Multi-Dimensional Database Modeling and Querying: Methods, Experiences and Challenging Problems

Alfredo Cuzzocrea University of Trieste & ICAR
Rim Moussa LaTICE Lab. & University of Carthage

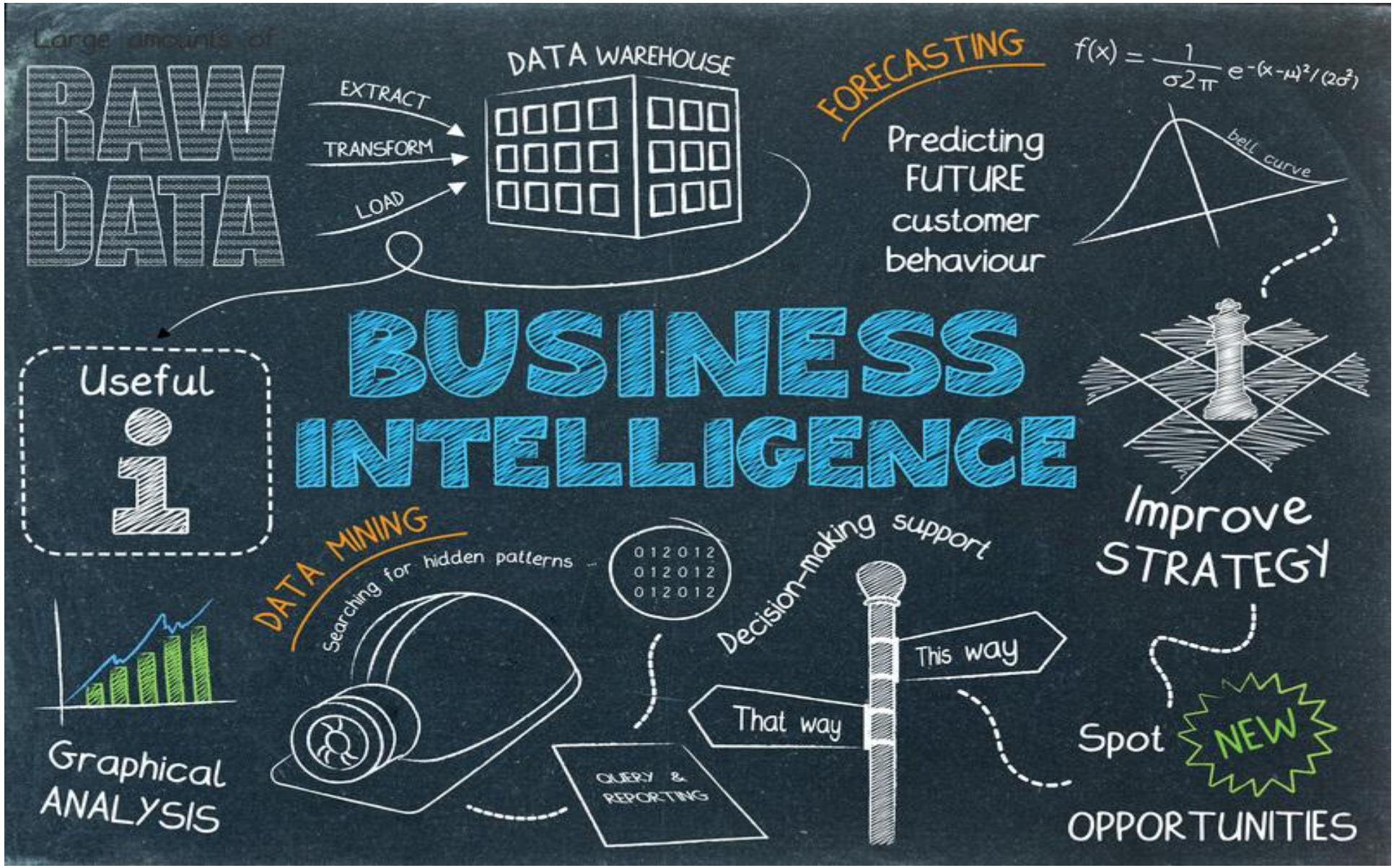
17th of November, 2016

The 35th Intl. Conference on Conceptual Modeling
@ Gifu, JAPAN

Tutorial Outline

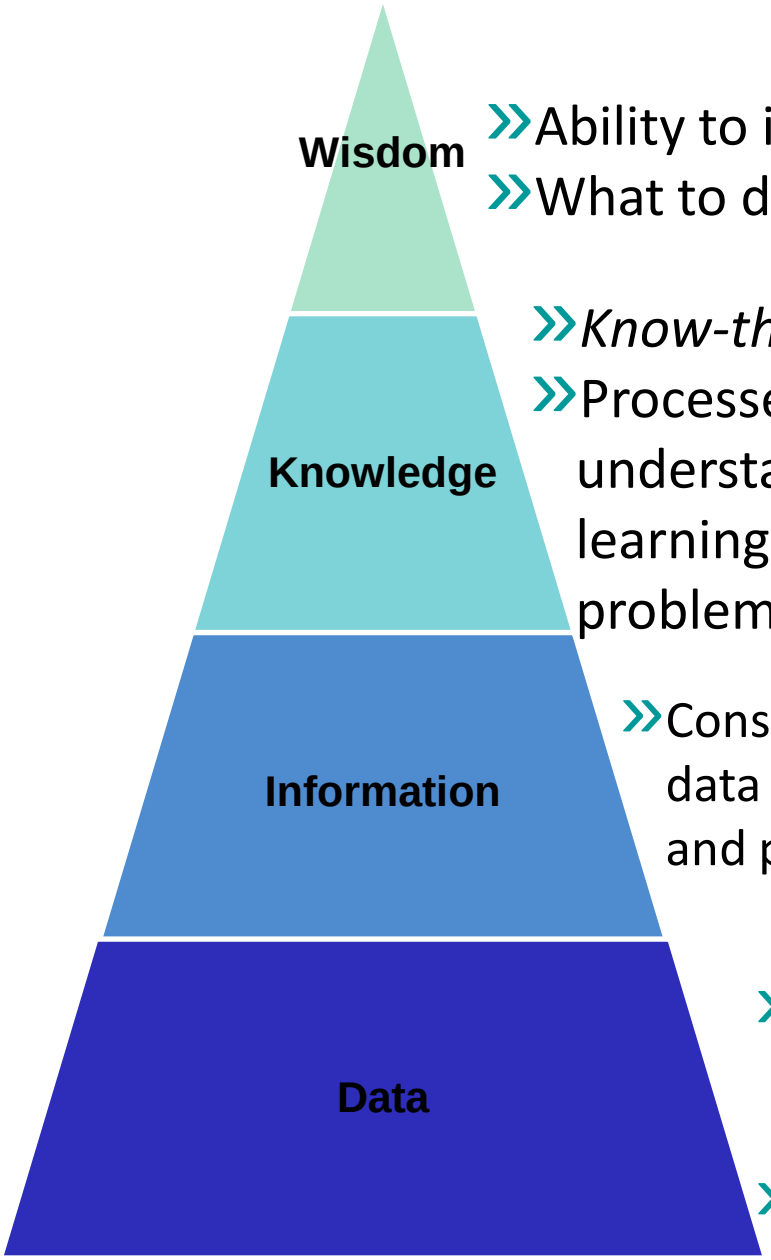
- Part I: Data Warehouse Systems
  - Decision Support Systems
  - DWS Architectures
  - DW Schemas
  - OLAP cube and OLAP operations
  - DSS Benchmarks
  - OLAP Mandate
- Part II: Multi-dimensional design
- Part III: Challenging Problems
- Conclusion

Translating Data into Insights & Opportunities!



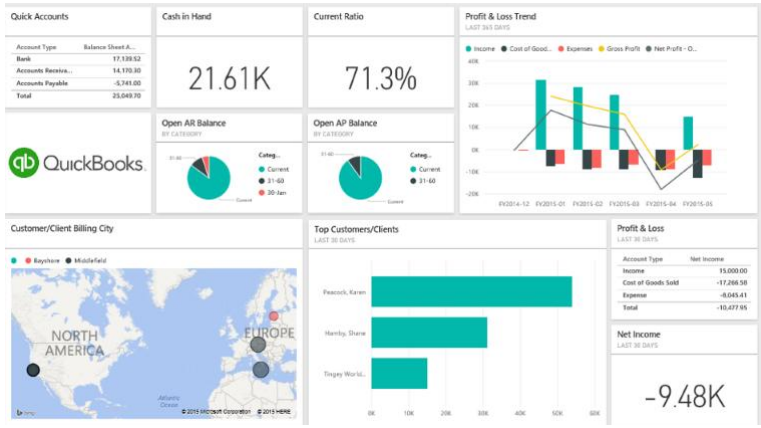
Source: datapine

From DIKW Pyramid To Decision Support System



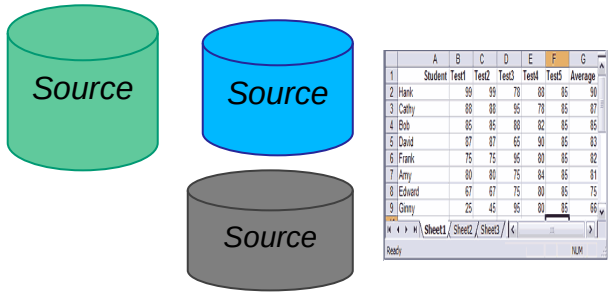
- Ability to increase effectiveness
- What to do, act …

- *Know-thats*
- Processed data that conveys understanding, experience or learning applicable to a problem|activity



- Consolidated and organized data that has meaning, value and purpose

- Descriptions of things, events, activities and transactions
- internal or external



Decision Support System

- Decision Support System
  - A collection of integrated software applications and hardware that form the backbone of an organization's decision making process.
  - Performs different types of analyses
    - What-if analysis
    - Simulation analysis
    - Goal-seeking analysis
- Business Intelligence
  - BI encompasses a variety of tools, applications and methodologies that enable organizations to collect data from internal systems and external sources, prepare it for analysis, develop and run queries against the data, and create reports, dashboards and data visualizations to make the analytical results available to corporate decision makers as well as operational workers.
  - Integrations (ETL) tools, data warehousing tools, OLAP technologies, OLAP clients, mining structures, reporting tools, dashboards' design tools
  - Integration workflows design methods**, Multi-dimensional design methods**, …

Decision Support System
OLTP vs OLAP (1/3)

- OLTP: On-Line Transaction Processing
  - Users/ applications interacting with database in real-time
  - e.g.: on-line banking, on-line payment
  - Required properties of memory access
    - ACID properties: Atomicity, Consistency, Isolation, Durability
    - Coherence
  - Benchmarks: TPC-C, TPC-E
- OLAP: On-Line Analytical Processing
  - Experts doing *offline* data analysis
  - e.g.: data warehouses, decision support systems
  - Memory patterns: Indexed and Sequential access patterns
  - Benchmarks: TPC-H, TPC-DS

Decision Support System
OLTP vs OLAP (2/3)

- OLTP: On-Line Transaction Processing
  - Users/ applications interacting with database in real-time
  - e.g.: on-line banking, on-line payment
  - Required properties of memory access
    - ACID properties: Atomicity, Consistency, Isolation, Durability
    - Coherence
  - Benchmarks: TPC-C, TPC-E
- OLAP: On-Line Analytical Processing
  - Experts doing *offline* data analysis
  - e.g.: data warehouses, decision support systems
  - Memory patterns: Indexed and Sequential access patterns
  - Benchmarks: TPC-H, TPC-DS

Decision Support System
OLTP vs OLAP (3/3)

OLTP	OLAP
Users/apps interacting with DB in real-time	Business Experts doing offline data Analysis
E.g., customers buying/selling books at Amazon	E.g., bestseller at Amazon
Many concurrent users, queries, connections	Few concurrent users, queries, connections
DB is 3rd NF	DB is denormalized
Simple queries, often predetermined	Complex ad-hoc queries
Relatively little data (~GB)	Very large data sets (~TB)
Each query touches little data	Queries touch large data sets
Little computation per query	Mining is compute intensive
Simple computation	Complex operations in mining
Data is continually updated	Data is mostly read-only
Accuracy and recovery important, hence strict transactions	Strict transactional
Throughput is most important	semantics is not needed
	Latency is more important

Data Integration

- *Data Integration* is the process of integrating data from multiple sources and probably have a single view over all these sources
  - Integration can be physical (copy the data to warehouse) or virtual (keep the data at the sources)
  - *Data Acquisition*: This is the process of moving company data from the source systems into the warehouse. Data acquisition is an ongoing, scheduled process, which is executed to keep the warehouse current to a pre-determined period in time.
  - *Changed Data Capture*: The periodic update of the warehouse from the transactional system(s) is complicated by the difficulty of identifying which records in the source have changed since the last update.
- *Data Cleansing*: This is typically performed in conjunction with data acquisition. It is a complicated process that validates and, if necessary, corrects the data before it is inserted into the warehouse

Data Integration
Software solutions

- *Software* Solutions
  - Known as ETL tools, Integration services, ...
  - Vendors (non-exhaustive list): IBM (InfoSphere Data Event Publisher), Informatica (Informatica Enterprise Data Integration), Information Builders (iWay DataMigrator), Microsoft (Microsoft's SQL Server Integration Services), Oracle (Oracle GoldenGate), Pentaho (Pentaho DI - prev. Ketlle), SAP (BusinessObjects Data Integrator), SAS (SAS DataFlux) and Talend (Talend Open Studio),
    - Kettle (Pentaho) and Talend offer open-source versions,
    - MicroSoft bundels integration services with database products

Integrating Heterogeneous data sources: Lazy or Eager?

- *Lazy Integration* aka *Query-driven approach*
  - Accept a query, determine the sources which answer the query, devise the execution tree: generate appropriate sub-queries for each source, obtain resultsets, perform required post-processing: translation, merging and filtering and return answer to application
- *Eager Integration* aka *Warehouse approach*
  - Information of each source of interest is extracted in-advance, translated, filtered and processed as appropriate, then merged with information from other sources and stored
  - Accept a query, answer query from repository

Data Integration
Eager vs. Lazy ?

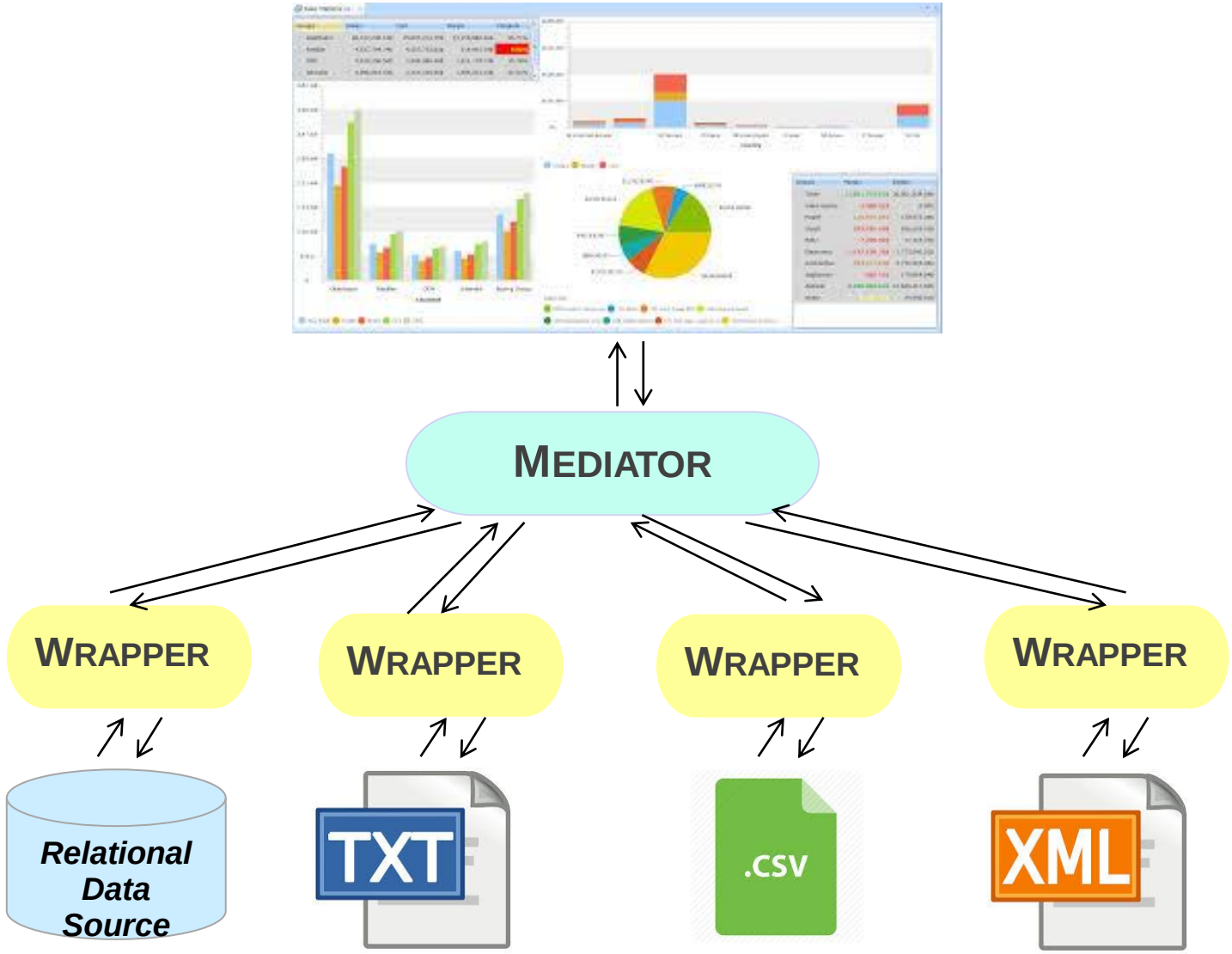
Eager

- Integrate in-advance i.e. before queries
- Copy data from sources
- Query answer-set could be stale
- Need to refresh the data warehouse
- Operates when sources are unavailable
- High query performance: through building data summaries and local processing at sources unaffected
- OLAP queries are not visible outside the warehouse

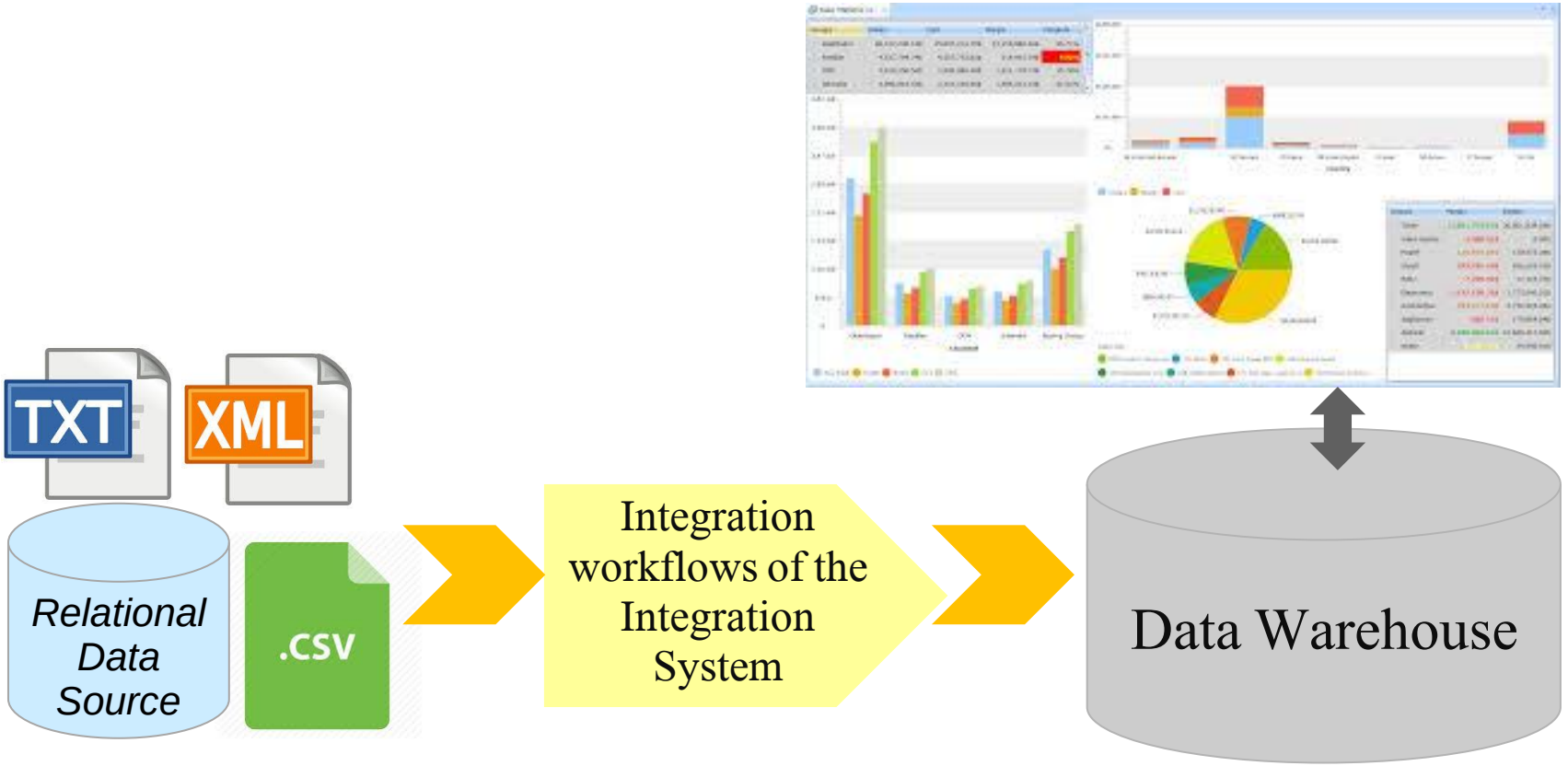
Lazy

- Integrate on-demand i.e. at query time
- Leave data at sources
- Query answer-set is up-to-date
- No copy so no refresh and storage cost
- Out of service if sources unavailable
- Sources are drained: interference with local processing

Lazy Data Integration
→ Query-driven Architecture



Eager Data Integration
→ Warehouse System Architecture



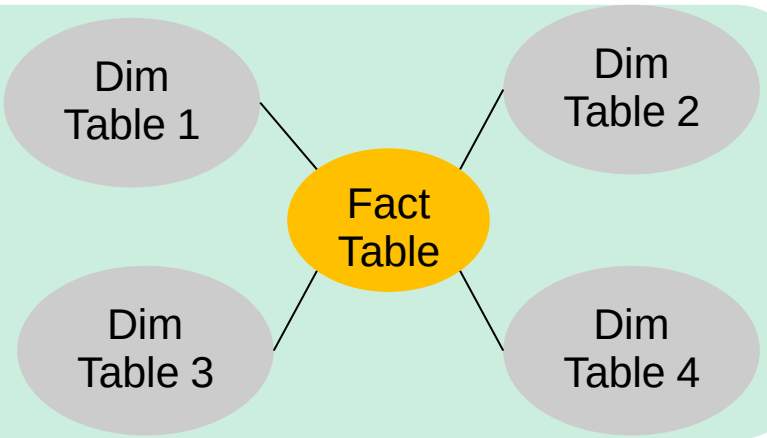
Data Warehouse Definition

- **By Bill Inmon** "*A warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process*"
  - *Subject-oriented*: The data in the database is organized so that all the data elements relating to the same real-world event or object are linked together; e.g. finance, human resources, sales …
  - *Time-variant*: The changes to the data in the database are tracked and recorded so that reports can be produced showing changes over time. Hence, all addresses of a customer are in the DW,
  - *Non-volatile*: Data in the database is never over-written or deleted - once committed, the data is static, read-only, but retained for future reporting;
  - *Integrated*: Data that is gathered into the data warehouse from a variety of sources and merged into a coherent whole. There is only a single way to identify a product
- **By Ralph Kimbal** *"A copy of transaction data specifically structured for query and analysis"*
- Inmon top-down approach vs. Kimbal bottom-up approach

Data Warehouse Schemas

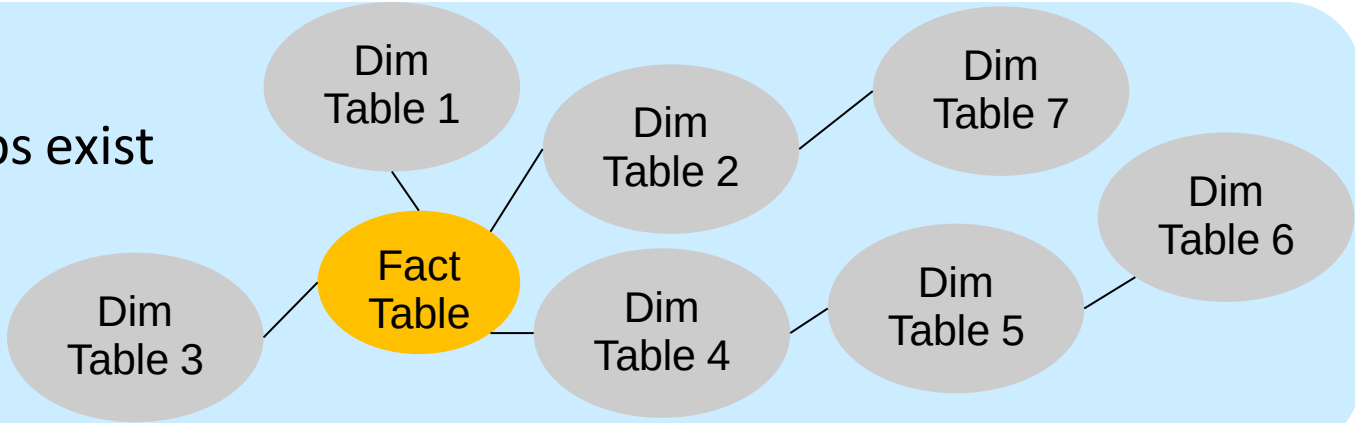
Star Schema

- A single, large and central table *Fact table* surrounded by multiple *Dimension tables*
- All Dimensions Tables have a hierarchical relationship with the Fact Table



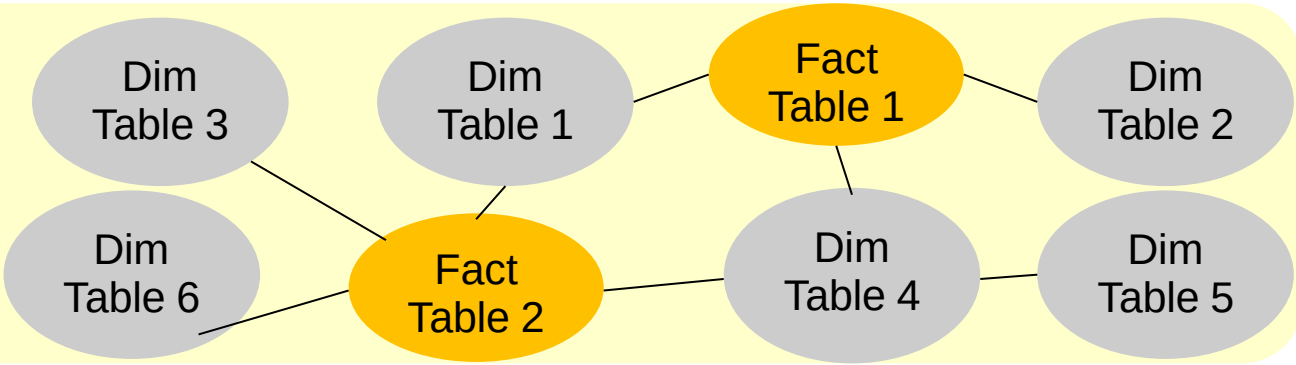
Snowflake Schema

- Hierarchical relationships exist among Dimensions Tables
- Normalized schema



Constellation Schema

- Galaxy schema
- Multiple fact tables which share dimensions



Multi-dimensional Data Analysis

- Multi-dimensional Data Analysis
  - refers to the process of summarizing data across multiple levels (called dimensions) and then presenting the results in a multi-dimensional grid format.
- OLAP Cube
  - Multi-dimensional representation of data
  - aka Crosstab, Data Pivot

OLAP Cube

- *Facts*: are the objects that represent the subject of the desired analyses.
  - Examples: sales records, weather records, cabs trips, …
  - The *fact table* contained 3 types of attributes: measured attributes, foreign keys to dimension tables, degenerate dimensions
- *Dimension(s)*:
  - *Levels* are individual values that make up dimensions
  - Examples
    - Date dimension (Trimester, month, day)
    - Time dimension (hour, min, sec)
    - Geography dimension (Country, city, postal code)
- *Measure(s)*:
  - Examples: revenue, lost revenue, sold quantities, expenses, forecasts, …
  - Use aggregate functions: min, max, count, distinct-count, sum, average, …

OLAP Operations
On-Line Analytical Processing

	Order Date			
Customer	1992	1993	1994	1995
ALGERIA	5 264 050,753	7 373 514,635	7 430 767,906	2 018 405,097
ARGENTINA	4 210 839,882	4 542 819,432	5 167 028,23	1 010 599,268
BRAZIL	6 747 872,866	6 574 905,869	8 524 884,48	1 207 012,031
CANADA	7 212 602,708	6 106 296,311	6 503 400,31	1 509 823,023
CHINA	4 521 669,942	4 848 721,846	3 674 702,48	645 263,958
EGYPT	6 538 746,72	7 556 396,212	8 226 937,087	953 629,15
ETHIOPIA	6 670 268,851	4 659 844,423	5 087 899,524	2 442 708,868
FRANCE	3 992 266,571	3 768 984,		0,616
GERMANY	6 554 803,872	6 451 164,		2,082
INDIA	5 553 473,099	5 202 243,		5,091
INDONESIA	6 108 221,198	5 246		106
IRAN	6 205 647,298	7 269		16
IRAQ	6 874 081,811	4 905		99
JAPAN	6 920 507,077	5 109		41
JORDAN	4 412 586,717	5 690		83
KENYA	5 089 554,29	4 087		59
MOROCCO	4 852 853,105	4 316		4,9
MOZAMBIQUE	4 908 755,902	6 228 931,225	6 037 134,828	1 472 150,692
PERU	4 837 477,525	4 015 941,591	4 801 642,271	918 718,471
ROMANIA	6 321 412,895	6 502 839,12	6 260 917,372	975 649,846
RUSSIA	4 540 220,505	5 654 561,97	4 865 228,695	946 701,784
SAUDI ARABIA	6 060 723,834	8 062 651,623	6 368 801,965	1 063 828,728
UNITED KINGDOM	5 973 693,6	5 826 140,148	6 365 734,513	1 509 592,537
UNITED STATES	3 597 918,308	4 674 529,097	2 875 807,861	966 513,254
VIETNAM	5 855 595,978	5 127 222,461	6 932 101,314	908 495,999

Pivot Table

- Lost Revenue per customer's region per order date (year-quarter)

- Lost Revenue during 1st quarter of 1992 of French customers
  - Show customers' details: Customer key, name, account balance, phone number

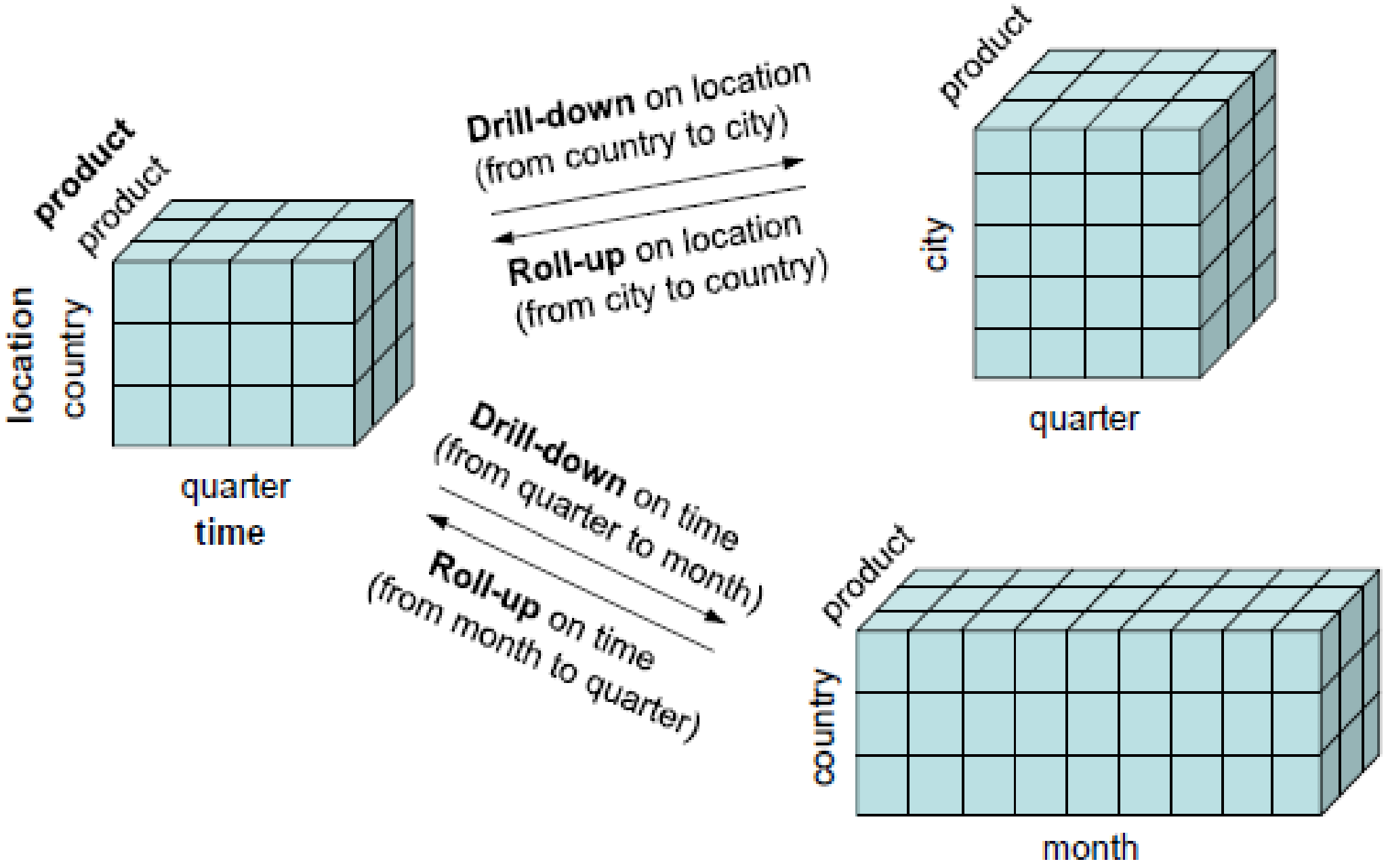
User Query

				Order Date	
Customer	Name	Acct Bal	Phone	1992	1
FRANCE				3 992 266,571	1 485 456,214
50	Customer#000000050	4266.13	16-658-112-3221	205 325,74	
271	Customer#000000271	1490.35	16-621-282-5685	193 585,758	92 935,691
281	Customer#000000281	4361.7	16-809-382-6446	139 235,302	88 494,328
284	Customer#000000284	593.52	16-161-235-2690	24 802,628	
319	Customer#000000319	1834.36	16-734-928-1642	428 505,573	101 989,447
413	Customer#000000413	5817.9	16-158-285-7336	215 399,988	
431	Customer#000000431	2273.5	16-326-904-6643	5 753,705	5 753,705
455	Customer#000000455	6860.34	16-863-225-9454	359 198,382	296 286,348
587	Customer#000000587	7077.79	16-585-233-5906	105 120,133	
667	Customer#000000667	3288.70	16-917-453-2490	346 230,65	

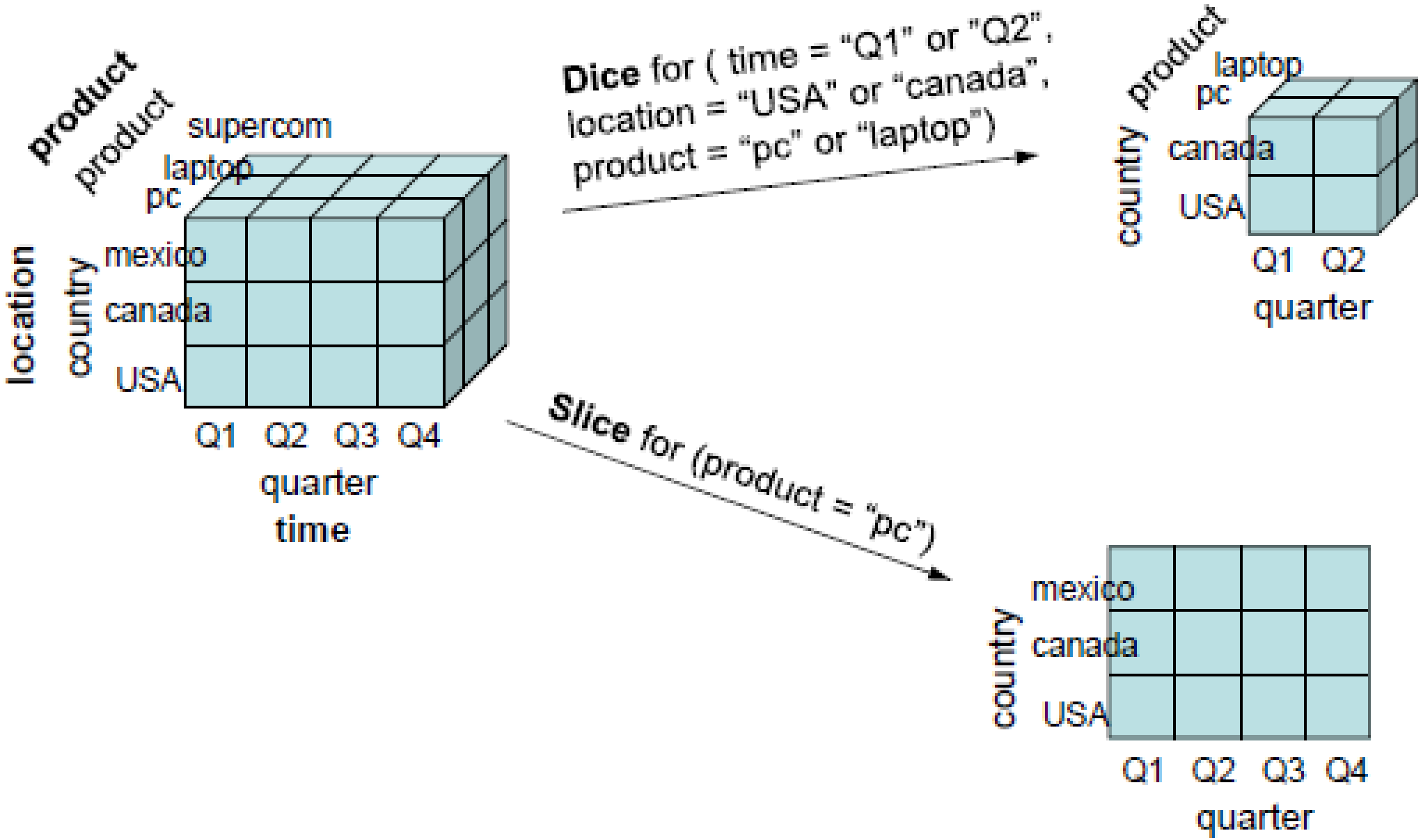
OLAP Operations

- Drill-down
  - stepping down to lower level data or introducing new dimensions
- Roll-up
  - summarizes data by climbing up hierarchy or by dimension reduction
- Pivot
  - rotate, reorienting the cube
- Slice
  - selecting data on one dimension
- Dice
  - selecting data on multiple dimensions

OLAP Operations
Roll-up & Drill-down

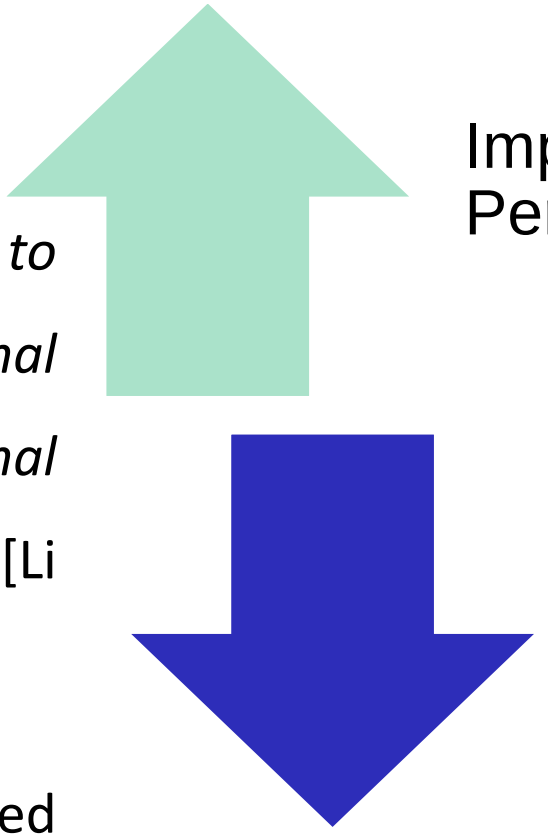


OLAP Operations
Slice & Dice



Derived Data

- OLAP Indices
  - Inverted lists
  - Bitmap Indices
  - Join indices
  - Text indices
- *Materialization of a data cube is a way to pre-compute and store multi-dimensional aggregates so that multi-dimensional analysis can be performed on-the-fly* [Li et al., 2004]
  - Calculated Attributes
  - Aggregate Tables (aka materialized tables)
  - Data synopsis: histograms and sketches

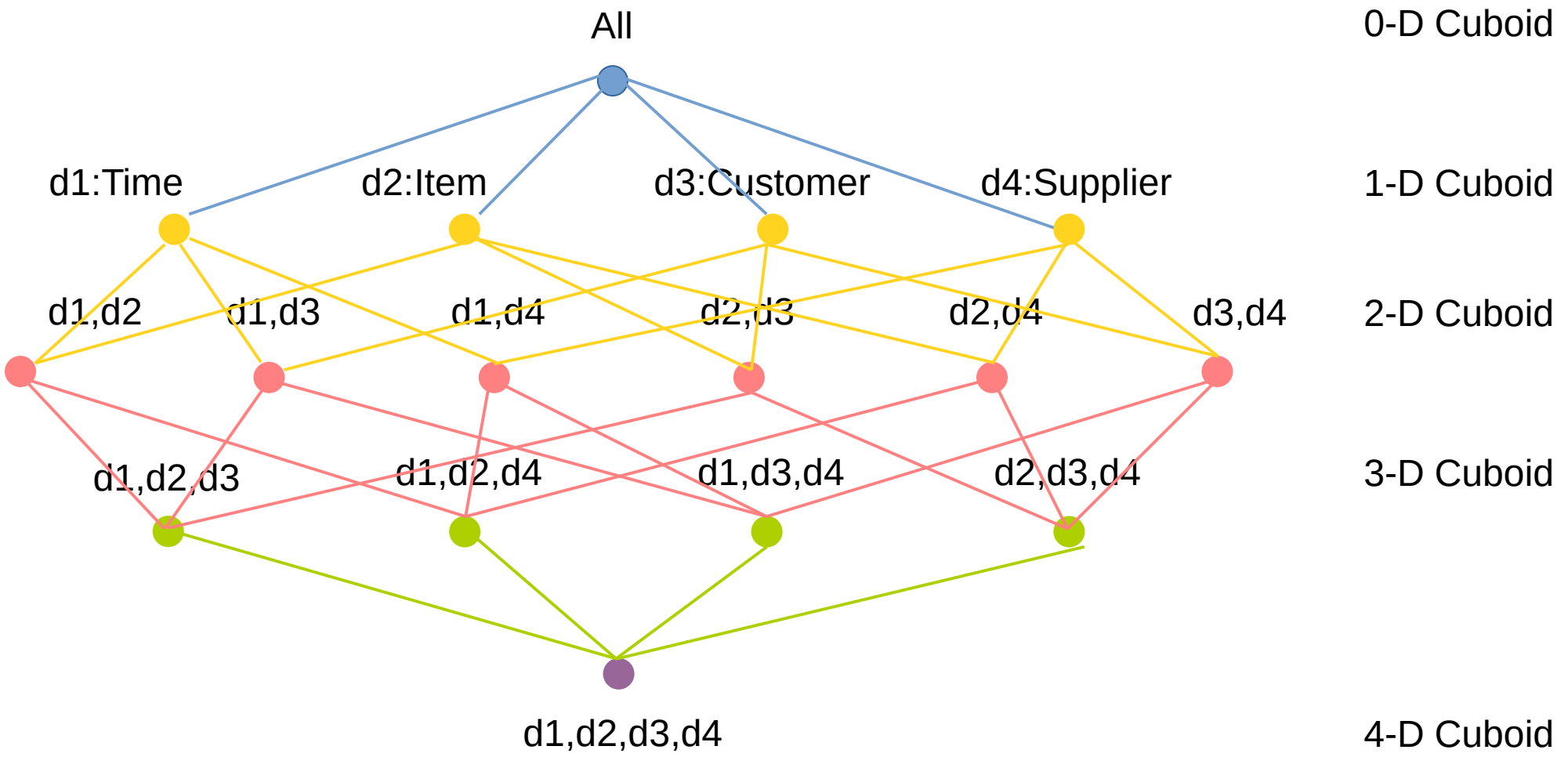


Improved Query Performance

Maintenance & Refresh

Storage Requirements

Cuboids Lattice
What to materialize?



- For n-dimensional cube
  - » 2^n cuboids

OLAP Servers

- ROLAP Server
  - Relational OLAP
  - Uses relational or extended-relational DBMS to store and manage warehouse data and OLAP middleware
  - Includes optimization of DBMS back-end, implementation of aggregation navigation logic, and additional tools and services
  - High scalability
  - E.g. Mondrian (Pentaho BI suite)
- MOLAP Server
  - Multidimensional OLAP
  - Sparse array-based multidimensional storage engine
  - Fast indexing to pre-computed summarized data
  - E.g. Palo
- HOLAP Server
  - Low-level: relational / high-level: array
  - High flexibility
  - E.g. MSAS (Microsoft)

Query Languages

- Structured Query Language (SQL)
  - Relational and static schema
  - Data Definition, data Manipulation, and Data Control Language
  - Analytic Functions (window functions `over partition by` …)
  - `Cube, roll-up and grouping sets operators`
- *MultiDimensional eXpressions (*MDX)
  - Invented by Microsoft in 1997
  - For querying and manipulating the multidimensional data stored OLAP cubes
  - Static schema
- Data Flow programming language
  - Google Sawzall, Apache Pig Latin, IBM Infosphere Streams
  - Dynamic schema
  - After data is loaded, multiple operators are applied on that data before the final output is stored.

Query Languages
SQL – Q16 of TPC-H benchmark

<Q16> The *Parts/Supplier Relationship Query* counts the number of suppliers who can supply parts that satisfy a particular customer's requirements. The customer is interested in parts of eight different sizes as long as they are not of a given type, not of a given brand, and not from a supplier who has had complaints registered at the Better Business Bureau.

```
SELECT p_brand, p_type, p_size, count(distinct ps_suppkey) as
supplier_cnt
FROM partsupp, part
WHERE  p_partkey = ps_partkey
AND p_brand <> '[BRAND]'
AND p_type NOT LIKE '[TYPE]%'
AND p_size in ([SIZE1], [SIZE2], [SIZE3], [SIZE4], [SIZE5],
[SIZE6], [SIZE7], [SIZE8])
AND  ps_suppkey NOT IN (SELECT s_suppkey FROM   supplier
            WHERE s_comment like '%Customer%Complaints%')
GROUP BY p_brand, p_type, p_size
ORDER BY  supplier_cnt DESC, p_brand, p_type, p_size;
```

Query Languages
MDX –Q16 of TPC-H Benchmark

```
WITH SET [Brands] AS  'Except({[Part Brand].Members}, {[Part
Brand].[Brand#45  ]})'
  SET [Types] AS 'Filter({[Part Type].Members}, (NOT ([Part
Type].CurrentMember.Name MATCHES "(?i)MEDIUM POLISHED.*")))'
   SET [Sizes] AS 'Filter({[Part Size].Members}, ([Part Size].CurrentMember IN
{[Part Size].[3], [Part Size].[9], [Part Size].[14], [Part Size].[19], [Part Size].[23],
[Part Size].[36], [Part Size].[45], [Part Size].[49]}))'
SELECT [Measures].[Supplier Count] ON COLUMNS,
  nonemptyCrossjoin(nonemptyCrossjoin([Brands], [Types]), [Sizes]) ON ROWS
FROM [Cube16]
```

Query Languages
Data Flow –Pig Latin script for Q16 of TPC-H benchmark

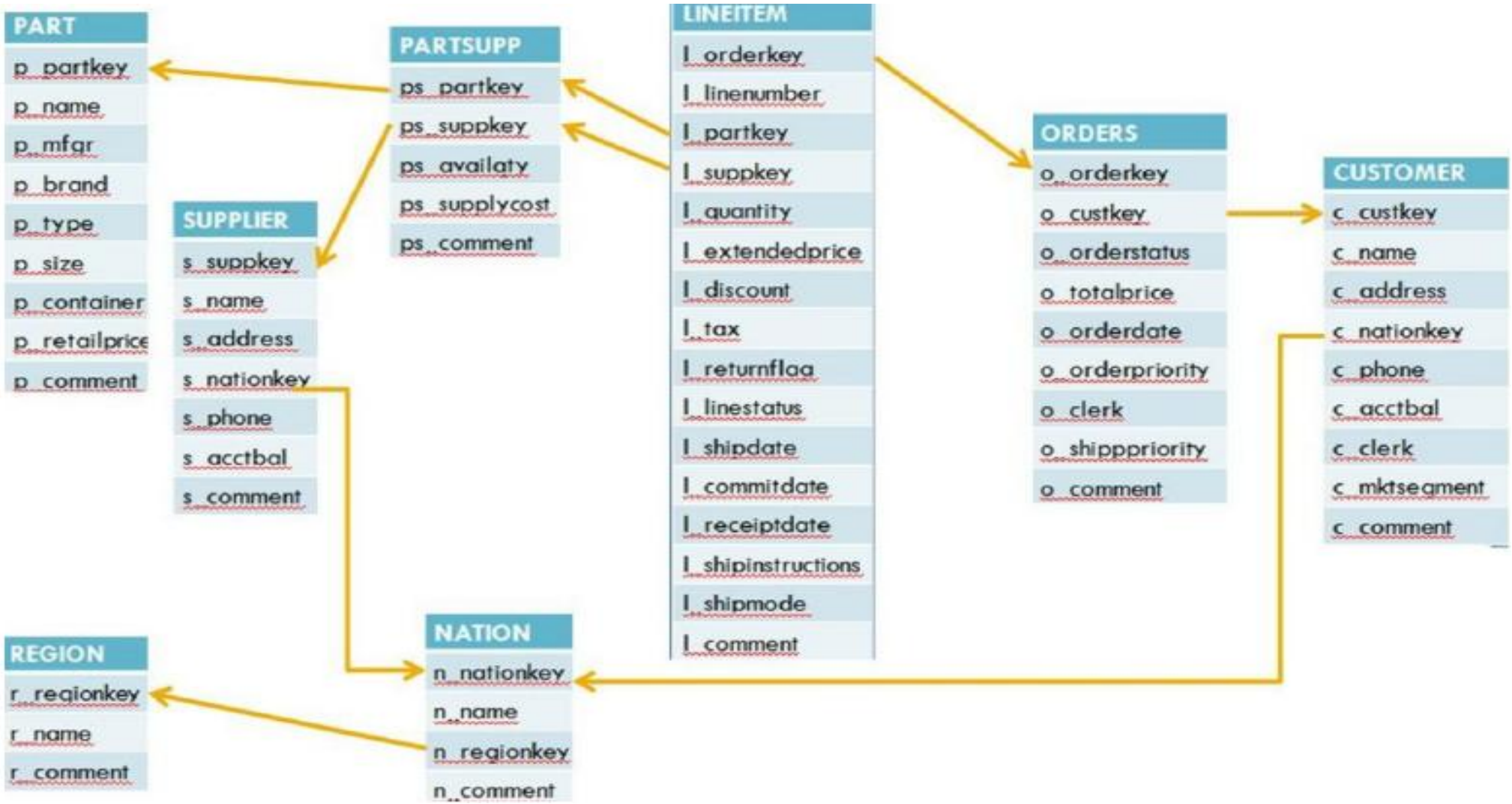
```
--- Suppliers with no complaints
supplier = LOAD 'TPCH/supplier.tbl' USING PigStorage('|') AS
(s_suppkey:int, s_name:chararray, s_address:chararray, s_nationkey:int,
s_phone:chararray, s_acctbal:double, s_comment:chararray);
supplier_pb= FILTER supplier BY NOT(s_comment matches
'.*Customer.*Complaints.*');
suppkeys_pb = FOREACH supplier_pb GENERATE s_suppkey;
--- Parts size in 49, 14, 23, 45, 19, 3, 36, 9
part = LOAD 'TPCH/part.tbl' USING PigStorage('|') AS (...);
parts = FILTER part BY (p_brand != 'Brand#45') AND  NOT (p_type matches
'MEDIUM POLISHED.*') AND (p_size IN (49, 14, 23, 45, 19, 3, 36 , 9);
---Join partsupp, selected parts, selected suppliers
partsupp = LOAD 'TPCH/partsupp.tbl' using PigStorage('|') AS (...);
part_partsupp = JOIN partsupp BY ps_partkey, parts BY p_partkey;
not_pb_supp = JOIN part_partsupp BY ps_suppkey, suppkeys_pb BY
s_suppkey;
selected = FOREACH not_pb_supp GENERATE ps_suppkey, p_brand, p_type,
p_size;
grouped = GROUP selected BY (p_brand,p_type,p_size);
count_supp = FOREEACH grouped GENERATE flatten(group),
COUNT(selected.ps_suppkey) as supplier_cnt;
result = ORDER count_supp BY supplier_cnt DESC, p_brand, p_type, p_size;
STORE result INTO 'OUTPUT_PATH/tpch_query16';
```

Decision Support Systems Benchmarks

- Non-TPC Benchmarks
  - Real datasets
    - Open data or proprietary data
    - fixed size
    - Devise a workload or trace the proprietary workload
  - APB-1: no scale factor
- TPC Benchmarks
  - The *Transaction Processing Council* founded in 1988 to define benchmarks
    - In 2009, *TPC-TC* is set up as an International Technology Conference Series on Performance Evaluation and Benchmarking
  - Examples of benchmarks relevant for benchmarking decision support systems: TPC-H, TPC-DS and TPC-DI
  - Common characteristics of TPC benchmarks
    - Synthetic data
    - Scale factor allowing generation of different volumes 1GB to 1PB

Decision Support Systems Benchmarks
TPC-H Benchmark Schema (1/2)

- TPC-H Benchmark: complex schema
  22 ad-hoc SQL statements (star queries, nested queries, …) + refresh functions

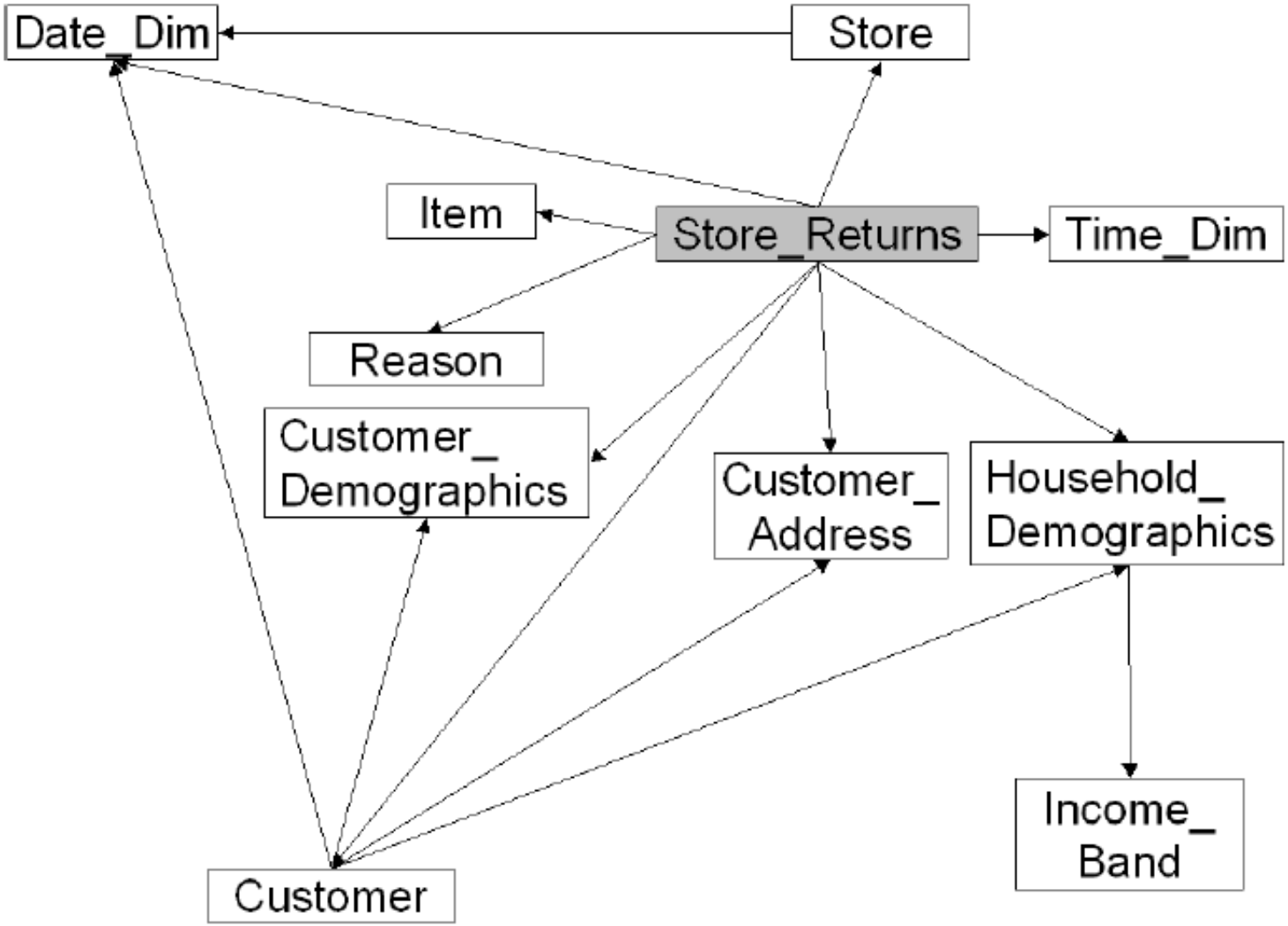


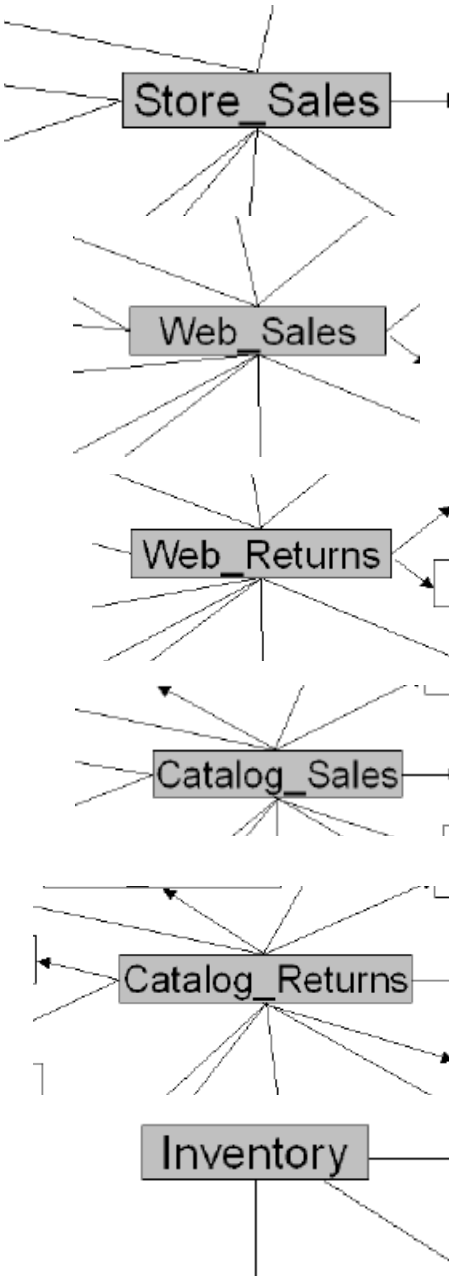
Decision Support Systems Benchmarks
TPC-H Benchmark (2/2)

- TPC-H Benchmark Metrics
  - 2 Metrics
    - *QphH@Size* is the number of queries processed per hour, that the system under test can handle for a fixed load
    - *$/QphH@Size* represents the ratio of cost to performance, where the cost is the cost of ownership of the SUT (hardware,software, maintenance).
- Variants of TPC-H Benchmarks
  - *TPC-H*d Benchmark* –detailed in Part II
    - Turning TPC-H benchmark into a Multi-dimensional benchmark
    - Few schema changes
    - No update to workload requirement
    - MDX workload for OLAP cubes and OLAP queries
  - *SSB: Star Schema Benchmark*
    - Turning TPC-H benchmark into star-schema
    - Workload composed of 12 queries
  - *TPC-H translated into Pig Latin (Apache Hadoop Ecosystem)*
    - 22 pig latin scripts which load and process TPC-H raw data files (.tbl files)

Decision Support Systems Benchmarks
TPC-DS Benchmark (1/2)

- TPC-DS Benchmark: 7 data marts





Decision Support Systems Benchmarks
TPC-DS Benchmark (2/2)

- TPC-DS Benchmark Workload
  - Hundred of queries (99 query templates)
  - OLAP, windowing functions, mining, and reporting queries
  - Concurrent data maintenance
- TPC-DS Benchmark Metrics
  - 3 Metrics
    - *QphDS@Size* is the number of queries processed per hour, that the system under test can handle for a fixed load.
      - Data Maintenance and Load Time are calculated
    - *$/QphDS@Size* represents the ratio of cost to performance, where the cost is a 3 year cost of ownership of the SUT (hardware,software, maintenance)
    - System Availability date: the date when the system is available to customers.
- TPC-DS implementations
  - TPC-DS v2.0
    - Extension for non-relational systems such as Hadoop/Spark big data systems

Decision Support Systems Benchmarks
TPC-DI Benchmark (1/3)

- For benchmarking Data Integration technologies
- Synthetic Data of a Factious Retail Brokerage Firm
  - Internal Trading system data, Internal Human resources data, Internal CRM System and External data
  - Different data scales
  - Data extracted from different sources:
    - Structured (csv)
    - Semi-structured data (xml)
    - Multi record
    - Change Data Capture (CDS)
- Complex Data Integration Tasks
  - Load large volumes of historical data
  - Load incremental updates
  - Execute complex transformations
  - Check and ensure consistency of data

OLAP Mandate
12 Rules for Evaluating OLAP products by E.F. Codd et al.

- Rule#1: Multidimensional Conceptual View
  - The multidimensional conceptual view facilitates OLAP model design and analysis, as well as inter and intra dimensional calculations.
- Rule#2: Transparency
  - Whether OLAP is or is not part of the user's customary front-end product, that fact should be transparent to the user. If OLAP is provided within the context of a client server architecture, then this fact should be transparent to the user-analyst as well.
- Rule#3: Accessibility
  - The OLAP tool must map its own logical schema to heterogeneous physical data stores, access the data, and perform any conversions necessary to present a single, coherent and consistent user view. Moreover, the tool and not the end-user analyst must be concerned about where or from which type of systems the physical data is actually coming

OLAP Mandate
12 Rules for Evaluating OLAP products by E.F. Codd et al.

- Rule#4: Consistent Reporting Performance
  - As the number of dimensions or the size of the database increases, the OLAP user-analyst should not perceive any significant degradation in reporting performance.
- Rule#5: Client-Server Architecture
  - Most data currently requiring on-line analytical processing is stored on mainframe systems and accessed via personal computers. It is imperative that the server component of OLAP tools be sufficiently intelligent such that various clients can be attached with minimum effort and integration programming.
- Rule#6: Generic Dimensionality
  - Every data dimension must be equivalent in both its structure and operational capabilities. Dimensions are symmetric, So the basic data structure, formulae, and reporting formats should not be biased toward any one data dimension.

- Rule#7: Dynamic Sparse Matrix Handling
  - The OLAP tools' physical schema must adapt fully to the specific analytical model being created to provide optimal sparse matrix handling.
  - By adapting its physical data schema to the specific analytical model, OLAP tools can empower user analysts to easily perform types of analysis which previously have been avoided because of their perceived complexity.
- Rule#8: Multi-User Support
  - OLAP tools must provide concurrent access (retrieval and update), integrity, and security.
- Rule#9: Unrestricted Cross-dimensional Operations
  - The various roll-up levels within consolidation paths, due to their inherent hierarchical nature, represent in outline form, the majority of 1:1, 1:M, and dependent relationships in an OLAP model or application. Accordingly, the tool itself should infer the associated calculations and not require the user-analyst to explicitly define these inherent calculations.

- Rule#10: Intuitive Data Manipulation
  - » Consolidation path re-orientation, drilling down across columns or rows, zooming out, and other manipulation inherent in the consolidation path outlines should be accomplished via direct action upon the cells of the analytical model, and should neither require the use of a menu nor multiple trips across the user interface.
- Rule#11: Flexible Reporting
  - » Reporting must be capable of presenting data to be synthesized, or information resulting from animation of the data model according to any possible orientation. This means that the rows, columns, or page headings must each be capable of containing/displaying from 0 to N dimensions each, where N is the number of dimensions in the entire analytical model.
- Rule#12: Unlimited Dimensions and Aggregation Levels
  - » An OLAP tool should be able to accommodate at least fifteen and preferably twenty data dimensions within a common analytical model.

References

- M. Fricke, *The Knowledge Pyramid: A Critique of the DIKW Hierarchy.* Journal of Information Science. 2009.
- E.F. Codd, S.B. Codd and C.T. Salley, *Providing OLAP to User Analysts: an IT mandate*, 1993.
- J. Widom, *Integrating Heterogeneous databases: eager or lazy?* ACM Computing Surveys (CSUR) Vol.4, 1996
- Y.R. Cho, *Data Warehouse and OLAP Operations* www.ecs.baylor.edu/faculty/cho/4352
- TPC homepage http://www.tpc.org/
- M. Poess, T. Rabl and B. Caufield: *TPC-DI: The First Industry Benchmark for Data Integration*. PVLDB 7(13): 1367-1378 (2014) http://www.vldb.org/pvldb/vol7/p1367-poess.pdf
- X. Li, J. Han, H. Gonzalez: *High-Dimensional OLAP: A Minimal Cubing Approach*. VLDB 2004.
- C. Imhoff, N. Galemmo, J. G. Geiger. *Mastering Data Warehouse Design: Relational and Dimensional Techniques*. 2003.
- R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, B. Becker. *The Data Warehouse Lifecycle Toolkit.* 2nd Edition.
- R. Kimball, M. Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. 2nd Edition.
- H. G. Molia. *Data Warehousing Overview: Issues, Terminology, Products.* www.cs.uh.edu/~ceick/6340/dw-olap.ppt (slides)