SQL for Hadoop:
Introducing Big SQL for BigInsights

C. M. Saracco, IBM Silicon Valley Lab (saracco@us.ibm.com)
March 13, 2014

Executive Summary

- **Why SQL?**
  - Easy on-ramp to Hadoop for SQL professionals
  - Support familiar SQL tools / applications (via JDBC and ODBC drivers)

- **What SQL operations are supported?**
  - Create tables / views (and, optionally, HBase indexes)
  - Load data into tables (from local files, distributed files, RDBMSs)
  - Query data (project, restrict, join, union, sub-queries . . . .)

- **What Hadoop-based storage mechanisms are supported?**
  - Hive
  - HBase
  - Distributed file system

Agenda

- **Big SQL: motivation and architecture**
- **Using Big SQL**
  - Invocation options
  - Creating tables
  - Populating tables with data
  - Querying data
  - Developing applications and working with tools
  - . . . And a peek at some additional topics
- **What RDBMS professionals should know about Big SQL**

Agenda

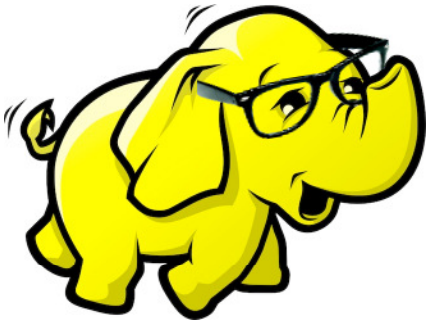
- **Big SQL: motivation and architecture** 

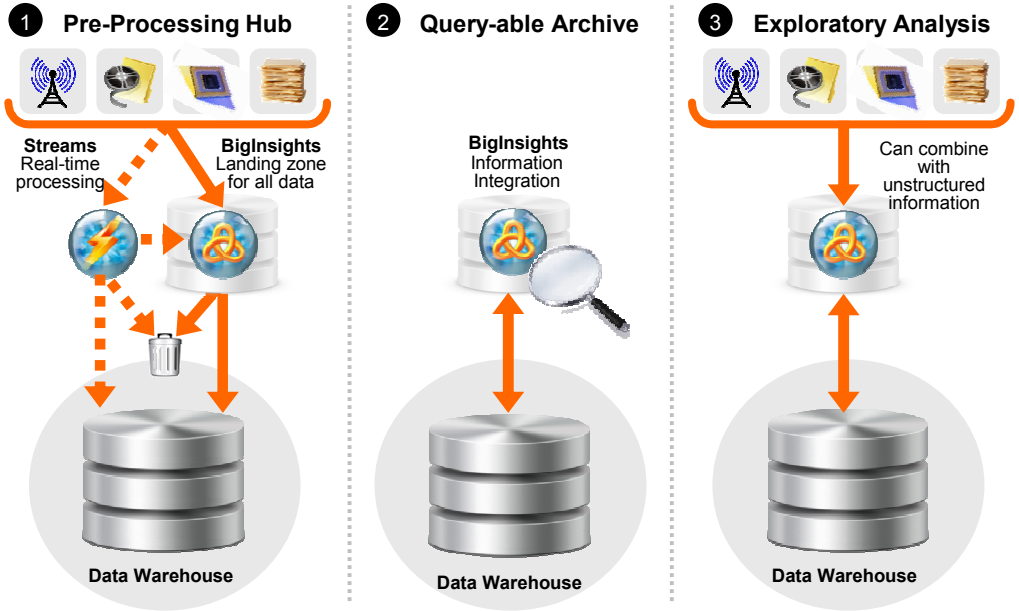
- **Using Big SQL**
  - Invocation options
  - Creating tables
  - Populating tables with data
  - Querying data
  - Developing applications and working with tools
  - . . . And a peek at some additional topics

- **What RDBMS professionals should know about Big SQL**

SQL Access for Hadoop: Why?

- **Data warehouse augmentation is a leading Hadoop use case**



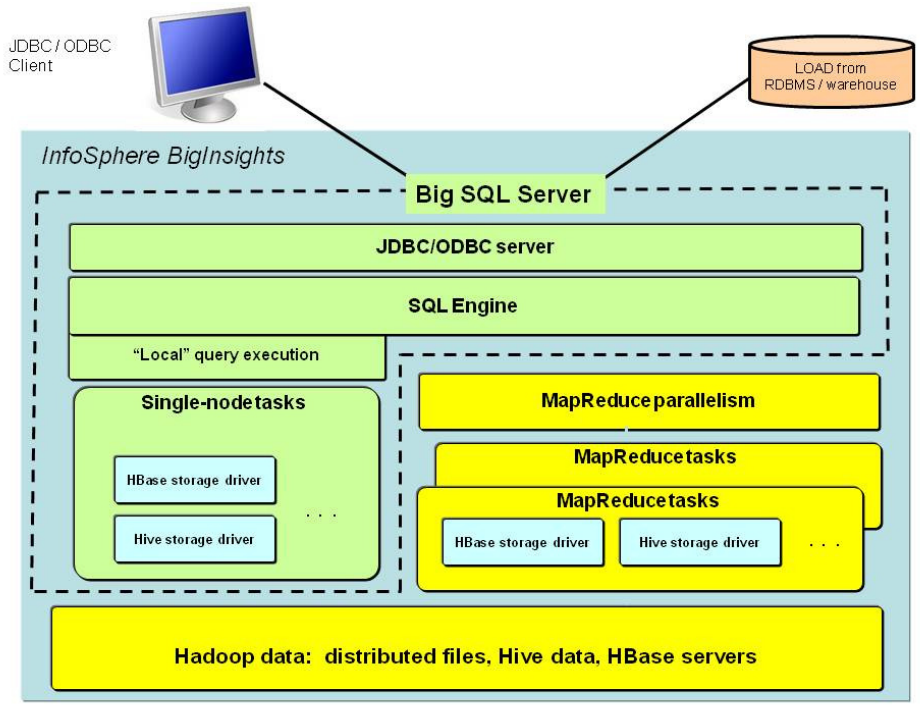


- **Hadoop often perceived as difficult**
  - MapReduce Java API requires programming expertise
  - Unfamiliar languages (such as Pig) also require special skills

- **SQL support opens the data to a much wider audience**
  - Familiar, widely known syntax
  - Common catalog for identifying data and structure

Big SQL Architecture and Feature Overview

- **Standard SQL syntax and data types**
  - Joins, unions, aggregates . . .
  - VARCHAR, decimal, TIMESTAMP, . . .
- **JDBC/ODBC drivers**
  - Prepared statements
  - Cancel support
  - Database metadata API support
  - Secure socket connections (SSL)
- **Optimization**
  - MapReduce parallelism
    or…
  - "Local" access for low-latency queries
- **Varied storage mechanisms appropriate for Hadoop ecosystem**
- **Integration**
  - Eclipse tools
  - DB2, Netezza, Teradata, Oracle*, MS-SQL*, Informix *(via LOAD)
  - Cognos Business Intelligence
  - , , ,



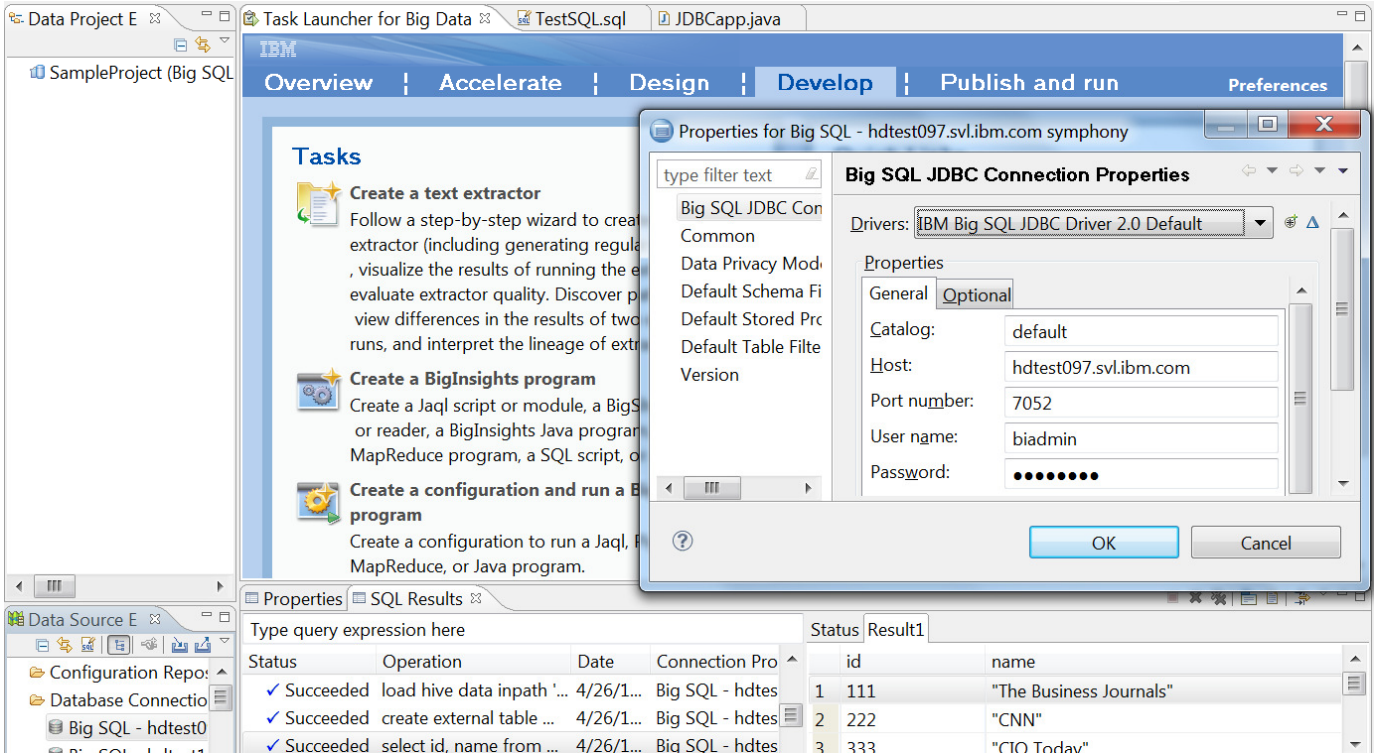
* In beta

Agenda

- **Big SQL: motivation and architecture**
- **Using Big SQL** 
  - Invocation options
  - Creating tables
  - Populating tables with data
  - Querying data
  - Developing applications and working with tools
  - . . . And a peek at some additional topics
- **What RDBMS professionals should know about Big SQL**

Invocation options provided with BigInsights

- **Command-line interface (JSqsh shell)**
- **Web-based interface (BigInsights web console)**
- **Eclipse (BigInsights plug-in)**



Creating a Big SQL Table

- **BigSQL supports CREATE TABLE and many data types including varchar, decimals, etc. Non-ISO standard clauses leverage Hadoop ecosystem**

```
CREATE TABLE TPCH.CUSTOMER ( C_CUSTKEY INTEGER, C_NAME VARCHAR(25),
C_ADDRESS VARCHAR(40), C_NATIONKEY INTEGER, C_PHONE CHAR(15), C_ACCTBAL
FLOAT, C_MKTSEGMENT CHAR(10), C_COMMENT VARCHAR(117) )
row format delimited fields terminated by '|'
stored as textfile;
```

- **Big SQL supports CREATE VIEW***

```
CREATE VIEW IF NOT EXISTS myschema.cust_view (key, name)
 AS SELECT c_custkey, c_name
      FROM TPCH.CUSTOMER;
```

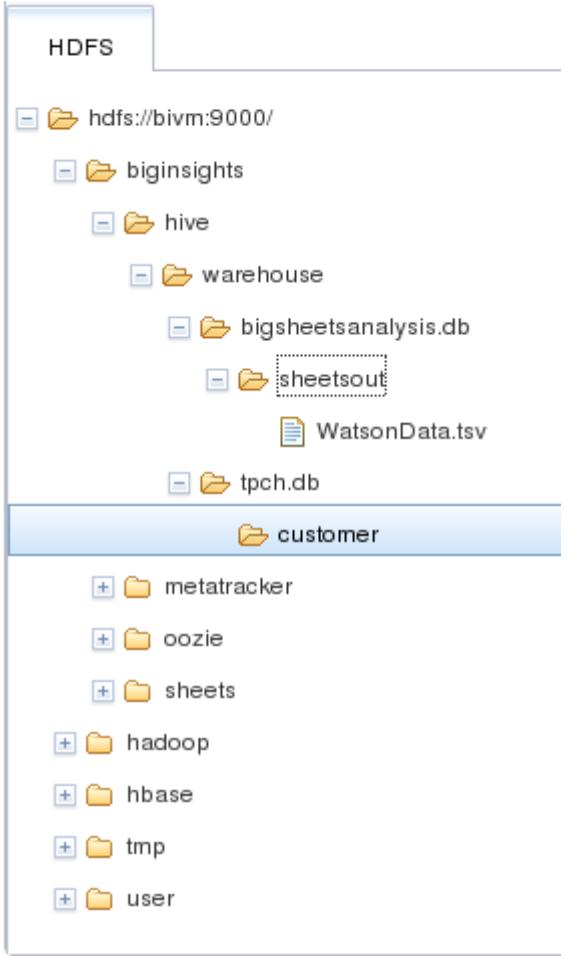
* In beta

Results from CREATE TABLE . . .

- **Table**
  - Subdirectory created in warehouse directory
    `/biginsights/hive/warehouse``/`*`tablename/`*
  - **External** tables may have their data stored anywhere in the DFS
  - Populated tables contain 1 or more data files

- **Schema (or database)**
  - Tables may be organized by schemas
  - Schema is just a collection of tables
  - Creating a schema creates a subdirectory in the warehouse to hold the tables
    `/biginsights/hive/warehouse/`*`schema`*`.db/`*`table name/`*

- **Catalog data (more later)**



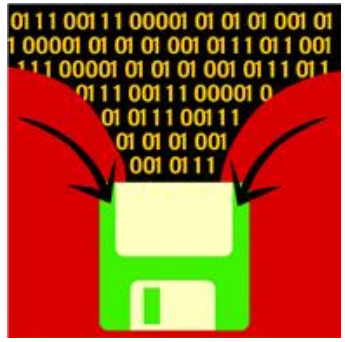
Big SQL Extensions to CREATE TABLE

- **Additional data types: BINARY(N), VARCHAR(N), DECIMAL(P,S)**
- **NULL/NOT NULL indicators**
  - These are advisory only – not enforced
  - Big SQL query re-write software takes advantage of this info
- **Table hints**
  - Certain optimizer hints can be attached to tables
  - Hint will automatically apply when the table is used in a query

```
create table offices
(
  office_id int           not null,
  name      string        not null
)
...
with hints (tablesize='small')
```

- **Explicit syntax for HBase tables (column mappings, column family options, . . .)**

Populating Tables



- **Data can be LOADed from . . .**
  - Local file system
  - Distributed file system
  - Netezza, DB2, Oracle, Informix, MS-SQL, Teradata
  - Example

```
CREATE TABLE EMPLOYEE (EMPNO INT, NAME STRING, AGE INT) . . . ;

// Overwrite any existing  data with new data from a local file
LOAD HIVE DATA  LOCAL INPATH '/home/user1/employee.data'  OVERWRITE  INTO TABLE EMPLOYEE;

// Append new data from a file in HDFS to the table
LOAD HIVE DATA INPATH '/user/biadmin/employee.data‘  INTO TABLE EMPLOYEE;
```

- **What LOAD does:**
  - Copies or moves the data, but doesn’t manipulate it
  - Format of the input file must match the format of the table

- **HBase notes:**
  - Similar LOAD syntax (LOAD HBASE ….). Composite keys, indexes, column encoding handled.
  - A single row INSERT may be used against HBase table

Querying data: Overview of SQL Support

- **Projection**

```
SELECT col1, col2  FROM t1
```

- **Restriction**

```
SELECT * FROM t1 WHERE col1 > 5
```

- **Union**

```
SELECT EMPNO FROM EMPLOYEE WHERE WORKDEPT LIKE 'E%'
UNION
SELECT EMPNO FROM ACTIVITIES WHERE PROJNO IN('MA2100', 'MA2110', 'MA2112')
```

- **Difference (EXCEPT)**

```
(SELECT * FROM T1) EXCEPT ALL (SELECT * FROM T2)
```

- **Intersection**

```
(SELECT * FROM T1) INTERSECT (SELECT * FROM T2)
```

- **Joins**
- **Subqueries**
- **Built-in functions**

SQL Support - Joins

- **Big SQL supports both common and ANSI / ISO join syntax**

```
select ... from tpch.orders,
tpch.lineitem
where o_orderkey = l_orderkey
```

```
select ... from tpch.orders
join tpch.lineitem
on o_orderkey = l_orderkey
```

SQL Support – Subqueries

- **Big SQL supports subqueries in SELECT and WHERE clauses**

```
select c1, (select
  count(*) from t2)
from t1
where ...
```

```
select c1
 from t1
 where c2 > (select ...)
```

SQL Support – Aggregates

- **Big SQL supports windowed aggregates**

```
SELECT EXTRACT(YEAR FROM CAST(CAST (order_day_key AS
varchar(100)) AS timestamp)) AS year,
SUM (sale_total) AS total_sales,
RANK () OVER (ORDER BY SUM (sale_total) DESC) AS ranked_sales
FROM gosalesdw.sls_sales_fact
GROUP BY EXTRACT(YEAR FROM CAST(CAST (order_day_key AS
varchar(100)) AS timestamp))
```

year	total_sales	ranked_sales
2006	1495891100.90	1
2005	1159195590.16	2
2007	1117336274.07	3
2004	914352803.72	4

SQL Support – Functions (partial list)

- **Numeric**

abs	ceil	floor	ln	log10
mod	power	sqrt	sign	width_bucket

- **Trigonometric**

cos	sin	tan	acos	asin
atan	cosh	sinh	tanh	

- **String**

char_length	bit_length	octet_length	upper	lower
substring	position	index	translate	trim
json_get_object				

- **Aggregates, etc.**

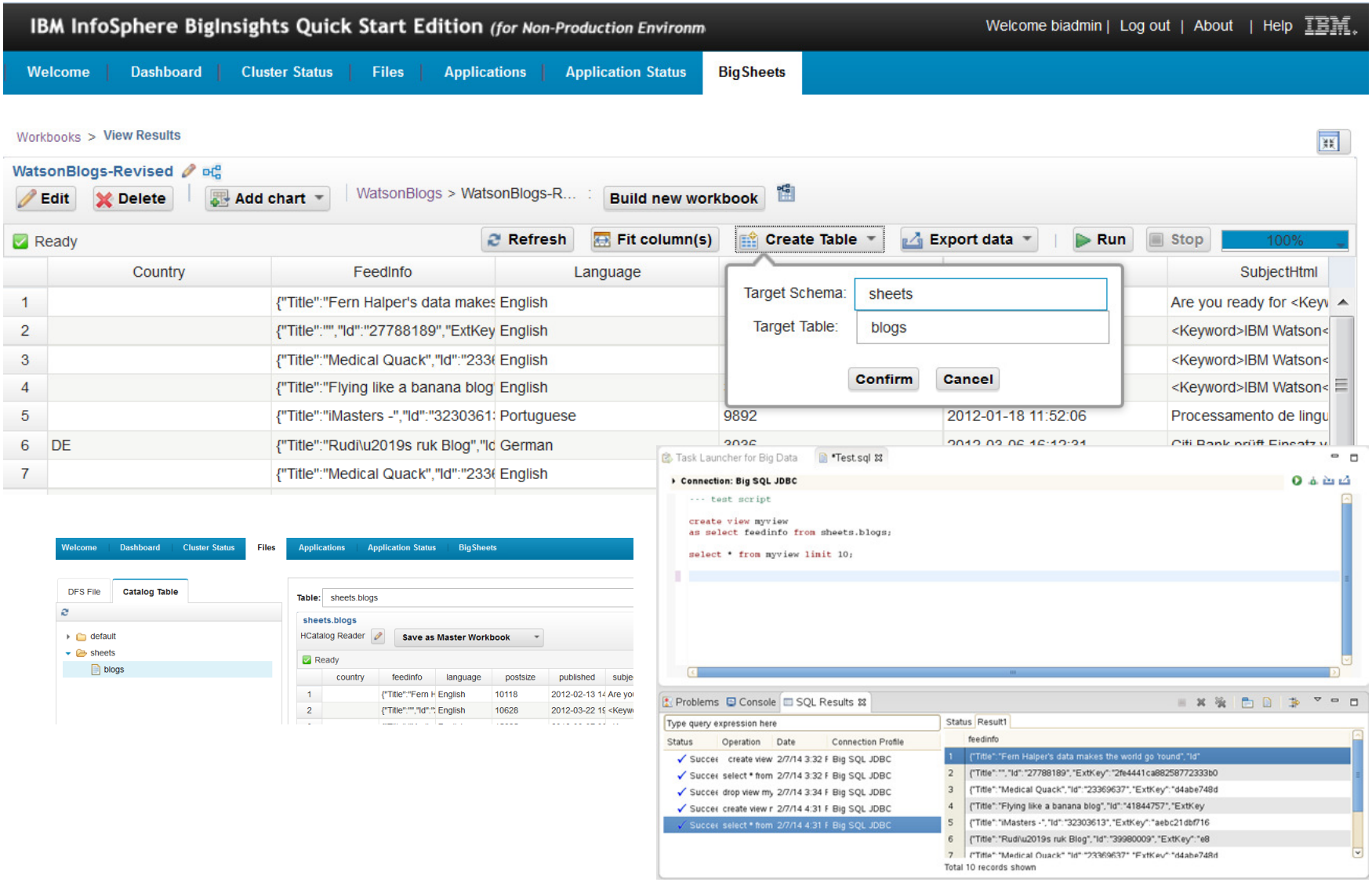
Catalog Tables (HCatalog)

```
[localhost][foo] 1> select * from syscat.tables where tablename='users';
+------------+-----------+
| schemaname | tablename |
+------------+-----------+
| default    | users     |
+------------+-----------+
1 row in results(first row: 0.14s; total: 0.15s)
```

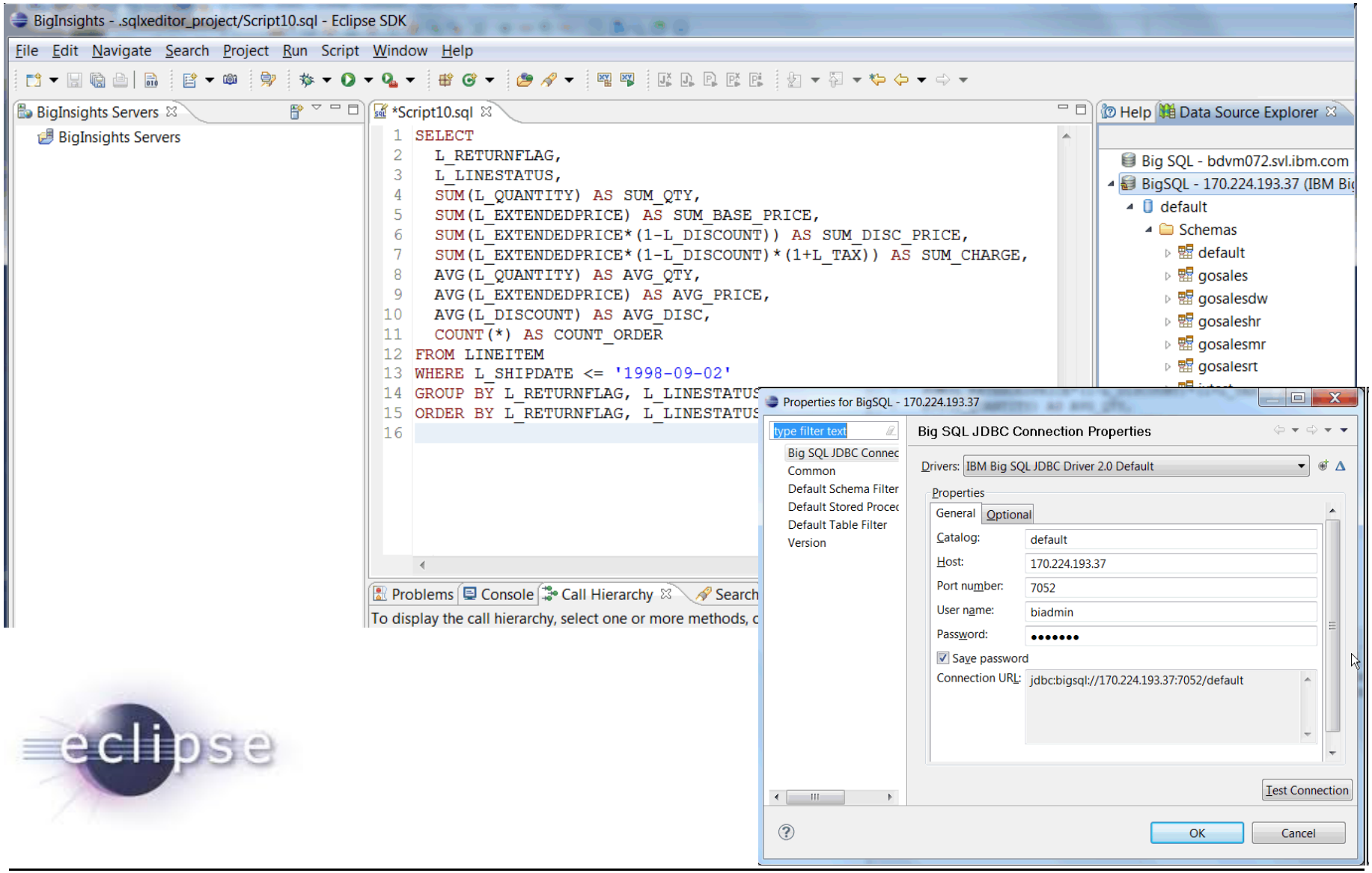
```
[localhost][foo] 1> select * from syscat.columns where tablename='users';
+------------+-----------+-----------+--------+-----------+-------+
| schemaname | tablename | name      | type   | precision | scale |
+------------+-----------+-----------+--------+-----------+-------+
| default    | users     | id        | INT    |        10 |     0 |
| default    | users     | office_id | INT    |        10 |     0 |
| default    | users     | name      | STRING |         0 |     0 |
| default    | users     | children  | ARRAY  |         0 |     0 |
+------------+-----------+-----------+--------+-----------+-------+
4 rows in results(first row: 0.19s; total: 0.21s)
```

Other BigInsights catalog tables track index and schema information

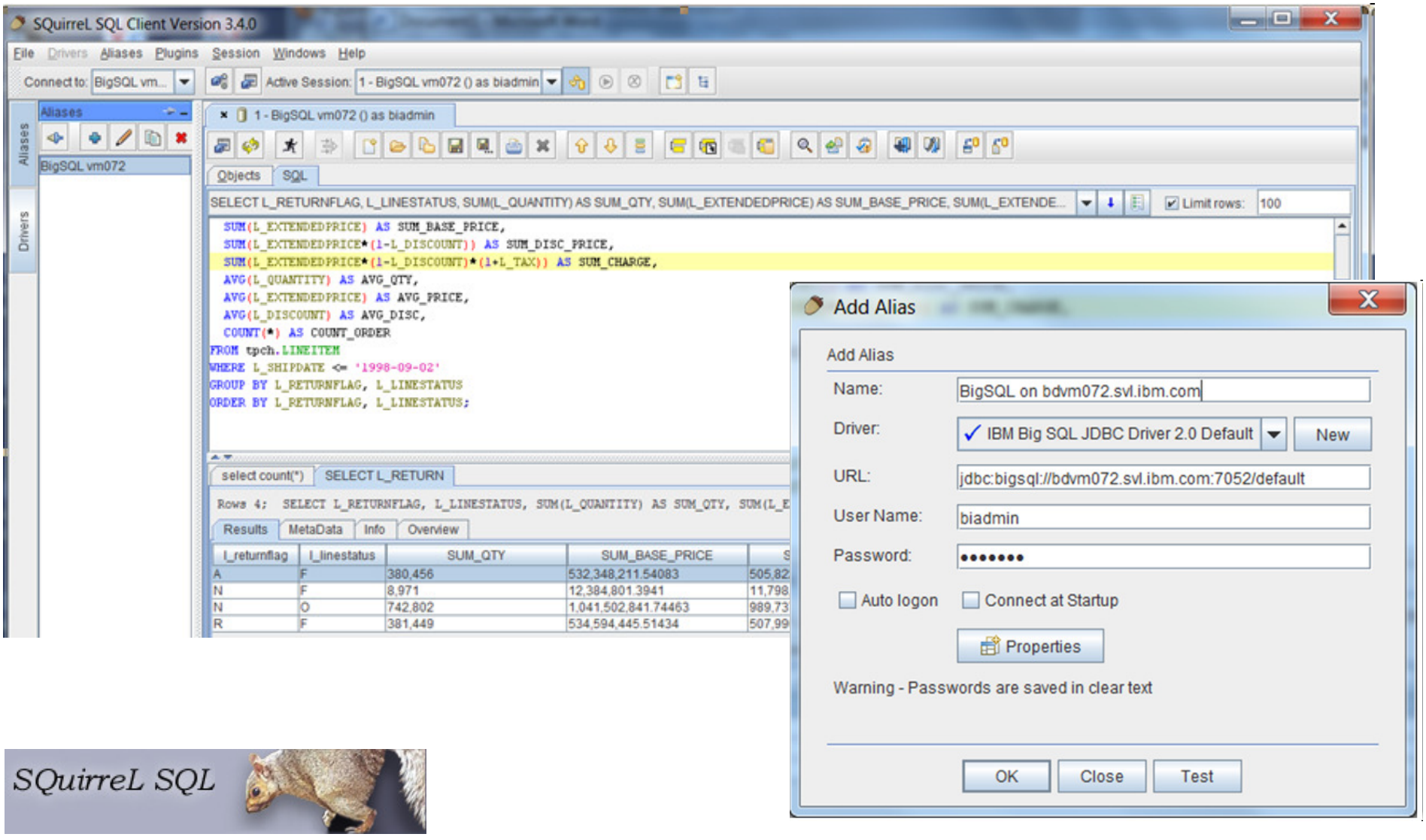
BigSheets and Big SQL



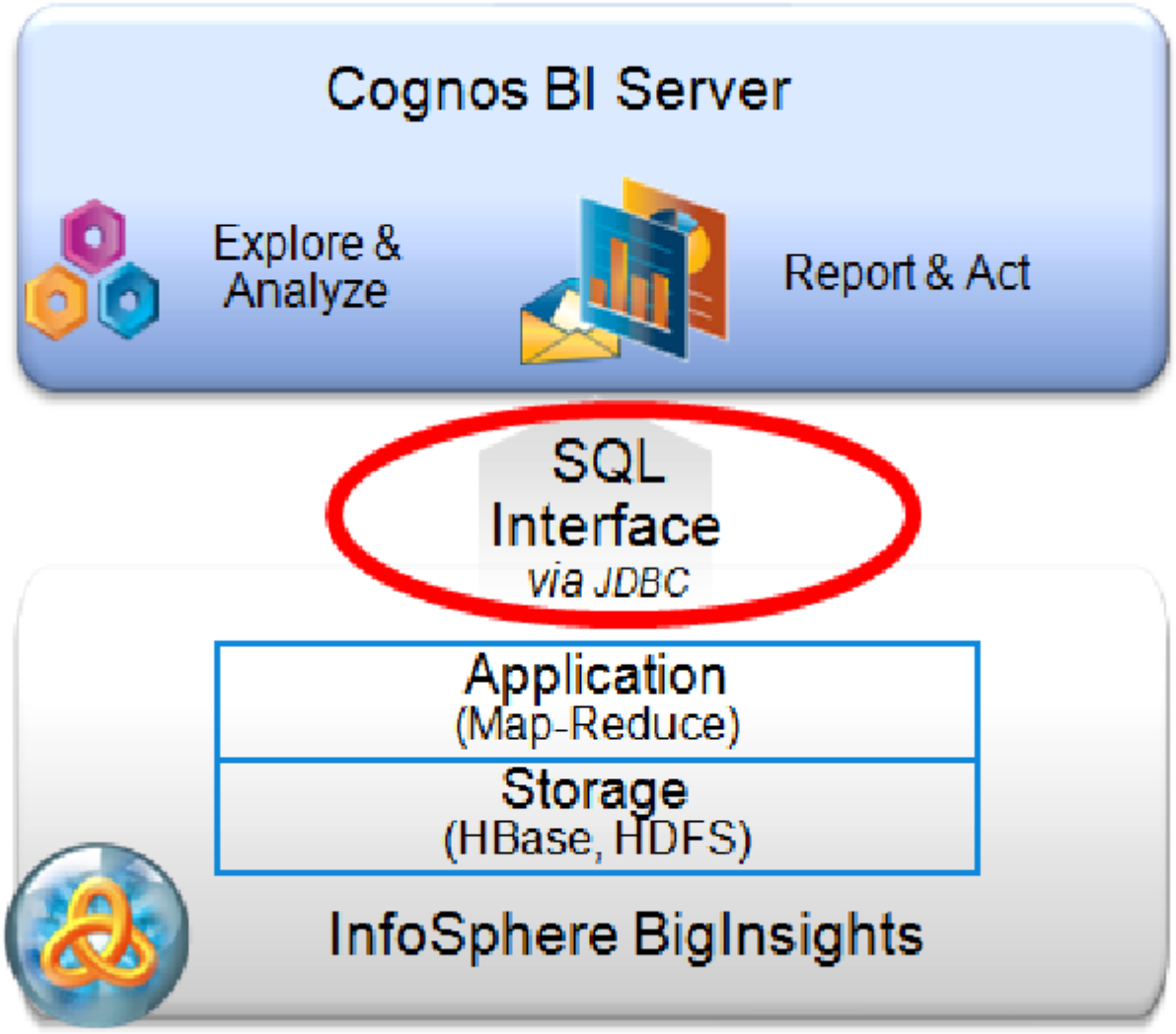
Using Existing Standard SQL Tools: Eclipse



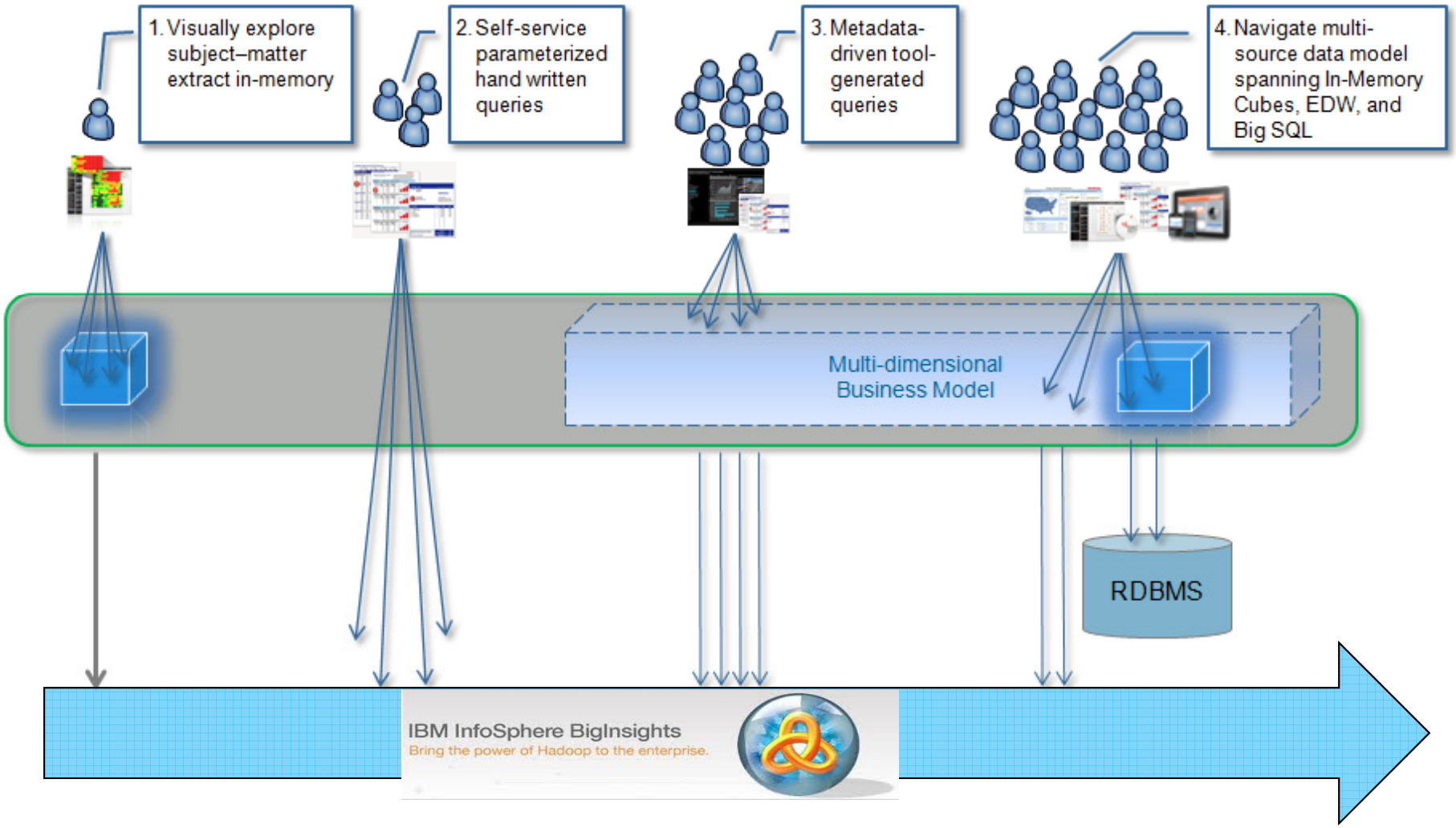
Using Existing Standard SQL Tools: SQuirreL SQL



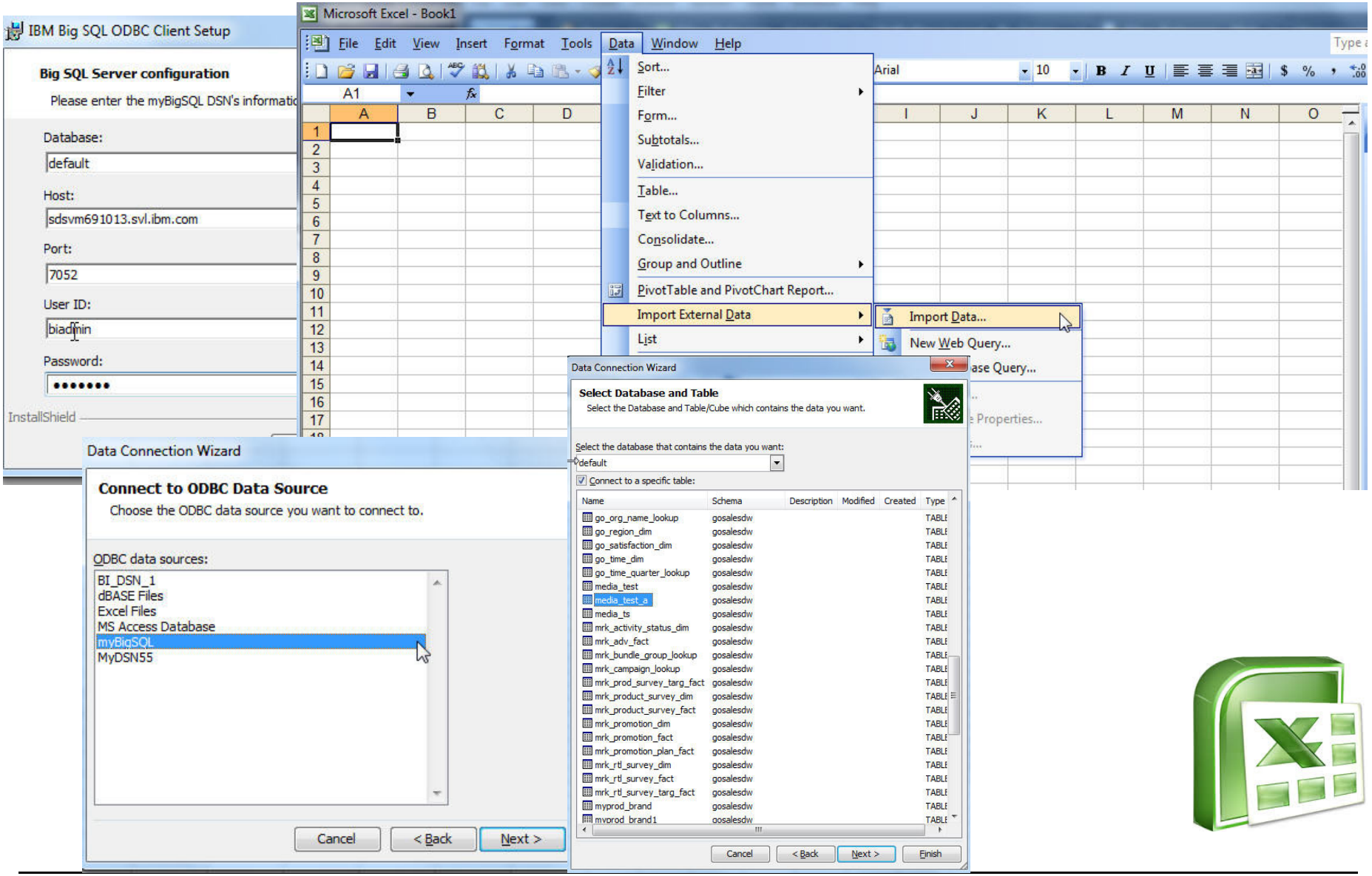
Cognos Business Intelligence



MicroStrategy use of Big SQL



MS Excel: Big SQL integration via ODBC



A word about . . . SerDes

- **Custom serializers / deserializers (SerDes)**
  - Read / write complex or “unusual” data formats (e.g., JSON)
  - Commonly used with Hive, HBase
  - Developed by user or available from open source community

- **Using SerDes with Big SQL**
  - Add the SerDe .jar file to `$BIGSQL_HOME/userlib` and `$HIVE_HOME/lib`
  - Stop / restart Big SQL service
  - Specify SerDe class name (not .jar file name) when creating table

- **Example**

```
-- Create a table for JSON data. Use open source hive-json-serde-0.2.jar SerDe
create table socialmedia-json (Country String, FeedInfo String, . . . )
row format serde 'org.apache.hadoop.hive.contrib.serde2.JsonSerde'
stored as textfile;

load hive data inpath '</hdfs_path>/WatsonBlogsData.json' overwrite into table
  socialmedia-json;

select * from socialmedia-json;
```

Sample JSON input for previous example

```
[biadmin@bdvm327 twitter]$ cat WatsonNewsBlogsData.json|more

[{"PostSize":5775,"ThreadId":"4f129a8be","Crawled":"2012-01-15 09:21:15","FeedInfo":"{\"Tit
le\":\"www.ibm.comnews\",\"Id\":\"44032787\",\"ExtKey\":\"879cd3257c296614160914c3d96f9b85\
",\"Url\":\"http://www-03.ibm.com\"}","Published":"2012-01-15 09:21:15","Url":"http://www.i
bm.com/innovation/us/watson/?lnk=ftpl","Country":"US","SubjectHtml":"<Keyword>IBM<\/Keyword
> - <Keyword>Watson<\/Keyword>","Inserted":"2012-05-29 00:52:57","Language":"English","Text
Html":"<![CDATA[<Keyword>IBM<\/Keyword> - <Keyword>Watson<\/Keyword>\n\n      Call to find
out how Watson's capabilities could benefit your business.\n      1-800-426-7630\n    \nRel
ated content\nDesigning the Computer for a Smarter Planet\nThere\u2019s an enormous amount
of science included when <Keyword>Watson<\/Keyword> answers a single Jeopardy! question, ho
w does it all work together?\nExplore <Keyword>Watson<\/Keyword>\nBeyond Jeopardy! The Busi
ness Implications of <Keyword>Watson<\/Keyword>\n<Keyword>IBM<\/Keyword> <Keyword>Watson<\/
Keyword> passed its first test onÂ Jeopardy!Â in February 2011, but the real test will be i
n
applying the underlying systems, data management and analytics technology in business and a
cross different industries. Watch the webcast now and learn about the present and future bu
siness implications of Deep QA and the other technologies behind <Keyword>Watson<\/Keyword>
 from David Ferrucci and other <Keyword>IBM<\/Keyword> executives.\nRegister now\nBetter Bu
```

JSON-based social media data to load into Big SQL Table *socialmedia-json* defined with SerDe

Sample Big SQL query output for JSON data

authorinfo	country	crawled	feedinfo	id	inserted	published
{"Nick":"","Id":"","Name":"","Url":""}	US	2012-04-11 03:31:47	{"Title":"www.ibm.comnews","Id":"44032787","ExtKey":"879cd3257c296614160914c3d96f9b85","Url":"http://www-03.ibm.com"}	3185931232	2012-05-23 20:18:08	2012-04-11 03:31:47
{"Nick":"","Id":"","Name":"","Url":""}	US	2012-02-22 23:50:14	{"Title":"www.ibm.comnews","Id":"44032787","ExtKey":"879cd3257c296614160914c3d96f9b85","Url":"http://www-03.ibm.com"}	3253594848	2012-05-26 05:21:06	2012-02-22 23:50:14

Sample output: Select * from socialmedia-json

A word about . . . performance

- **Tuning options**
  - Table design (e.g., storage formats for Hive, key & column family definitions for HBase)
  - Hints in queries, table definitions
  - ANALYZE TABLE … COMPUTE STATISTICS command
  - Secondary indexes (HBase tables only)
  - MapReduce job properties
  - . . .
- **Query hints provided in comments:** `/*+ name=value [, …] +*/`

```
select * from foo /*+ accessmode='local' +*/ where c1 < 1000;
```

- **Access mode hint**
  - Causes query to be executed in the Big SQL server
  - HBase indexed queries can return extremely rapidly
  - Local access can be forced on for your entire session

Agenda

- **Big SQL: motivation and architecture**

- **Using Big SQL**
  - Invocation options
  - Creating tables
  - Populating tables with data
  - Querying data
  - Developing applications and working with tools
  - . . . And a peek at some additional topics

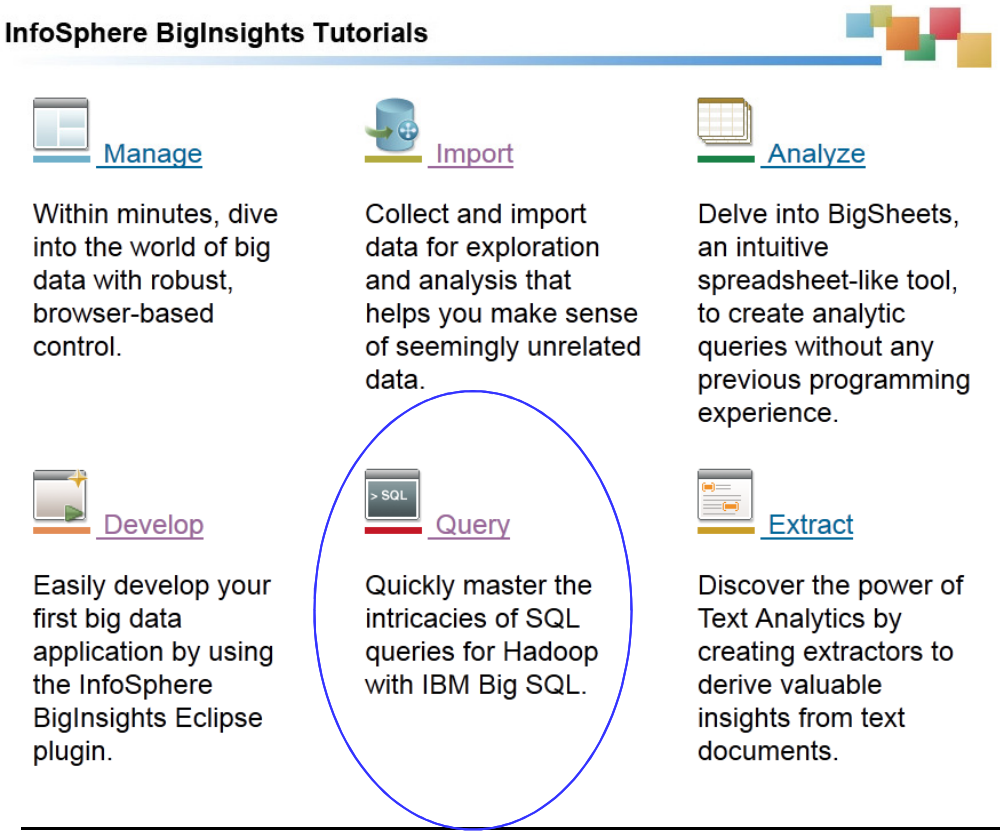
- **What RDBMS professionals should know about Big SQL**

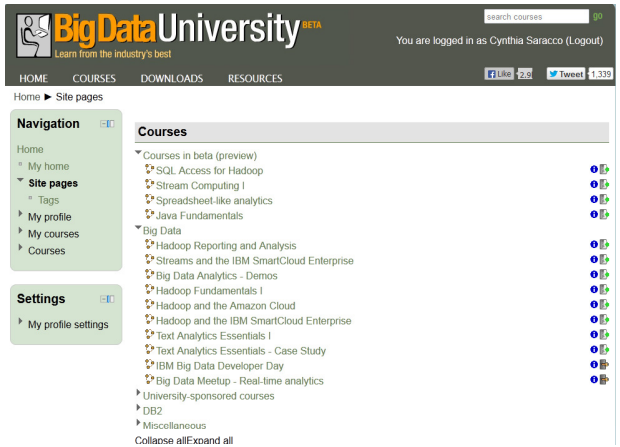
Big SQL – what RDBMS experts should know

- **Big SQL provides industry-standard query support for Hadoop-based storage managers**
  - Exploits Hadoop environment
  - Includes Hadoop-specific extensions
  - Introduces Hadoop-specific concepts
  - Copes with “unconventional” data structures and formats (e.g., JSON) via SerDes, other features
- **RDBMS = more than query & storage management**
  - Transaction management
  - Stored procedures
  - INSERT / UPDATE / DELETE
  - GRANT / REVOKE
  - 3GL language support (e.g., COBOL)
  - Rich catalog statistics and decades of cost-based optimization development
- **Bottom line: Big SQL provides SQL experts with on-ramp to Hadoop, but doesn’t turn Hadoop into one big relational database**

Want to learn more?

- **Big SQL tutorial (product Information Center)**
- **Videos , articles, downloads, etc.**
  - Technical portal at http://tinyurl.com/biginsights





YouTube

BigInsights Technical Enablement Wiki

Get up to speed on InfoSphere BigInsights, IBM's software platform designed to help firms store, manage, and analyze "big data".

Technical materials

- Articles, white papers, and books
- BigInsights InfoCenter

Videos and Demos

- Video guide

Downloads

- BigInsights Basic Edition (free)
- Karmasphere Studio Community Edition Virtual Appliance with BigInsights (free)
- Fix packs for BigInsights Enterprise Edition (licensed)

Discussion Forum

- BigInsights forum on IBM developerWorks