
inventory append;
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 3 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
inventory append;
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 4 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
inventory append;
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 5 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
inventory append;
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 6 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
inventory append;
load hadoop using file url '/HADOOPDS10000G_PARQ/inventory/inventory_7_16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
inventory append;
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 8 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
inventory append;
load hadoop using file url '/HADOOPDS10000G PARQ/inventory/inventory 9 16.dat' with
source properties ('field.delimiter'='|', 'ignore.extra.fields'='true') into table
inventory append;
```

Analyze:

Set schema \$schema;

ANALYZE TABLE call_center COMPUTE STATISTICS FOR COLUMNS cc_call_center_sk,
cc_call_center_id, cc_rec_start_date, cc_rec_end_date, cc_closed_date_sk,
cc_open_date_sk, cc_name, cc_class, cc_employees, cc_sq_ft, cc_hours, cc_manager,
cc_mkt_id, cc_mkt_class, cc_mkt_desc, cc_market_manager, cc_division,
cc_division_name, cc_company, cc_company_name, cc_street_number, cc_street_name,
cc_street_type, cc_suite_number, cc_city, cc_county, cc_state, cc_zip, cc_country,
cc_gmt_offset, cc_tax_percentage;

ANALYZE TABLE catalog page COMPUTE STATISTICS FOR COLUMNS cp catalog page sk,

ANALYZE TABLE catalog_returns COMPUTE STATISTICS FOR COLUMNS cr_returned_date_sk, cr returned time sk, cr item sk, cr refunded customer sk, cr refunded cdemo sk,

cp_catalog_page_id, cp_start_date_sk, cp_end_date_sk, cp_department, cp_catalog_number, cp_catalog_page_number, cp_description, cp_type; cr_refunded_hdemo_sk, cr_refunded_addr_sk, cr_returning_customer_sk,
cr_returning_cdemo_sk, cr_returning_hdemo_sk, cr_returning_addr_sk,
cr_call_center_sk, cr_catalog_page_sk, cr_ship_mode_sk, cr_warehouse_sk,
cr_reason_sk, cr_order_number, cr_return_quantity, cr_return_amount, cr_return_tax,
cr_return_amt_inc_tax, cr_fee, cr_return_ship_cost, cr_refunded_cash,
cr_reversed_charge, cr_store_credit, cr_net_loss;

ANALYZE TABLE catalog_sales COMPUTE STATISTICS FOR COLUMNS cs_sold_date_sk, cs_sold_time_sk, cs_ship_date_sk, cs_bill_customer_sk, cs_bill_cdemo_sk, cs_bill_hdemo_sk, cs_bill_addr_sk, cs_ship_customer_sk, cs_ship_cdemo_sk, cs_ship_hdemo_sk, cs_ship_addr_sk, cs_call_center_sk, cs_catalog_page_sk, cs_ship_mode_sk, cs_warehouse_sk, cs_item_sk, cs_promo_sk, cs_order_number, cs_quantity, cs_wholesale_cost, cs_list_price, cs_sales_price, cs_ext_discount_amt, cs_ext_sales_price, cs_ext_wholesale_cost, cs_ext_list_price, cs_ext_tax, cs_coupon_amt, cs_ext_ship_cost, cs_net_paid, cs_net_paid_inc_tax, cs_net_paid_inc_ship, cs_net_paid_inc_sh

ANALYZE TABLE customer COMPUTE STATISTICS FOR COLUMNS c_customer_sk, c_customer_id, c_current_cdemo_sk, c_current_hdemo_sk, c_current_addr_sk, c_first_shipto_date_sk, c_first_sales_date_sk, c_salutation, c_first_name, c_last_name, c_preferred_cust_flag, c_birth_day, c_birth_month, c_birth_year, c_birth_country, c_login, c_email_address, c_last_review_date;

ANALYZE TABLE customer_address COMPUTE STATISTICS FOR COLUMNS ca_address_sk, ca_address_id, ca_street_number, ca_street_name, ca_street_type, ca_suite_number, ca_city, ca_county, ca_state, ca_zip, ca_country, ca_gmt_offset, ca_location_type;

ANALYZE TABLE customer_demographics COMPUTE STATISTICS FOR COLUMNS cd_demo_sk, cd_gender, cd_marital_status, cd_education_status, cd_purchase_estimate, cd_credit_rating, cd_dep_count, cd_dep_employed_count, cd_dep_college_count;

ANALYZE TABLE date_dim COMPUTE STATISTICS FOR COLUMNS d_date_sk, d_date_id, d_date, d_month_seq, d_week_seq, d_quarter_seq, d_year, d_dow, d_moy, d_dom, d_qoy, d_fy_year, d_fy_quarter_seq, d_fy_week_seq, d_day_name, d_quarter_name, d_holiday, d_weekend, d_following_holiday, d_first_dom, d_last_dom, d_same_day_ly, d_same_day_lq, d_current_day, d_current_week, d_current_month, d_current_quarter, d_current_year;

ANALYZE TABLE household_demographics COMPUTE STATISTICS FOR COLUMNS hd_demo_sk, hd_income_band_sk, hd_buy_potential, hd_dep_count, hd_vehicle_count;

ANALYZE TABLE income_band COMPUTE STATISTICS FOR COLUMNS ib_income_band_sk, ib_lower_bound, ib_upper_bound;

ANALYZE TABLE inventory COMPUTE STATISTICS FOR COLUMNS inv_date_sk, inv_item_sk, inv_warehouse_sk, inv_quantity_on_hand;

ANALYZE TABLE item COMPUTE STATISTICS FOR COLUMNS i_item_sk, i_item_id, i_rec_start_date, i_rec_end_date, i_item_desc, i_current_price, i_wholesale_cost, i_brand_id, i_brand, i_class_id, i_class, i_category_id, i_category, i_manufact_id, i_manufact, i_size, i_formulation, i_color, i_units, i_container, i_manager_id, i product name;

ANALYZE TABLE promotion COMPUTE STATISTICS FOR COLUMNS p_promo_sk, p_promo_id, p_start_date_sk, p_end_date_sk, p_item_sk, p_cost, p_response_target, p_promo_name, p_channel_dmail, p_channel_email, p_channel_catalog, p_channel_tv, p_channel_radio, p_channel_press, p_channel_event, p_channel_demo, p_channel_details, p_purpose, p_discount_active;

ANALYZE TABLE reason COMPUTE STATISTICS FOR COLUMNS r_reason_sk, r_reason_id, r reason desc;

ANALYZE TABLE ship_mode COMPUTE STATISTICS FOR COLUMNS sm_ship_mode_sk, sm_ship_mode_id, sm_type, sm_code, sm_carrier, sm_contract;

ANALYZE TABLE store COMPUTE STATISTICS FOR COLUMNS s_store_sk, s_store_id, s_rec_start_date, s_rec_end_date, s_closed_date_sk, s_store_name, s number employees, s floor space, s hours, s manager, s market id,

s_geography_class, s_market_desc, s_market_manager, s_division_id, s_division_name,
s_company_id, s_company_name, s_street_number, s_street_name, s_street_type,
s_suite_number, s_city, s_county, s_state, s_zip, s_country, s_gmt_offset,
s tax precentage;

ANALYZE TABLE store_returns COMPUTE STATISTICS FOR COLUMNS sr_returned_date_sk, sr_return_time_sk, sr_item_sk, sr_customer_sk, sr_cdemo_sk, sr_hdemo_sk, sr_addr_sk, sr_store_sk, sr_reason_sk, sr_ticket_number, sr_return_quantity, sr_return_amt, sr_return_tax, sr_return_amt_inc_tax, sr_fee, sr_return_ship_cost, sr_refunded_cash, sr_reversed_charge, sr_store_credit, sr_net_loss;

ANALYZE TABLE store_sales COMPUTE STATISTICS FOR COLUMNS ss_sold_date_sk, ss_sold_time_sk, ss_item_sk, ss_customer_sk, ss_cdemo_sk, ss_hdemo_sk, ss_addr_sk, ss_store_sk, ss_promo_sk, ss_ticket_number, ss_quantity, ss_wholesale_cost, ss_list_price, ss_sales_price, ss_ext_discount_amt, ss_ext_sales_price, ss_ext_wholesale_cost, ss_ext_list_price, ss_ext_tax, ss_coupon_amt, ss_net_paid, ss_net_paid inc_tax, ss_net_profit;

ANALYZE TABLE time_dim COMPUTE STATISTICS FOR COLUMNS t_time_sk, t_time_id, t_time, t hour, t minute, t second, t am pm, t shift, t sub shift, t meal time;

ANALYZE TABLE warehouse COMPUTE STATISTICS FOR COLUMNS w_warehouse_sk, w_warehouse_id, w_warehouse_name, w_warehouse_sq_ft, w_street_number, w_street_name, w_street_type, w_suite_number, w_city, w_county, w_state, w_zip, w_country, w_gmt_offset;

ANALYZE TABLE web_page COMPUTE STATISTICS FOR COLUMNS wp_web_page_sk, wp_web_page_id, wp_rec_start_date, wp_rec_end_date, wp_creation_date_sk, wp_access_date_sk, wp_autogen_flag, wp_customer_sk, wp_url, wp_type, wp_char_count, wp link count, wp image count, wp max ad count;

ANALYZE TABLE web_returns COMPUTE STATISTICS FOR COLUMNS wr_returned_date_sk, wr_returned_time_sk, wr_item_sk, wr_refunded_customer_sk, wr_refunded_cdemo_sk, wr_refunded_hdemo_sk, wr_refunded_addr_sk, wr_returning_customer_sk, wr_returning_cdemo_sk, wr_returning_hdemo_sk, wr_returning_addr_sk, wr_web_page_sk, wr_reason_sk, wr_order_number, wr_return_quantity, wr_return_amt, wr_return_tax, wr_return_amt_inc_tax, wr_fee, wr_return_ship_cost, wr_refunded_cash, wr_reversed_charge, wr_account_credit, wr_net_loss;

ANALYZE TABLE web_sales COMPUTE STATISTICS FOR COLUMNS ws_sold_date_sk, ws_sold_time_sk, ws_ship_date_sk, ws_item_sk, ws_bill_customer_sk, ws_bill_cdemo_sk, ws_bill_hdemo_sk, ws_bill_addr_sk, ws_ship_customer_sk, ws_ship_cdemo_sk, ws_ship_hdemo_sk, ws_ship_addr_sk, ws_web_page_sk, ws_web_site_sk, ws_ship_mode_sk, ws_warehouse_sk, ws_promo_sk, ws_order_number, ws_quantity, ws_wholesale_cost, ws_list_price, ws_sales_price, ws_ext_discount_amt, ws_ext_sales_price, ws_ext_wholesale_cost, ws_ext_list_price, ws_ext_tax, ws_coupon_amt, ws_ext_ship_cost, ws_net_paid, ws_net_paid_inc_tax, ws_net_paid_inc_ship, ws_net_paid_inc_ship_tax, ws_net_profit;

ANALYZE TABLE web_site COMPUTE STATISTICS FOR COLUMNS web_site_sk, web_site_id, web_rec_start_date, web_rec_end_date, web_name, web_open_date_sk, web_close_date_sk, web_class, web_manager, web_mkt_id, web_mkt_class, web_mkt_desc, web_market_manager, web_company_id, web_company_name, web_street_number, web_street_name, web_street_type, web_suite_number, web_city, web_county, web_state, web_zip, web_country, web_gmt_offset, web_tax_percentage;

Stats views:

DBNAME=\$1
schema=\$2

db2 connect to \${DBNAME}
db2 -v set schema \${schema}

db2 -v "drop view cr_gview"
db2 -v "drop view sr_gview"
db2 -v "drop view ss_gview"

```
db2 -v "drop view wr gview"
db2 -v "drop view ws gview"
db2 -v "drop view c gview"
db2 -v "drop view inv gview"
db2 -v "drop view sv date dim"
db2 -v "create view CR GVIEW (c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12,
c13, c14, c15, c16, c1\overline{7}, c18, c19, c20, c21, c22, c23, c24, c25, c26, c27, c28,
c29, c30, c31, c32, c33, c34, c35, c36, c37, c38, c39, c40, c41, c42, c43, c44, c45, c46, c47, c48, c49, c50, c51, c52, c53, c54, c55, c56, c57, c58, c59, c60, c61, c62, c63, c64, c65, c66, c67, c68, c69, c70, c71, c72, c73, c74, c75, c76,
c77, c78, c79, c80, c81, c82, c83,c84, c85, c86, c87, c88, c89, c90, c91, c92, c93,
c94, c95, c96, c97, c98, c99, d d date) as
       select T2.*, T3.*, T4.*, T5.*, T6.*, T7.*, DATE(T5.D DATE) as D D DATE
               CATALOG RETURNS as T1,
               CATALOG PAGE as T2, CUSTOMER ADDRESS as T3, CUSTOMER as T4,
               DATE DIM as T5, CUSTOMER ADDRESS as T6, CUSTOMER as T7
                                           = T2.CP_CATALOG PAGE SK and
              T1.CR CATALOG PAGE SK
               T1.CR_REFUNDED_ADDR_SK
                                              = T3.CA_ADDRESS_SK
               T1.CR_REFUNDED_CUSTOMER_SK = T4.C_CUSTOMER_SK
T1.CR_RETURNED_DATE_SK = T5.D_DATE_SK
                                                                          and
                                             = T6.CA ADDRESS SK
               T1.CR RETURNING ADDR SK
                                                                          and
               T1.CR RETURNING CUSTOMER SK = T7.C CUSTOMER SK
) "
db2 -v "create view SR GVIEW as
        select T2.*, T3.*, T4.*, T5.*, DATE(T3.D DATE) as D D DATE
       from STORE RETURNS as T1,
               CUSTOMER as T2, DATE DIM as T3, TIME DIM as T4, STORE as T5
                                             = T2.C CUSTOMER SK
                                                                     and
       where T1.SR CUSTOMER SK
               T1.SR RETURNED DATE SK
                                              = T3.D DATE SK
                                                                          and
               T1.SR RETURN TIME SK
                                             = T4.T TIME SK
                                              = T5.S_STORE_SK
               T1.SR STORE SK
) "
db2 -v "create view SS GVIEW as
       select T2.*, T3.*, T4.*, DATE(T2.D_DATE) as D_D_DATE
               STORE SALES as T1,
               DATE DIM as T2, TIME DIM as T3, STORE as T4
       where
               T1.SS_SOLD_DATE_SK = T2.D_DATE_SK
                                                                         and
               T1.SS SOLD TIME SK
                                             = T3.T TIME SK
                                                                         and
                                              = T4.S STORE SK
               T1.SS STORE SK
) "
db2 -v "create view WR GVIEW (c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12,
c13, c14, c15, c16, c1\overline{7}, c18, c19, c20, c21, c22, c23, c24, c25, c26, c27, c28,
c29, c30, c31, c32, c33, c34, c35, c36, c37, c38, c39, c40, c41, c42, c43, c44, c45, c46, c47, c48, c49, c50, c51, c52, c53, c54, c55, c56, c57, c58, c59, c60,
c61, c62, c63, c64, c65, c66, c67, c68, c69, c70, c71, c72, c73, c74, c75, c76,
c77, c78, c79, c80, c81, c82, c83,c84, c85, c86, c87, c88, c89, c90, c91, c92, c93,
c94, c95, c96, c97, c98, c99, c100, c101, c102, c103, c104, c105, c106, c107, c108,
D D DATE) as
       select T2.*, T3.*, T4.*, T5.*, T6.*, T7.*, T8.*, DATE(T5.D DATE) as D D DATE
        from WEB RETURNS as T1,
               CUSTOMER ADDRESS as T2, CUSTOMER DEMOGRAPHICS as T3, CUSTOMER as T4,
DATE_DIM as T5,
               CUSTOMER ADDRESS as T6, CUSTOMER DEMOGRAPHICS as T7, CUSTOMER as T8
       where T1.WR REFUNDED ADDR SK = T2.CA ADDRESS SK
                                                                      and
               T1.WR REFUNDED CDEMO SK
                                              = T3.CD DEMO SK
               T1.WR REFUNDED CUSTOMER SK = T4.C CUSTOMER SK
                                                                          and
```

```
T1.WR RETURNED DATE SK
                                               = T5.D DATE SK
                                                                          and
               T1.WR RETURNING ADDR SK
                                              = T6.CA ADDRESS SK
                                                                          and
               T1.WR_RETURNING_CDEMO_SK = T7.CD_DEMO_SK
                                                                           and
               T1.WR_RETURNING_CUSTOMER_SK = T8.C_CUSTOMER_SK
) "
db2 -v "create view WS GVIEW (c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12,
c13, c14, c15, c16, c1\overline{7}, c18, c19, c20, c21, c22, c23, c24, c25, c26, c27, c28,
c29, c30, c31, c32, c33, c34, c35, c36, c37, c38, c39, c40, c41, c42, c43, c44, c45, c46, c47, c48, c49, c50, c51, c52, c53, c54, c55, c56, c57, c58, c59, c60,
c61, c62, c63, c64, c65, c66, c67, c68, c69, c70, c71, c72, c73, c74, c75, c76,
c77, c78, c79, c80, c81, c82, c83,c84, c85, c86, c87, c88, c89, c90, c91, c92,
D_D_DATE, E_D_DATE) as
       select T2.*, T3.*, T4.*, T5.*, DATE(T3.D DATE) as D D DATE, DATE(T5.D DATE)
as E D DATE
      from
              WEB SALES as T1,
              CUSTOMER as T2, DATE DIM as T3, CUSTOMER as T4, DATE DIM as T5
              T1.WS BILL CUSTOMER SK = T2.C CUSTOMER SK and
      where
              T1.WS_SHIP_CUSTOMER_SK
T1.WS_SHIP_DATE_SK
                                              = T4.C_CUSTOMER_SK
= T3.D_DATE_SK
                                                                           and
                                                                           and
              T1.WS SOLD DATE SK
                                               = T5.D DATE SK
) "
db2 -v "create view C_GVIEW (c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12,
c13, c14, c15, c16, \overline{c17}, c18, c19, c20, c21, c22, c23, c24, c25, c26, c27, c28, c29, c30, c31, c32, c33, c34, c35, c36, c37, c38, c39, c40, c41, c42, c43, c44,
c45, c46, c47, c48, c49, c50, c51, c52, c53, c54, c55, c56, c57, c58, c59, c60,
c61, c62, c63, c64, c65, c66, c67, c68, c69, c70, c71, c72, c73, c74, c75, c76,
c77, c78, D_D_DATE, E_D_DATE) as
      select T2.*, T3.*, T4.*, T5.*, DATE(T4.D DATE) as D D DATE, DATE(T5.D DATE)
as E D DATE
              CUSTOMER as T1,
              CUSTOMER_ADDRESS as T2, CUSTOMER DEMOGRAPHICS as T3, DATE DIM as T4,
DATE DIM as T5
      where T1.C CURRENT ADDR SK
                                               = T2.CA ADDRESS SK
              T1.C_CURRENT_CDEMO_SK = T3.CD_DEMO_SK
T1.C_FIRST_SALES_DATE_SK = T4.D_DATE_SK
                                                                          and
                                                                           and
              T1.C FIRST SHIPTO DATE SK = T5.D DATE SK
db2 -v "create view INV GVIEW as (select T2.*, DATE(T2.D DATE) as D D DATE from
INVENTORY as T1, DATE DIM as T2 where T1.INV DATE SK=T2.D DATE SK)"
db2 -v "create view SV DATE DIM as (select date(d date) as d d date from DATE DIM)"
db2 -v "alter view CR GVIEW enable query optimization"
db2 -v "alter view SR GVIEW enable query optimization"
db2 -v "alter view SS_GVIEW enable query optimization"
db2 -v "alter view WR_GVIEW enable query optimization"
db2 -v "alter view WS GVIEW enable query optimization"
db2 -v "alter view C_{\overline{G}VIEW} enable query optimization"
db2 -v "alter view INV GVIEW enable query optimization"
db2 -v "alter view SV_DATE_DIM enable query optimization"
time db2 -v "runstats on table SV DATE DIM with distribution"
time db2 -v "runstats on table CR GVIEW with distribution tablesample BERNOULLI(1)"
time db2 -v "runstats on table SR GVIEW with distribution tablesample BERNOULLI(1)"
time db2 -v "runstats on table SS GVIEW with distribution tablesample BERNOULLI(1)"
time db2 -v "runstats on table WR_GVIEW with distribution tablesample BERNOULLI(1)" time db2 -v "runstats on table WS_GVIEW with distribution tablesample BERNOULLI(1)"
time db2 -v "runstats on table C GVIEW with distribution tablesample BERNOULLI(1)"
time db2 -v "runstats on table INV GVIEW with distribution tablesample
BERNOULLI(1)"
```

```
db2 commit
db2 terminate
DBNAME=$1
schema=$2
db2 connect to ${DBNAME}
db2 -v set schema ${schema}
db2 -v "drop view cs gview1"
db2 -v "drop view cs_gview2"
db2 -v "drop view cs_gview3"
db2 -v "drop view cs_gview4"
db2 -v "drop view cs_gview5"
db2 -v "create view cs gview1 as (
select t2.* from CATALOG SALES as t1, CUSTOMER as t2
t1.CS BILL CUSTOMER SK=t2.C CUSTOMER SK
) "
db2 -v "create view cs gview2 as (
select t2.* from CATALOG SALES as t1, CATALOG PAGE as t2
t1.CS CATALOG PAGE SK=t2.CP CATALOG PAGE SK
db2 -v "create view cs_gview3 as (
select t2.* from CATALOG SALES as t1, CUSTOMER as t2
where
t1.CS_SHIP_CUSTOMER_SK=t2.C_CUSTOMER_SK
db2 -v "create view cs_gview4 as (
select t2.*, DATE(t2.D DATE) as D D DATE from CATALOG SALES as t1, DATE DIM as t2
where
t1.CS_SHIP_DATE_SK=t2.D_DATE_SK
db2 -v "create view cs_gview5 as (
select t2.*, DATE(t2.D_DATE) as D_D_DATE from CATALOG_SALES as t1, DATE DIM as t2
where
t1.CS SOLD DATE SK=t2.D DATE SK
db2 -v "alter view cs gview1 enable query optimization"
db2 -v "alter view cs gview2 enable query optimization"
db2 -v "alter view cs gview3 enable query optimization"
db2 -v "alter view cs_gview4 enable query optimization"
db2 -v "alter view cs_gview5 enable query optimization"
time db2 -v "runstats on table cs_gview1 with distribution tablesample
BERNOULLI(1)"
time db2 -v "runstats on table cs_gview2 with distribution tablesample
BERNOULLI(1)"
time db2 -v "runstats on table cs gview3 with distribution tablesample
BERNOULLI(1)"
time db2 -v "runstats on table cs\_gview4 with distribution tablesample
BERNOULLI(1)"
time db2 -v "runstats on table cs_gview5 with distribution tablesample
BERNOULLI(1)"
db2 commit
db2 terminate
```

Informational Constraints:

```
set schema $schema;
-- primary key definitions
alter table call_center
    add primary key (cc call center sk)
    not enforced enable query optimization;
commit work;
alter table catalog_page
    add primary key (cp_catalog_page_sk)
    not enforced enable query optimization;
commit work;
alter table catalog_returns
    add primary key (cr_item_sk, cr_order_number)
    not enforced enable query optimization;
commit work;
alter table catalog sales
    add primary key (cs_item_sk, cs_order_number)
    not enforced enable query optimization;
commit work;
alter table customer
    add primary key (c customer sk)
   not enforced enable query optimization;
commit work;
alter table customer address
    add primary key (ca address sk)
    not enforced enable query optimization;
commit work;
alter table customer_demographics
   add primary key (cd demo sk)
    not enforced enable query optimization;
commit work;
alter table date dim
    add primary key (d date sk)
    not enforced enable query optimization;
commit work;
alter table household_demographics
   add primary key (hd_demo_sk)
   not enforced enable query optimization;
commit work;
alter table income band
    add primary key (ib income band sk)
    not enforced enable query optimization;
commit work;
alter table inventory
   add primary key (inv date sk, inv item sk, inv warehouse sk)
   not enforced enable query optimization;
commit work;
alter table item
    add primary key (i item sk)
    not enforced enable query optimization;
```

```
commit work;
alter table promotion
   add primary key (p_promo_sk)
   not enforced enable query optimization;
commit work;
alter table reason
   add primary key (r_reason_sk)
   not enforced enable query optimization;
commit work;
alter table ship mode
   add primary key (sm_ship_mode_sk)
   not enforced enable query optimization;
commit work;
alter table store
   add primary key (s store sk)
   not enforced enable query optimization;
commit work;
alter table store returns
   add primary key (sr item sk, sr ticket number)
   not enforced enable query optimization;
commit work;
alter table store sales
   add primary key (ss_item_sk, ss_ticket_number)
   not enforced enable query optimization;
commit work;
alter table time dim
   add primary key (t_time_sk)
   not enforced enable query optimization;
commit work;
alter table warehouse
   add primary key (w_warehouse_sk)
   not enforced enable query optimization;
commit work;
alter table web_page
   add primary key (wp_web_page_sk)
   not enforced enable query optimization;
commit work;
alter table web_returns
   add primary key (wr_item_sk, wr_order_number)
   not enforced enable query optimization;
commit work;
alter table web_sales
   add primary key (ws_item_sk, ws_order_number)
   not enforced enable query optimization;
commit work;
alter table web_site
   add primary key (web_site_sk)
   not enforced enable query optimization;
commit work;
-- foreign key definitions
_____
```

-- tables with no FKs

```
-- customer address
-- customer demographics
___
    item
     date dim
     warehouse
-- ship_mode
-- time dim
    reason
    income_band
alter table promotion
    add constraint fk1 foreign key (p_start_date_sk)
       references date dim (d date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table promotion
    add constraint fk2 foreign key (p_end_date_sk)
       references date dim (d date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
alter table promotion
    add constraint fk3 foreign key (p item sk)
       references item (i item sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table store
    add constraint fk foreign key (s_closed_date_sk)
    references date_dim (d_date_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table call_center
    add constraint fkl foreign key (cc_closed_date_sk)
    references date_dim (d_date_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table call center
    add constraint fk2 foreign key (cc_open_date_sk)
    references date_dim (d_date_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table customer
    add constraint fk1 foreign key (c_current_cdemo_sk)
       references customer demographics (cd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION:
commit work;
alter table customer
    add constraint fk2 foreign key (c_current_hdemo_sk)
        references household demographics (hd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table customer
    add constraint fk3 foreign key (c_current_addr_sk)
    references customer_address (ca_address_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table customer
    add constraint fk4 foreign key (c_first_shipto_date_sk)
      references date dim (d date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table customer
    add constraint fk5 foreign key (c_first_sales_date_sk)
    references date_dim (d_date_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
```

```
alter table web site
    add constraint fk1 foreign key (web open date sk)
       references date_dim (d_date_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web site
    add constraint fk2 foreign key (web close date sk)
       references date dim (d date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table catalog_page
    add constraint fk1 foreign key (cp_start_date_sk)
       references date dim (d date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table catalog_page
    add constraint fk2 foreign key (cp end date sk)
       references date dim (d date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
alter table household_demographics
    add constraint fk foreign key (hd_income_band_sk)
      references income band (ib income band sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table web page
    add constraint fk1 foreign key (wp_creation_date_sk)
       references date dim (d date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web page
    add constraint fk2 foreign key (wp access date sk)
      references date_dim (d_date_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web page
    add constraint fk3 foreign key (wp customer sk)
       references customer (c_customer_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table store_sales
    add constraint fkl foreign key (ss_sold_date_sk) references date_dim (d_date_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table store sales
    add constraint fk2 foreign key (ss_sold_time_sk)
    references time_dim (t_time_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table store sales
    add constraint fk3a foreign key (ss item sk)
       references item (i item sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table store sales
    add constraint fk4 foreign key (ss_customer_sk)
       references customer (c_customer_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table store sales
    add constraint fk5 foreign key (ss cdemo sk)
       references customer_demographics (cd_demo_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table store sales
```

```
add constraint fk6 foreign key (ss hdemo sk)
       references household demographics (hd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table store sales
    add constraint fk7 foreign key (ss addr sk)
       references customer address (ca address sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table store_sales
    add constraint fk8 foreign key (ss store sk)
       references store (s_store_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table store sales
    add constraint fk9 foreign key (ss promo sk)
      references promotion (p promo sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table store returns
    add constraint fk1 foreign key (sr_returned_date_sk)
      references date dim (d date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table store returns
    add constraint fk2 foreign key (sr return time sk)
      references time dim (t time sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
alter table store returns
    add constraint fk3a foreign key (sr_item_sk)
      references item (i item sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table store returns
    add constraint fk3b foreign key (sr item sk, sr ticket number)
       references store_sales (ss_item_sk, ss_ticket_number) NOT ENFORCED ENABLE
QUERY OPTIMIZATION;
commit work;
alter table store returns
    add constraint fk4 foreign key (sr customer sk)
      references customer (c customer sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table store_returns
    add constraint fk5 foreign key (sr cdemo sk)
       references customer_demographics (cd_demo_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table store returns
    add constraint fk6 foreign key (sr hdemo sk)
      references household_demographics (hd_demo_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table store returns
    add constraint fk7 foreign key (sr_addr_sk)
      references customer address (ca address sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table store returns
    add constraint fk8 foreign key (sr store sk)
       references store (s store sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
```

```
commit work:
alter table store returns
    add constraint fk9 foreign key (sr_reason_sk)
       references reason (r reason sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table catalog sales
    add constraint fk1 foreign key (cs_sold_date_sk)
      references date dim (d date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table catalog sales
    add constraint Tk2 foreign key (cs_sold_time_sk)
      references time dim (t time sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table catalog_sales
    add constraint fk3 foreign key (cs ship date sk)
      references date_dim (d_date_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
alter table catalog sales
    add constraint fk4 foreign key (cs bill customer sk)
      references customer (c customer sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table catalog_sales
    add constraint fk5 foreign key (cs bill cdemo sk)
       references customer demographics (cd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog_sales
    add constraint fk6 foreign key (cs bill hdemo sk)
       references household demographics (hd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog_sales
    add constraint fk7 foreign key (cs_bill_addr sk)
      references customer address (ca address sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog sales
    add constraint fk8 foreign key (cs ship customer sk)
       references customer (c customer sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table catalog sales
    add constraint fk9 foreign key (cs ship cdemo sk)
      references customer demographics (cd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog sales
    add constraint fk10 foreign key (cs_ship_hdemo_sk)
       references household_demographics (hd_demo_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog sales
    add constraint fk11 foreign key (cs_ship_addr_sk)
      references customer address (ca address sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION:
commit work;
```

```
alter table catalog_sales
    add constraint fk12 foreign key (cs call center sk)
       references call_center (cc_call_center_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog_sales
    add constraint fk13 foreign key (cs catalog page sk)
       references catalog_page (cp_catalog_page_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog_sales
    add constraint fk14 foreign key (cs_ship_mode_sk)
       references ship mode (sm ship mode sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog sales
    add constraint fk15 foreign key (cs_warehouse_sk)
       references warehouse (w warehouse sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION:
commit work;
alter table catalog_sales
    add constraint fk16a foreign key (cs_item_sk)
       references item (i item sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table catalog sales
    add constraint fk17 foreign key (cs_promo_sk)
      references promotion (p_promo sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table catalog returns
    add constraint fk1 foreign key (cr_returned_date_sk)
       references date_dim (d_date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table catalog_returns
    add constraint fk2 foreign key (cr_returned time sk)
      references time dim (t time sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
alter table catalog returns
    add constraint fk3 foreign key (cr item sk, cr order number)
       references catalog sales (cs item sk, cs order number) NOT ENFORCED ENABLE
OUERY OPTIMIZATION;
commit work;
alter table catalog returns
    add constraint fk4 foreign key (cr item sk)
      references item (i item sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
alter table catalog_returns
    add constraint fk5 foreign key (cr refunded customer sk)
      references customer (c_customer_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table catalog_returns
    add constraint fk6 foreign key (cr refunded cdemo sk)
       references customer demographics (cd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog returns
    add constraint fk7 foreign key (cr refunded hdemo sk)
```

```
references household_demographics (hd_demo_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work:
alter table catalog_returns
    add constraint fk8 foreign key (cr_refunded_addr_sk)
       references customer address (ca address sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog_returns
    add constraint fk9 foreign key (cr_returning_customer_sk)
      references customer (c customer sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table catalog_returns
    add constraint fk10 foreign key (cr_returning_cdemo_sk)
       references customer demographics (cd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog returns
    add constraint fk11 foreign key (cr returning hdemo sk)
       references household demographics (hd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog_returns
    add constraint fk12 foreign key (cr returning addr sk)
       references customer address (ca address sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog_returns
    add constraint fk13 foreign key (cr call center sk)
       references call center (cc call center sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog_returns
    add constraint fk14 foreign key (cr_catalog_page_sk)
      references catalog_page (cp_catalog_page_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog returns
    add constraint fk15 foreign key (cr_ship_mode_sk)
       references ship mode (sm ship mode sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog returns
    add constraint fk16 foreign key (cr warehouse sk)
       references warehouse (w warehouse sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table catalog_returns
    add constraint fk17 foreign key (cr_reason_sk)
       references reason (r reason sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web sales
    add constraint fk1 foreign key (ws_sold_date_sk)
      references date dim (d date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web sales
```

```
add constraint fk2 foreign key (ws sold time sk)
      references time dim (t time sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web sales
    add constraint fk3 foreign key (ws ship date sk)
      references date dim (d date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
alter table web sales
    add constraint fk4a foreign key (ws_item_sk)
      references item (i_item_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web sales
    add constraint fk5 foreign key (ws bill customer sk)
      references customer (c customer sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web sales
    add constraint fk6 foreign key (ws bill cdemo sk)
      references customer demographics (cd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION:
commit work;
alter table web sales
    add constraint fk7 foreign key (ws bill hdemo sk)
       references household demographics (hd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table web sales
    add constraint fk8 foreign key (ws bill addr sk)
      references customer_address (ca_address_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table web sales
    add constraint fk9 foreign key (ws_ship_customer_sk)
      references customer (c customer sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web sales
    add constraint fk10 foreign key (ws_ship_cdemo_sk)
      references customer_demographics (cd_demo_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table web sales
   add constraint fk11 foreign key (ws ship hdemo sk)
      references household demographics (hd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table web sales
   add constraint fk12 foreign key (ws ship addr sk)
       references customer address (ca address sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table web sales
    add constraint fk13 foreign key (ws web page sk)
      references web_page (wp_web_page_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web sales
    add constraint fk14 foreign key (ws web site sk)
       references web site (web site sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
```

```
commit work;
alter table web sales
    add constraint fk15 foreign key (ws_ship_mode_sk)
       references ship mode (sm ship mode sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION:
commit work;
alter table web_sales
    add constraint fk16 foreign key (ws warehouse sk)
       references warehouse (w_warehouse_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION:
commit work;
alter table web sales
    add constraint fk17 foreign key (ws_promo_sk)
    references promotion (p_promo_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web_returns
    add constraint fk1 foreign key (wr returned date sk)
       references date dim (d date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web returns
    add constraint fk2 foreign key (wr_returned_time_sk)
       references time dim (t time sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web returns
    add constraint fk3a foreign key (wr_item_sk)
      references item (i item sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web returns
    add constraint fk3b foreign key (wr_item_sk, wr_order_number)
       references web sales (ws item sk, ws order number) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table web returns
    add constraint fk4 foreign key (wr_refunded_customer_sk)
      references customer (c customer sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web returns
    add constraint fk5 foreign key (wr_refunded_cdemo_sk)
       references customer demographics (cd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table web returns
    add constraint fk6 foreign key (wr refunded hdemo sk)
       references household demographics (hd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table web returns
    add constraint fk7 foreign key (wr_refunded_addr_sk)
       references customer address (ca \overline{a}ddress \overline{s}\overline{k}) NO\overline{T} ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table web returns
    add constraint fk8 foreign key (wr returning customer sk)
       references customer (c customer sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
```

```
alter table web_returns
   add constraint fk9 foreign key (wr returning cdemo sk)
       references customer demographics (cd demo sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table web returns
    add constraint fk10 foreign key (wr returning hdemo sk)
      references household_demographics (hd_demo_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table web returns
    add constraint fk11 foreign key (wr_returning_addr_sk)
       references customer address (ca address sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
alter table web returns
    add constraint fk12 foreign key (wr_web_page_sk)
      references web page (wp web page sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table web returns
    add constraint fk13 foreign key (wr_reason_sk)
      references reason (r_reason_sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table inventory
    add constraint fk1 foreign key (inv date sk)
      references date dim (d date sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
alter table inventory
    add constraint fk2 foreign key (inv item sk)
      references item (i item sk) NOT ENFORCED ENABLE QUERY OPTIMIZATION;
commit work;
alter table inventory
    add constraint fk3 foreign key (inv warehouse sk)
       references warehouse (w_warehouse_sk) NOT ENFORCED ENABLE QUERY
OPTIMIZATION;
commit work;
```

F.2 Impala Load & Analyze scripts:

<u>Load:</u>

```
#!/bin/bash
```

```
impala-shell -d tpcds10000g <<EOF
create table date_dim like et_date_dim stored as parquetfile;
insert overwrite table date_dim select * from et_date_dim;

create table time_dim like et_time_dim stored as parquetfile;
insert overwrite table time_dim select * from et_time_dim;

create table customer like et customer stored as parquetfile;</pre>
```

```
insert overwrite table customer select * from et_customer;
create table customer address like et customer address stored as parquetfile;
insert overwrite table customer address select * from et customer address;
create table customer demographics like et customer demographics stored as
insert overwrite table customer demographics select * from
et customer demographics;
create table household_demographics like et_household_demographics stored as
insert overwrite table household demographics select * from
et_household_demographics;
create table item like et_item stored as parquetfile;
insert overwrite table item select * from et_item;
create table promotion like et promotion stored as parquetfile;
insert overwrite table promotion select * from et promotion;
create table store like et_store stored as parquetfile;
insert overwrite table store select * from et_store;
create table store returns like et store returns stored as parquetfile;
insert overwrite table store returns select * from et store returns;
create table web sales like et web sales stored as parquetfile;
insert overwrite table web sales select * from et web sales;
create table web returns like et web returns stored as parquetfile;
insert overwrite table web returns select * from et web returns;
create table catalog sales like et catalog sales stored as parquetfile;
insert overwrite table catalog_sales select * from et_catalog_sales;
create table catalog_returns like et_catalog_returns stored as parquetfile;
insert overwrite table catalog_returns select * from et_catalog_returns;
create table store sales like et store sales stored as parquetfile;
insert overwrite table store_sales select * from et_store_sales;
create table call_center like et_call_center stored as parquetfile;
insert overwrite table call center select * from et call center;
```

```
create table income_band like et_income_band stored as parquetfile;
insert overwrite table income band select * from et income band;
create table ship_mode like et_ship_mode stored as parquetfile;
insert overwrite table ship_mode select * from et_ship_mode;
create table reason like et reason stored as parquetfile;
insert overwrite table reason select * from et reason;
create table reason like et_reason stored as parquetfile;
insert overwrite table reason select * from et_reason;
create table inventory like et_inventory stored as parquetfile;
insert overwrite table inventory select * from et_inventory;
create table warehouse like et_warehouse stored as parquetfile;
insert overwrite table warehouse select * from et_warehouse;
create table web_site like et_web_site stored as parquetfile;
insert overwrite table web_site select * from et_web_site;
create table web_page like et_web_page stored as parquetfile;
insert overwrite table web page select * from et web page;
create table catalog_page like et_catalog_page stored as parquetfile;
insert overwrite table catalog_page select * from et_catalog_page;
show tables;
EOF
```

Analyze:

#!/bin/bash

compute stats item;
compute stats promotion;
compute stats reason;

```
impala-shell -d TPCDS10000G <<EOF
compute stats call_center;
compute stats catalog_page;
compute stats catalog_returns;
compute stats catalog_sales;
compute stats customer;
compute stats customer_address;
compute stats customer_demographics;
compute stats date_dim;
compute stats household_demographics;
compute stats income_band;
compute stats inventory;</pre>
```

```
compute stats ship_mode;
compute stats store;
compute stats store_returns;
compute stats store_sales;
compute stats time_dim;
compute stats warehouse;
compute stats web_page;
compute stats web_returns;
compute stats web_sales;
compute stats web_site;
EOF
```

F.3 Hive0.13 Load & Analyze scripts:

Load:

```
-- Use the following to execute this script and create the tables in Hive:
    $HIVE HOME/bin/hive -hiveconf DB NAME=300 -f
$testhome/ddl/070.hive.populateTables.ORC.sql
USE TPCDS${hiveconf:DB NAME}G HIVE;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.customer address
  SELECT * FROM customer address;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB_NAME}G_HIVE_ORC_B.customer_demographics
  SELECT * FROM customer demographics;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.date dim
   SELECT * FROM date dim;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.warehouse
  SELECT * FROM warehouse;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.ship mode
  SELECT * FROM ship mode;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.time dim
  SELECT * FROM time_dim;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.reason
  SELECT * FROM reason;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.income band
  SELECT * FROM income band;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.item
  SELECT * FROM item:
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.store
  SELECT * FROM store;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.call center
  SELECT * FROM call center;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB_NAME}G_HIVE_ORC_B.customer
  SELECT * FROM customer;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.web site
  SELECT * FROM web site;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB_NAME}G_HIVE_ORC_B.household_demographics
   SELECT * FROM household demographics;
```

```
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.web page
   SELECT * FROM web page;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.promotion
  SELECT * FROM promotion;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.catalog page
  SELECT * FROM catalog_page;
set mapred.min.split.size=128000000;
set mapred.max.split.size=128000000;
set hive.enforce.bucketing=true;
set hive.enforce.sorting=true;
set hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.input.format=org.apache.hadoop.hive.ql.io.BucketizedHiveInputFormat;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.store returns
  SELECT * FROM store returns;
set mapred.min.split.size=128000000;
set mapred.max.split.size=128000000;
set hive.enforce.bucketing=true;
set hive.enforce.sorting=true;
set hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.input.format=org.apache.hadoop.hive.ql.io.BucketizedHiveInputFormat;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.inventory
  SELECT * FROM inventory;
set mapred.min.split.size=512000000;
set mapred.max.split.size=512000000;
set hive.enforce.bucketing=true;
set hive.enforce.sorting=true;
set hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.input.format=org.apache.hadoop.hive.ql.io.BucketizedHiveInputFormat;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.catalog returns
  SELECT * FROM catalog returns;
set mapred.min.split.size=128000000;
set mapred.max.split.size=128000000;
set hive.enforce.bucketing=true;
set hive.enforce.sorting=true;
set hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.input.format=org.apache.hadoop.hive.ql.io.BucketizedHiveInputFormat;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.web returns
  SELECT * FROM web returns;
set mapred.min.split.size=128000000;
set mapred.max.split.size=128000000;
set hive.enforce.bucketing=true;
set hive.enforce.sorting=true;
set hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.input.format=org.apache.hadoop.hive.ql.io.BucketizedHiveInputFormat;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.web sales
  SELECT * FROM web sales;
set mapred.min.split.size=1000000000;
set mapred.max.split.size=1000000000;
set hive.enforce.bucketing=true;
```

```
set hive.enforce.sorting=true;
set hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.input.format=org.apache.hadoop.hive.ql.io.BucketizedHiveInputFormat;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.catalog sales
  SELECT * FROM catalog sales;
set mapred.min.split.size=1500000000;
set mapred.max.split.size=1500000000;
set hive.enforce.bucketing=true;
set hive.enforce.sorting=true;
set hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.input.format=org.apache.hadoop.hive.ql.io.BucketizedHiveInputFormat;
INSERT OVERWRITE TABLE TPCDS${hiveconf:DB NAME}G HIVE ORC B.store sales
   SELECT * FROM store sales;
Analyze:
USE TPCDS${hiveconf:DB_NAME}G_HIVE_ORC_B_NEW;
ANALYZE TABLE call_center COMPUTE STATISTICS;
ANALYZE TABLE call center COMPUTE STATISTICS FOR COLUMNS
      cc call center sk, cc call center id, cc rec start date,
      cc_rec_end_date, cc_closed_date_sk, cc_open_date_sk, cc_name,
      cc_class, cc_employees, cc_sq_ft, cc_hours, cc_manager,
      cc_mkt_id, cc_mkt_class, cc_mkt_desc, cc_market_manager,
      cc_division, cc_division_name, cc_company, cc_company_name,
      cc street number, cc street name, cc street type,
      cc suite number, cc city, cc county, cc state, cc zip,
      cc country, cc gmt offset, cc tax percentage;
ANALYZE TABLE catalog page COMPUTE STATISTICS;
ANALYZE TABLE catalog page COMPUTE STATISTICS FOR COLUMNS
      cp_catalog_page_sk, cp_catalog_page_id, cp_start_date_sk,
      cp_end_date_sk, cp_department, cp_catalog_number,
      cp_catalog_page_number, cp_description, cp_type;
ANALYZE TABLE catalog returns COMPUTE STATISTICS;
ANALYZE TABLE catalog_returns COMPUTE STATISTICS FOR COLUMNS
      cr returned date sk, cr returned time sk, cr item sk,
      cr_refunded_customer_sk, cr_refunded_cdemo_sk,
      cr_refunded_hdemo_sk, cr_refunded_addr_sk,
cr_returning_customer_sk, cr_returning_cdemo_sk,
      cr returning hdemo sk, cr returning addr sk, cr call center sk,
      cr catalog page sk, cr ship mode sk, cr warehouse sk,
      cr_reason_sk, cr_order_number, cr_return_quantity,
      cr_return_amount, cr_return_tax, cr_return_amt_inc_tax, cr_fee,
      cr_return_ship_cost, cr_refunded_cash, cr_reversed_charge,
      cr_store_credit, cr_net_loss;
ANALYZE TABLE catalog_sales COMPUTE STATISTICS;
cs_sold_date_sk, cs_sold_time_sk, cs_ship_date_sk,
cs_bill_customer_sk, cs_bill_cdemo_sk, cs_bill_hdemo_sk,
      cs bill addr sk, cs ship customer sk, cs ship cdemo sk,
      cs ship hdemo sk, cs ship addr sk, cs call center sk,
      cs_catalog_page_sk, cs_ship_mode_sk, cs_warehouse_sk,
      cs_item_sk, cs_promo_sk, cs_order_number, cs_quantity,
      cs_wholesale_cost, cs_list_price, cs_sales_price,
      cs ext discount amt, cs ext sales price, cs ext wholesale cost,
      cs_ext_list_price, cs_ext_tax, cs_coupon_amt, cs_ext_ship_cost,
      cs_net_paid, cs_net_paid_inc_tax, cs_net_paid_inc_ship,
      cs_net_paid_inc_ship_tax, cs_net_profit;
```

```
ANALYZE TABLE customer COMPUTE STATISTICS;
ANALYZE TABLE customer COMPUTE STATISTICS FOR COLUMNS
       c customer sk, c customer id, c current cdemo sk,
       c_current_hdemo_sk, c_current_addr_sk, c_first_shipto_date_sk, c_first_sales_date_sk, c_salutation, c_first_name, c_last_name, c_preferred_cust_flag, c_birth_day, c_birth_month,
       c_birth_year, c_birth_country, c_login, c_email_address,
       c last review date;
ANALYZE TABLE customer_address COMPUTE STATISTICS;
ANALYZE TABLE customer_address COMPUTE STATISTICS FOR COLUMNS
       ca_address_sk, ca_address_id, ca_street_number, ca_street_name,
       ca_street_type, ca_suite_number, ca_city, ca_county, ca_state,
       ca_zip, ca_country, ca_gmt_offset, ca_location_type;
ANALYZE TABLE customer demographics COMPUTE STATISTICS;
ANALYZE TABLE customer demographics COMPUTE STATISTICS FOR COLUMNS
       cd demo sk, cd gender, cd marital status, cd education status,
       cd purchase estimate, cd credit rating, cd dep count,
       cd dep employed count, cd dep college count;
ANALYZE TABLE date dim COMPUTE STATISTICS;
ANALYZE TABLE date dim COMPUTE STATISTICS FOR COLUMNS
       d_date_sk, d_date_id, d_date, d_month_seq, d_week_seq,
       d_quarter_seq, d_year, d_dow, d_moy, d_dom, d_qoy, d_fy_year,
       d_fy_quarter_seq, d_fy_week_seq, d_day_name, d_quarter_name,
       d_holiday, d_weekend, d_following_holiday, d_first_dom,
       d_last_dom, d_same_day_ly, d_same_day_lq, d_current_day,
       d_current_week, d_current_month, d_current_quarter,
       d current year;
ANALYZE TABLE household_demographics COMPUTE STATISTICS;
ANALYZE TABLE household_demographics COMPUTE STATISTICS FOR COLUMNS
       hd_demo_sk, hd_income_band_sk, hd_buy_potential, hd_dep_count,
       hd vehicle count;
ANALYZE TABLE income_band COMPUTE STATISTICS;
ANALYZE TABLE income band COMPUTE STATISTICS FOR COLUMNS
       ib_income_band_sk, ib_lower_bound, ib_upper_bound;
ANALYZE TABLE inventory COMPUTE STATISTICS;
ANALYZE TABLE inventory COMPUTE STATISTICS FOR COLUMNS inv_date_sk, inv_item_sk, inv_warehouse_sk,
       inv_quantity_on_hand;
ANALYZE TABLE item COMPUTE STATISTICS;
ANALYZE TABLE item COMPUTE STATISTICS FOR COLUMNS
       i_item_sk, i_item_id, i_rec_start_date, i_rec_end_date,
       i_item_desc, i_current_price, i_wholesale_cost, i_brand_id,
i_brand, i_class_id, i_class, i_category_id, i_category,
       i manufact id, i manufact, i size, i formulation, i color,
       i units, i container, i manager id, i product name;
ANALYZE TABLE promotion COMPUTE STATISTICS;
ANALYZE TABLE promotion COMPUTE STATISTICS FOR COLUMNS
       p_promo_sk, p_promo_id, p_start_date_sk, p_end_date_sk,
       p_item_sk, p_cost, p_response_target, p_promo_name,
       p_channel_dmail, p_channel_email, p_channel_catalog,
       p_channel_tv, p_channel_radio, p_channel_press,
       p channel event, p channel demo, p channel details, p purpose,
       p discount active;
ANALYZE TABLE reason COMPUTE STATISTICS;
ANALYZE TABLE reason COMPUTE STATISTICS FOR COLUMNS
       r reason sk, r reason id, r reason desc;
ANALYZE TABLE ship mode COMPUTE STATISTICS;
ANALYZE TABLE ship mode COMPUTE STATISTICS FOR COLUMNS
```

```
sm_ship_mode_sk, sm_ship_mode_id, sm_type, sm_code, sm_carrier,
      sm contract;
ANALYZE TABLE store COMPUTE STATISTICS;
ANALYZE TABLE store COMPUTE STATISTICS FOR COLUMNS
      s_store_sk, s_store_id, s_rec_start_date, s_rec_end_date,
      s closed date sk, s store name, s number employees,
      s floor space, s hours, s manager, s market id,
      s_geography_class, s_market_desc, s_market_manager,
      s_division_id, s_division_name, s_company_id, s_company_name,
      s_street_number, s_street_name, s_street_type, s_suite_number,
      s_city, s_county, s_state, s_zip, s_country, s_gmt_offset,
      s_tax_precentage;
ANALYZE TABLE store_returns COMPUTE STATISTICS;
ANALYZE TABLE store returns COMPUTE STATISTICS FOR COLUMNS
      sr_returned_date_sk, sr_return_time_sk, sr_item_sk,
      sr customer sk, sr cdemo sk, sr hdemo sk, sr addr sk,
      sr store sk, sr reason sk, sr ticket number,
      sr_return_quantity, sr_return_amt, sr_return_tax,
      sr_return_amt_inc_tax, sr_fee, sr_return_ship_cost,
sr_refunded_cash, sr_reversed_charge, sr_store_credit,
      sr net loss;
ANALYZE TABLE store_sales COMPUTE STATISTICS;
ANALYZE TABLE store_sales COMPUTE STATISTICS FOR COLUMNS
       ss_sold_date_sk, ss_sold_time_sk, ss_item_sk, ss_customer_sk,
      ss_cdemo_sk, ss_hdemo_sk, ss_addr_sk, ss_store_sk, ss_promo_sk,
      ss ticket_number, ss_quantity, ss_wholesale_cost,
      ss list price, ss sales price, ss ext discount amt,
      ss_ext_sales_price, ss_ext_wholesale_cost, ss_ext_list_price,
      ss_ext_tax, ss_coupon_amt, ss_net_paid, ss_net_paid_inc_tax,
      ss net profit;
ANALYZE TABLE time dim COMPUTE STATISTICS;
ANALYZE TABLE time dim COMPUTE STATISTICS FOR COLUMNS
       t_time_sk, t_time_id, t_time, t_hour, t_minute, t_second,
      t am pm, t shift, t sub shift, t meal time;
ANALYZE TABLE warehouse COMPUTE STATISTICS;
ANALYZE TABLE warehouse COMPUTE STATISTICS FOR COLUMNS
      w_warehouse_sk, w_warehouse_id, w_warehouse_name,
      w_warehouse_sq_ft, w_street_number, w_street_name,
      w_street_type, w_suite_number, w_city, w_county, w_state,
      w_zip, w_country, w_gmt_offset;
ANALYZE TABLE web page COMPUTE STATISTICS;
ANALYZE TABLE web page COMPUTE STATISTICS FOR COLUMNS
      wp web page sk, wp web page id, wp rec start date,
      wp_rec_end_date, wp_creation_date_sk, wp_access_date_sk,
      wp autogen flag, wp customer sk, wp url, wp type,
      wp char count, wp link count, wp image count, wp max ad count;
ANALYZE TABLE web_returns COMPUTE STATISTICS;
ANALYZE TABLE web returns COMPUTE STATISTICS FOR COLUMNS
      wr_returned_date_sk, wr_returned_time_sk, wr_item_sk,
      wr refunded customer sk, wr refunded cdemo sk,
      wr_refunded_hdemo_sk, wr_refunded_addr_sk,
      wr_returning_customer_sk, wr_returning_cdemo_sk,
         returning hdemo sk, wr returning addr sk, wr web page sk,
      wr_reason_sk, wr_order_number, wr_return_quantity,
      wr return amt, wr return tax, wr return amt inc tax, wr fee,
      wr return ship cost, wr refunded cash, wr reversed charge,
      wr_account_credit, wr_net_loss;
ANALYZE TABLE web sales COMPUTE STATISTICS;
ANALYZE TABLE web sales COMPUTE STATISTICS FOR COLUMNS
      ws sold date sk, ws sold time sk, ws ship date sk, ws item sk,
```

ws_bill_customer_sk, ws_bill_cdemo_sk, ws_bill_hdemo_sk,
ws_bill_addr_sk, ws_ship_customer_sk, ws_ship_cdemo_sk,
ws_ship_hdemo_sk, ws_ship_addr_sk, ws_web_page_sk,
ws_web_site_sk, ws_ship_mode_sk, ws_warehouse_sk, ws_promo_sk,
ws_order_number, ws_quantity, ws_wholesale_cost, ws_list_price,
ws_sales_price, ws_ext_discount_amt, ws_ext_sales_price,
ws_ext_wholesale_cost, ws_ext_list_price, ws_ext_tax,
ws_coupon_amt, ws_ext_ship_cost, ws_net_paid,
ws_net_paid_inc_tax, ws_net_paid_inc_ship,
ws_net_paid_inc_ship_tax, ws_net_profit;

ANALYZE TABLE web_site COMPUTE STATISTICS;

ANALYZE TABLE web_site COMPUTE STATISTICS FOR COLUMNS

web_site_sk, web_site_id, web_rec_start_date, web_rec_end_date,

web_name, web_open_date_sk, web_close_date_sk, web_class,

web_manager, web_mkt_id, web_mkt_class, web_mkt_desc,

web_market_manager, web_company_id, web_company_name,

web_street_number, web_street_name, web_street_type,

web_suite_number, web_city, web_county, web_state, web_zip,

web_country, web_gmt_offset, web_tax_percentage;

Appendix G: Attestation Letter:





Benchmark sponsor: Berni Schiefer

IBM

8200 Warden Avenue

Markham, Ontario, L6C 1C7

October 24, 2014

At IBM's request I verified the implementation and results of a **10TB Big Data Decision Support** (Hadoop-DS) benchmark, with most features derived from the TPC-DS Benchmark.

The Hadoop-DS benchmark was executed on three identical clusters, each running a different query engine. The test clusters were configured as follows:

IBM x3650BD Cluster - 17 Nodes (configuration per node)

Operating System: Red Hat Enterprise Linux 6.4

CPUs 2 x Intel Xeon Processor E5-2680 v2 (2.8 GHz, 25MB L3)

Memory 128GB (1866MHz DDR3)

Storage 10 x 2TB SATA 3.5" HDD

The intent of the benchmark was to measure the performance of the following three Hadoop based SQL query engines, all executing an identical workload:

- IBM BigInsights Big SQL v3.0
- Cloudera CDH 5.1.2 Impala v1.4.1
- HortonWorks Hive v0.13

The results were:

	Big SQL	Impala	Hive
Single-User Run Duration (h:m:s)	0:48:28	2:55:36	4:25:49
Multi-User Run Duration (h:m:s)	1:55:45	4:08:40	16:32:30
Qph Hadoop-DS @10TB - Single-User	5,694	1,571	1,038
Qph Hadoop-DS @10TB - Multi-User (x4)	9,537	4,439	1,112

These results are for a non-TPC benchmark. A subset of the TPC-DS Benchmark standard requirements was implemented.

The Hadoop-DS benchmark implementation complied with the following subset of requirements from the latest version of the TPC-DS Benchmark standard.

- The database schemas were defined with the proper layout and data types
- The population for the databases was generated using the TPC provided dsdgen
- The three databases were properly scaled to 10TB and populated accordingly
- The auxiliary data structure requirements were met since none were defined
- The query input variables were generated by the TPC provided dsqgen
- The execution times for queries were correctly measured and reported

The following features and requirements from the latest version of the TPC-DS Benchmark standard were not adhered to:

- A subset of 46 queries out of the total set of 99 were executed
- The database load time was neither measured nor reported
- The defined referential integrity constraints were not enforced
- The statistics collection did not meet the required limitations
- The data persistence properties were not demonstrated
- The data maintenance functions were neither implemented nor executed
- A single throughput test was used to measure multi-user performance
- The system pricing was not provided or reviewed
- The report did not meet the defined format and content

The white paper documenting the details of the Hadoop-DS benchmark executed against the three query engines was verified for accuracy.

Respectfully Yours,

François Raab, President