

ORACLE®

ORACLE
OPEN
WORLD

Oracle Big Data SQL Deep Dive

Subtitle

October 1–5, 2017
SAN FRANCISCO, CA

Marty Gubar
Big Data SQL PM
Oracle Corporation

ORACLE®

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Big Data SQL Goals



Easily access any data
across big data stores



Provide a unified security
model across the sources

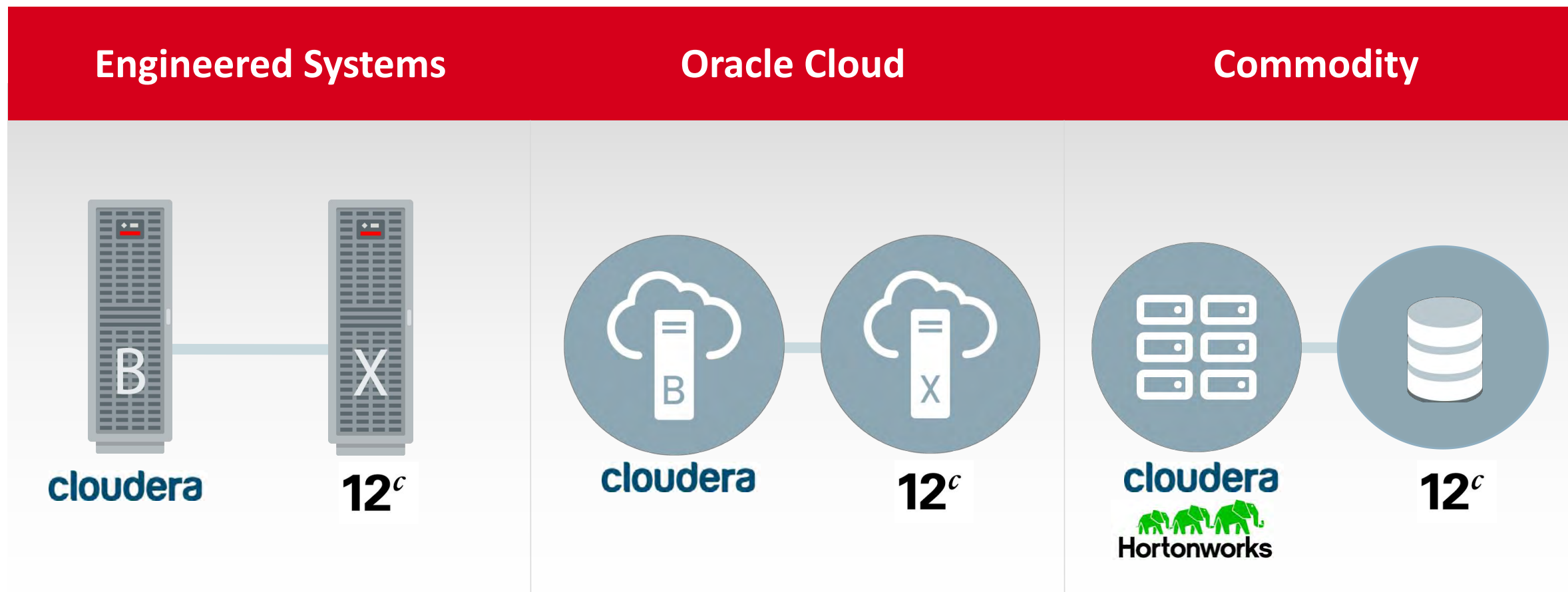


Analyze all data using
Oracle's rich SQL dialect

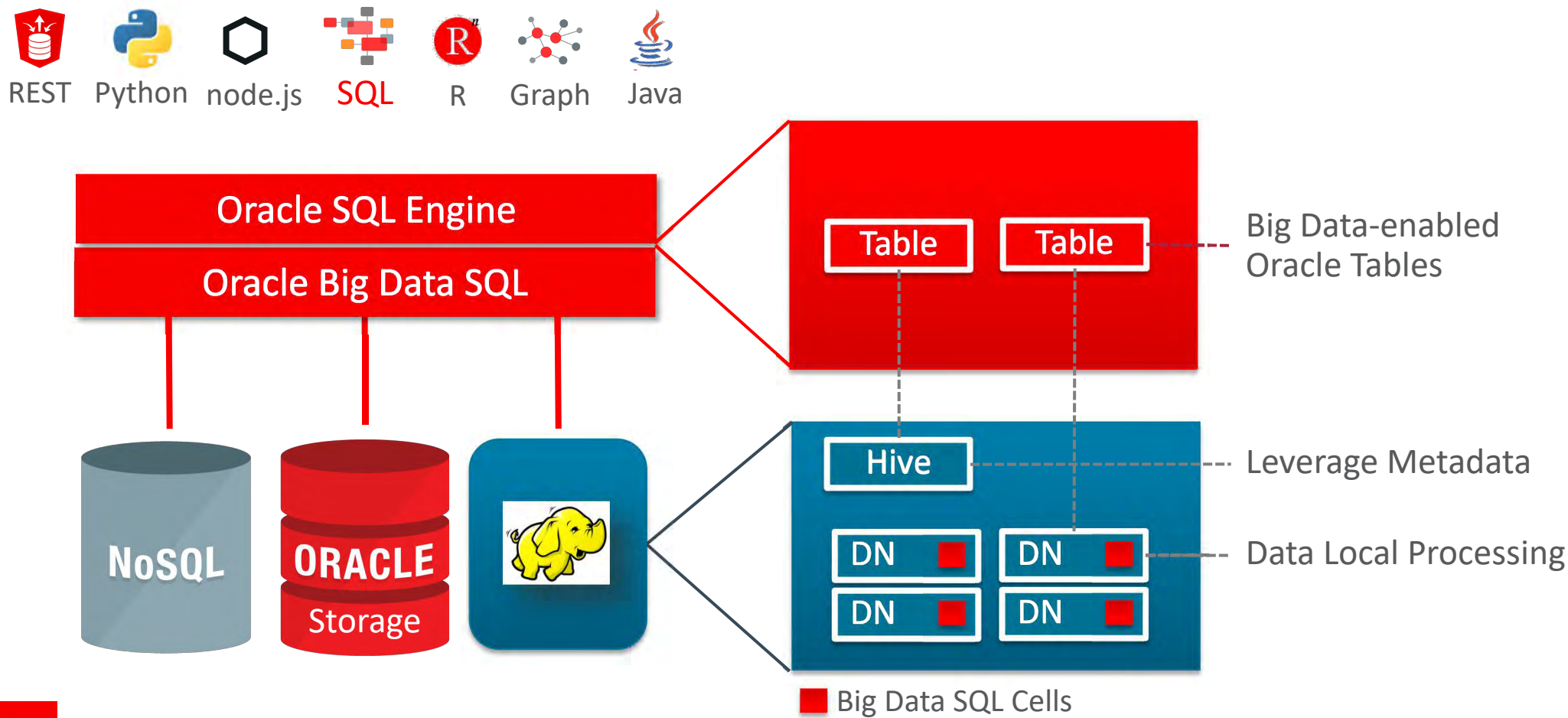


Fast performance using Big
Data SQL Smart Scan

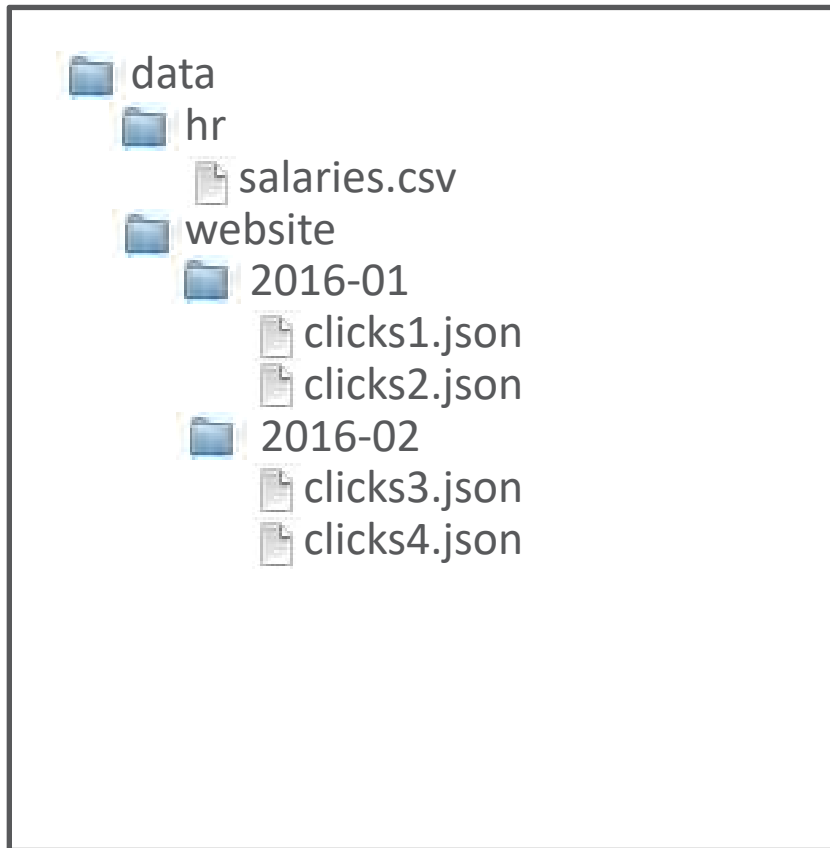
Flexible Deployment Options



Big Data SQL Architecture

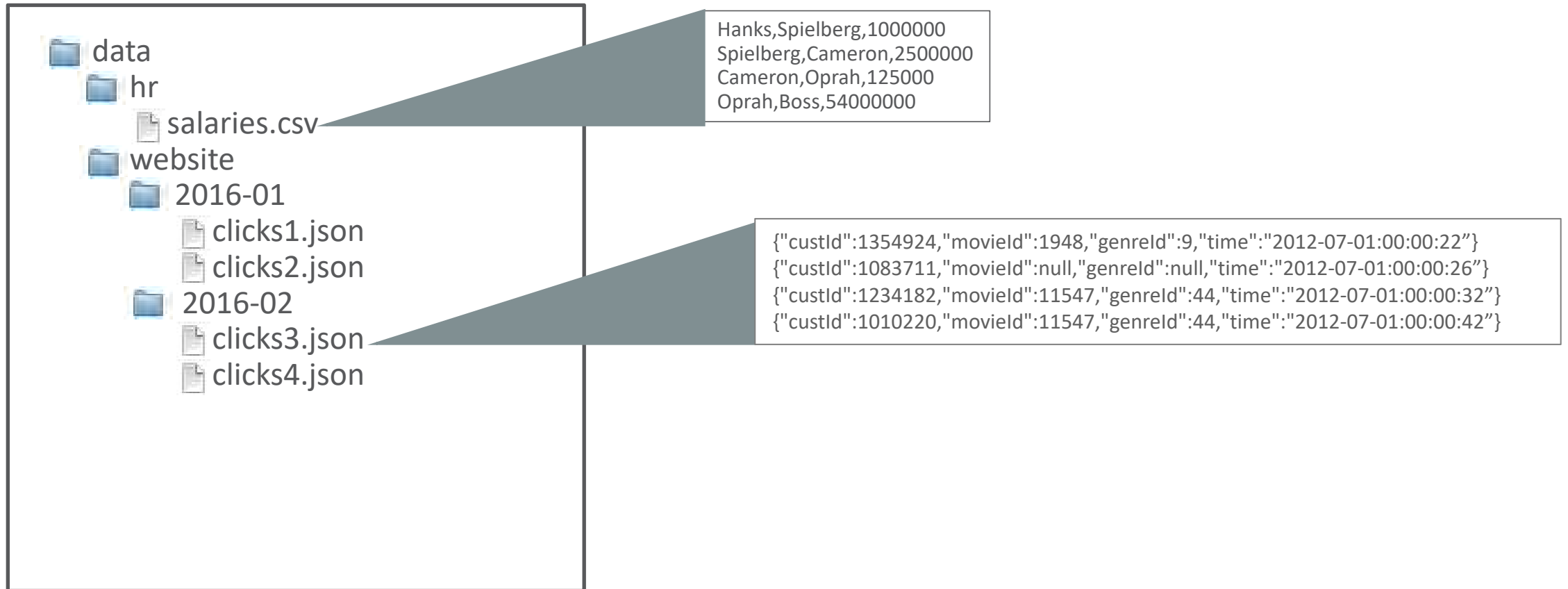


How Data is Stored in HDFS



- Data stored in files and organized into folders
 - Can be any file type
 - Replicated 3x across cluster
- Schema on Read
 - The tool reading the data interprets the data as it sees fit

How Data is Stored in HDFS

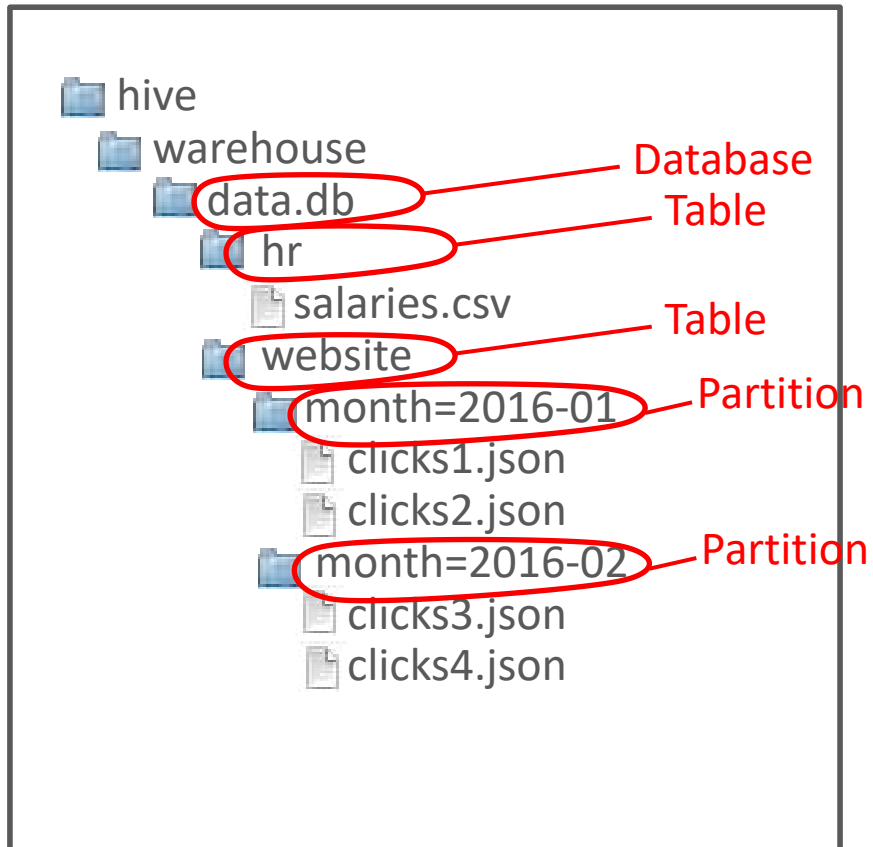


Demonstration: Accessing Data in HDFS

The screenshot displays the Oracle HUE web interface. On the left, the 'File Browser' pane shows the file structure: / user / hive / warehouse / csv.db / web_clickstreams_json / 000000_0. The main pane shows the content of this file as a JSON array of clickstream records. On the right, the 'SQL Worksheet' pane is active, showing a SQL query to create an external table named WEB_CLICKSTREAMS_JSON. The query uses the ORACLE_HDFS organization type and points to the same HDFS location as the file browser. The SQL text is as follows:

```
CREATE TABLE WEB_CLICKSTREAMS_JSON
( VAL VARCHAR2(4000 BYTE)
)
ORGANIZATION EXTERNAL
( TYPE ORACLE_HDFS
  DEFAULT DIRECTORY DEFAULT_DIR
  LOCATION ('/user/hive/warehouse/csv.db/web_clickstreams_json')
)
REJECT LIMIT UNLIMITED ;
```

Organize and Describe Data with Hive



- Information is captured in **Hive Metastore**
- HDFS Folders become:
 - Databases
 - Tables
 - Partitions
- **Table** includes metadata for parsing files using Java classes
 - **InputFormat** defines chunks called splits based on file type
 - **RecordReader** creates rows out of splits
 - **SerDe** creates columns

Recommended Approach

Use **ORACLE_HIVE** When Possible

- Oracle Database query execution accesses Hive metadata at describe time
 - Changes to underlying Hive access parameters will not impact Oracle table (one exception... column list)
- Metadata an enabler for performance optimizations
 - Partition pruning and predicate pushdown into intelligent sources
- Utilize tooling for simplified table definitions
 - SQL Developer and DBMS_HADOOP packages

Demonstration: Big Data SQL Leverages Hive Metadata

The image shows two screenshots from the HUE interface. The left screenshot displays the 'Metastore Manager' tab, showing a list of tables on the left and a detailed view of the 'createtab_stmt' table on the right. The right screenshot shows the 'Connections' panel with a context menu open over the 'customer' table, highlighting the option 'Use in Oracle Big Data SQL...'. An arrow points from the 'customer' table in the left screenshot to the 'customer' table in the right screenshot.

Table: createtab_stmt

25	ws_ext_sales_price decimal(1,2),
26	ws_ext_wholesale_cost decimal(7,2),
27	ws_ext_list_price decimal(7,2),
28	ws_ext_tax decimal(7,2),
29	ws_coupon_amt decimal(7,2),
30	ws_ext_ship_cost decimal(7,2),
31	ws_net_paid decimal(7,2),
32	ws_net_paid_inc_tax decimal(7,2),
33	ws_net_paid_inc_ship decimal(7,2),
34	ws_net_paid_inc_ship_tax decimal(7,2),
35	ws_net_profit decimal(7,2))
36	ROW FORMAT SERDE
37	'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
38	STORED AS INPUTFORMAT
39	'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
40	OUTPUTFORMAT
41	'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
42	LOCATION

Connections Panel:

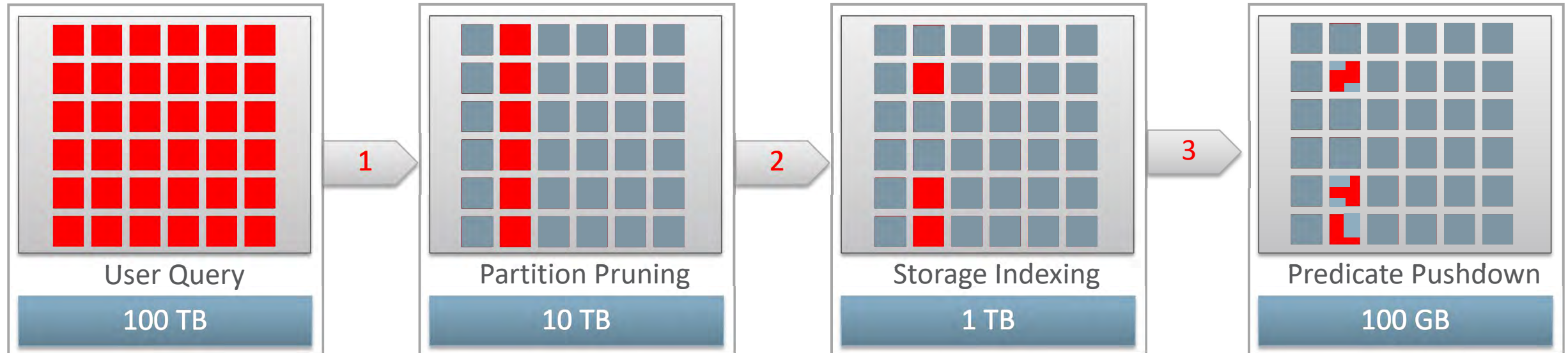
- Connections
 - bds
 - hive
 - csv
 - default
 - hr
 - orc
 - parq
- Tables
 - customer
 - customer
 - customer
 - date_dim
 - household_demographics
 - income_band
 - inventory
 - item

Context Menu:

- Edit...
- Open
- Export...
- Use as Template...
- Use in Oracle Big Data SQL...

Big Data SQL Performance Features

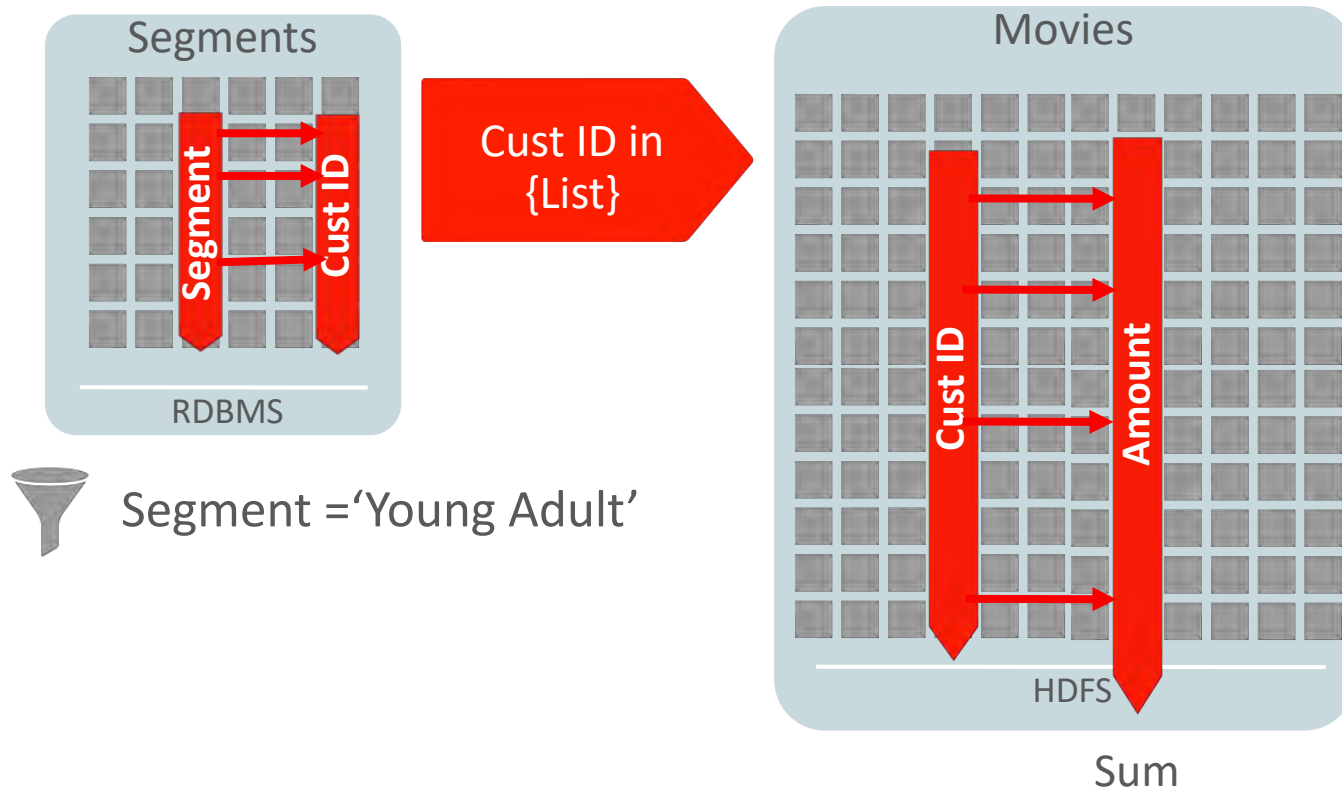
Compound IO Reduction thru Smart Scans



Big Data SQL Performance Features

Smart Scan – Execute Joins as Bloom Filters on Hadoop Nodes

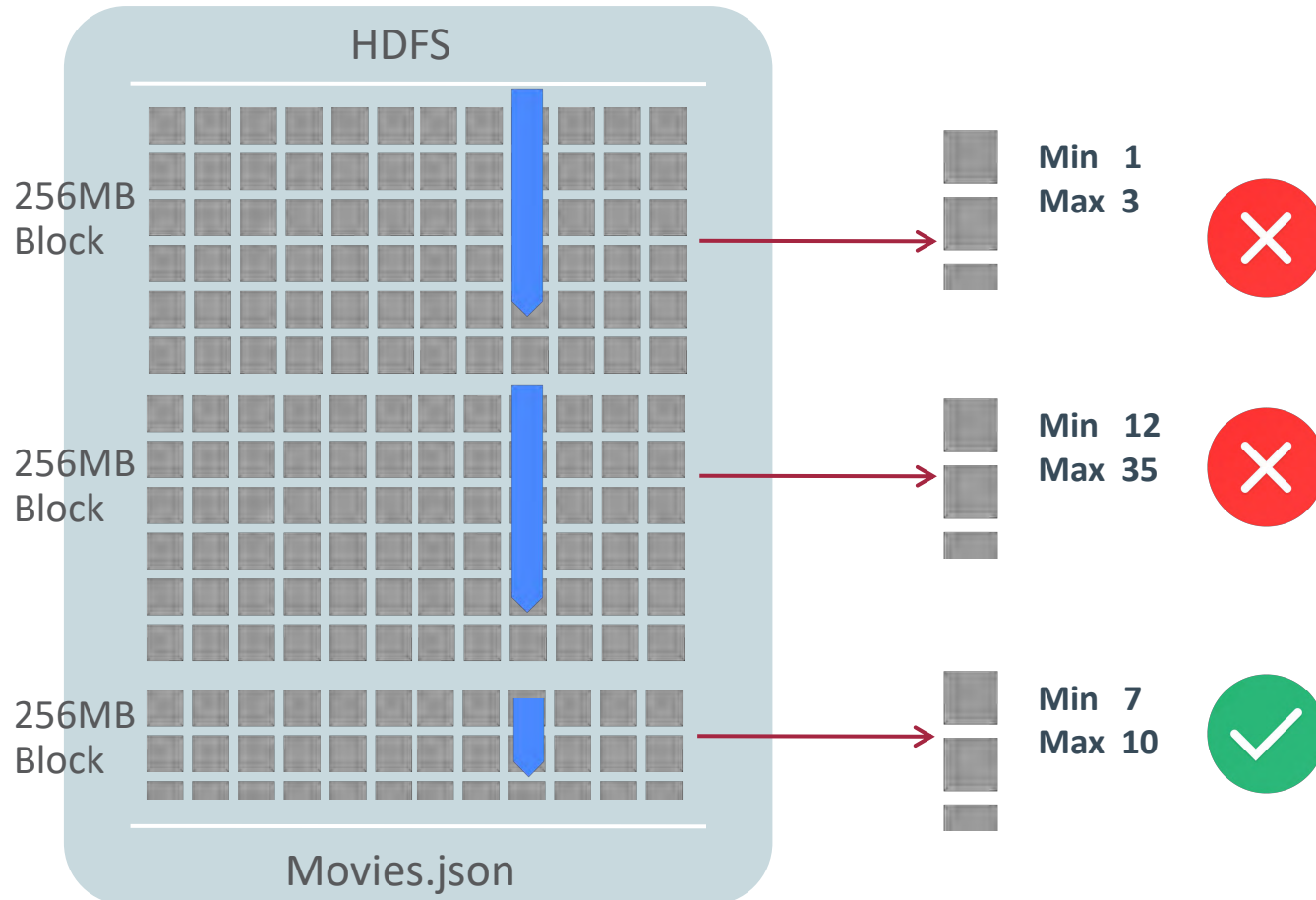
Example: Total movie sales for customer segment



- Converts joins of data in multiple tables into scans
- Result:
 - Scans are pushed down to Hadoop nodes and executed locally
 - No data moved to Database to process joins
 - Massive speed up of query
- Works with data spanning DB and Hadoop as well as data in two Hadoop data sets

Big Data SQL Storage Index

Example: Find revenue for movies in a category 9 (Comedy)



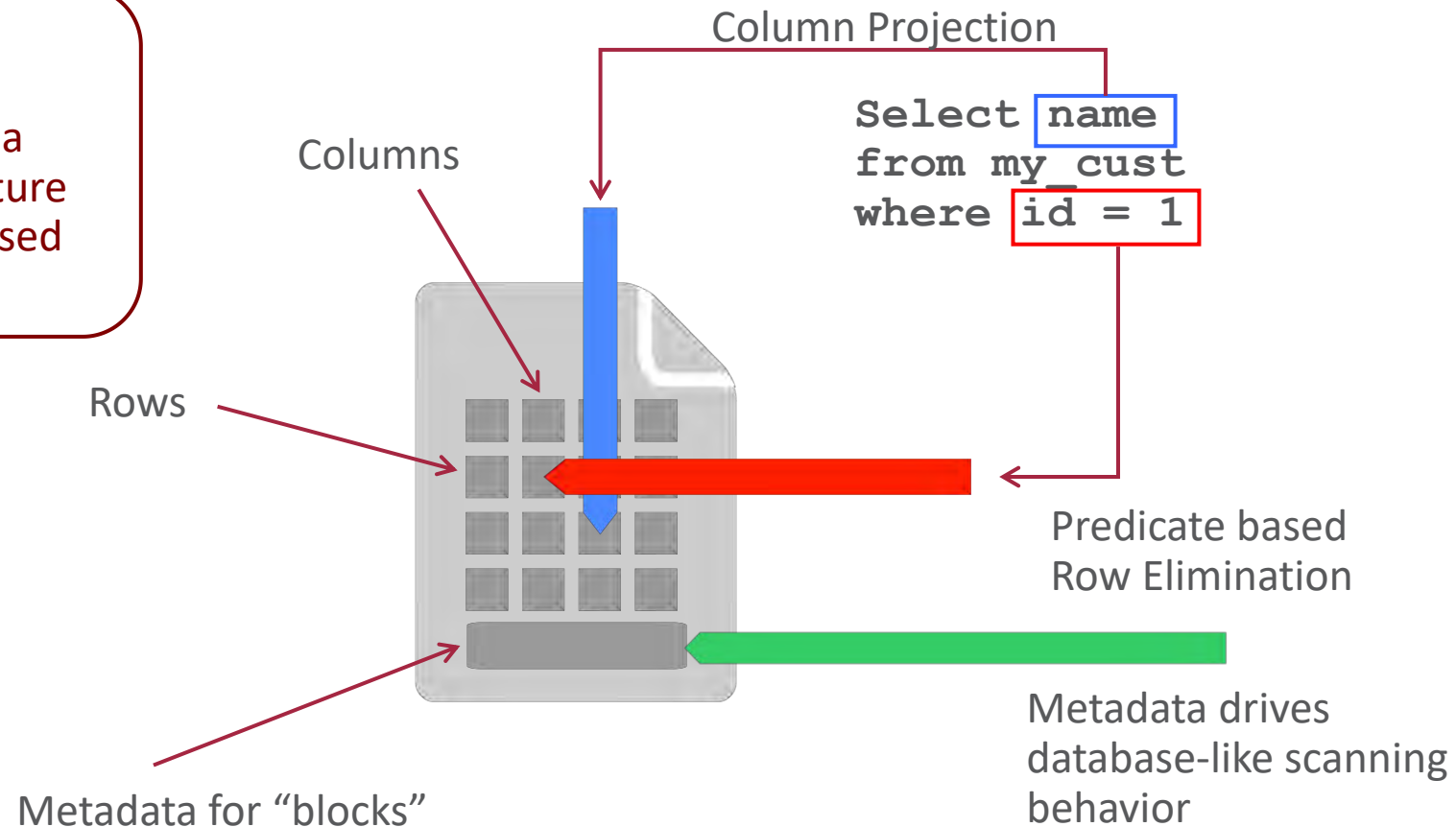
- Storage index provides query speed-up through transparent IO elimination of HDFS Blocks. It's a *negative index*
- Min / max value is recorded for columns included in a storage index (max # of columns = 32)
- Storage index provides partition pruning like performance for un-modeled data sets

Parquet: Columnar Database File in HDFS

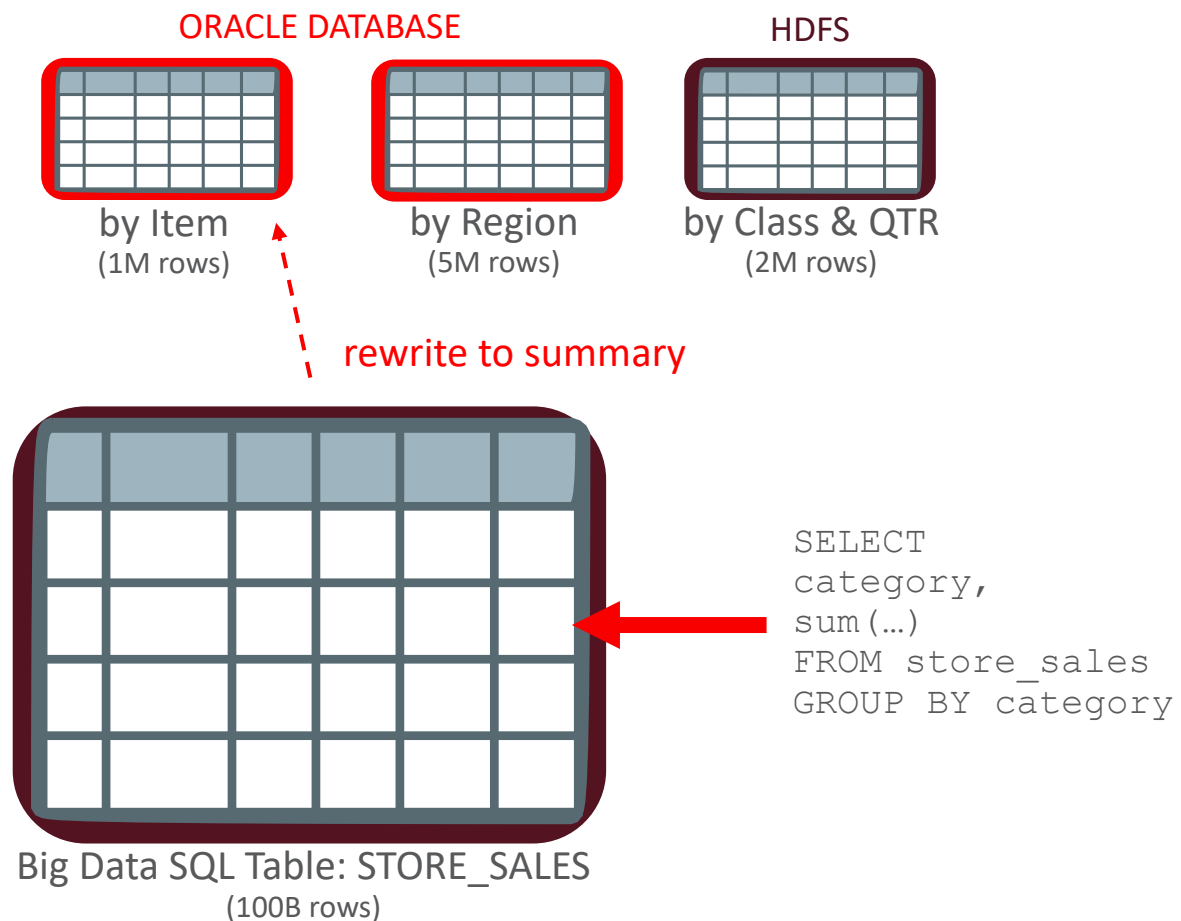
Big Data SQL Pushes Predicates to Parquet

Schema on Write

Parquet implements a database storage structure with metadata and parsed data elements

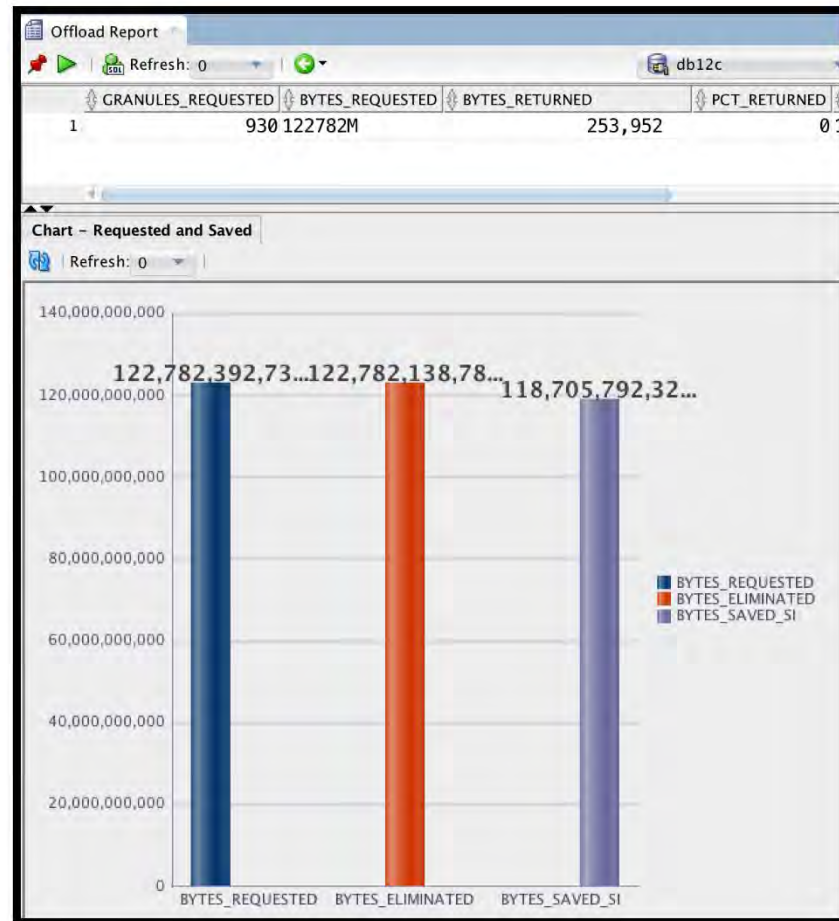


Enhance Performance with Automatic Query Rewrite



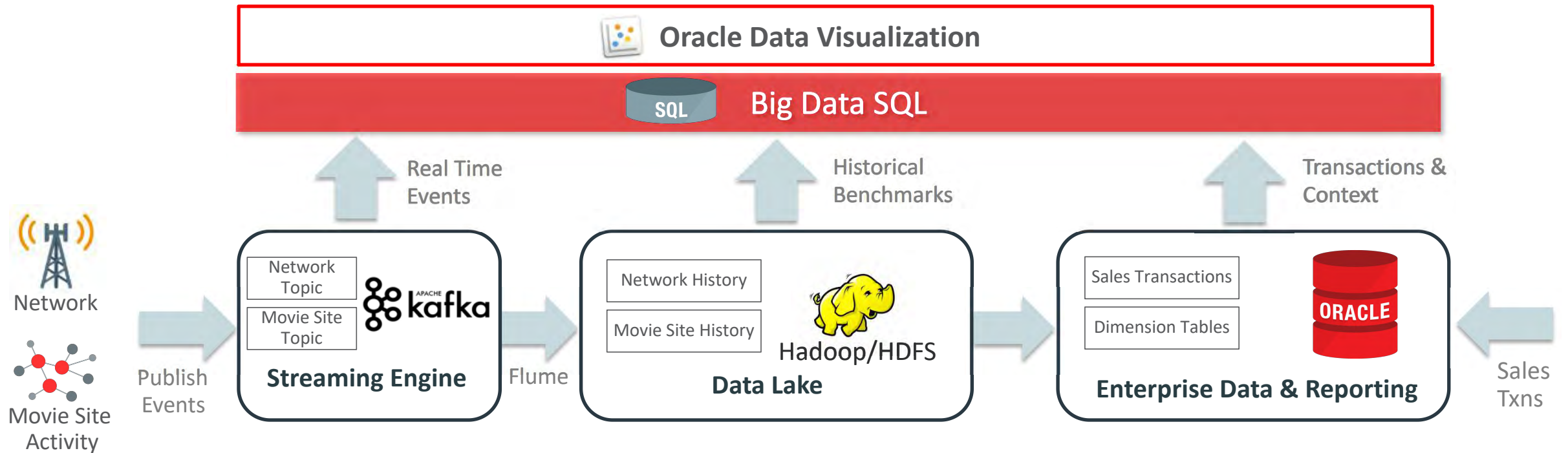
- **Orders of magnitude** performance improvement
- **Materialized view** query rewrite automatically redirects detail query to appropriate summary data
 - Store summaries in Oracle Database
 - If available, use existing summaries in HDFS
- No changes to query required

Performance Features



Demonstration Scenario

Analyzing Real-time Streams

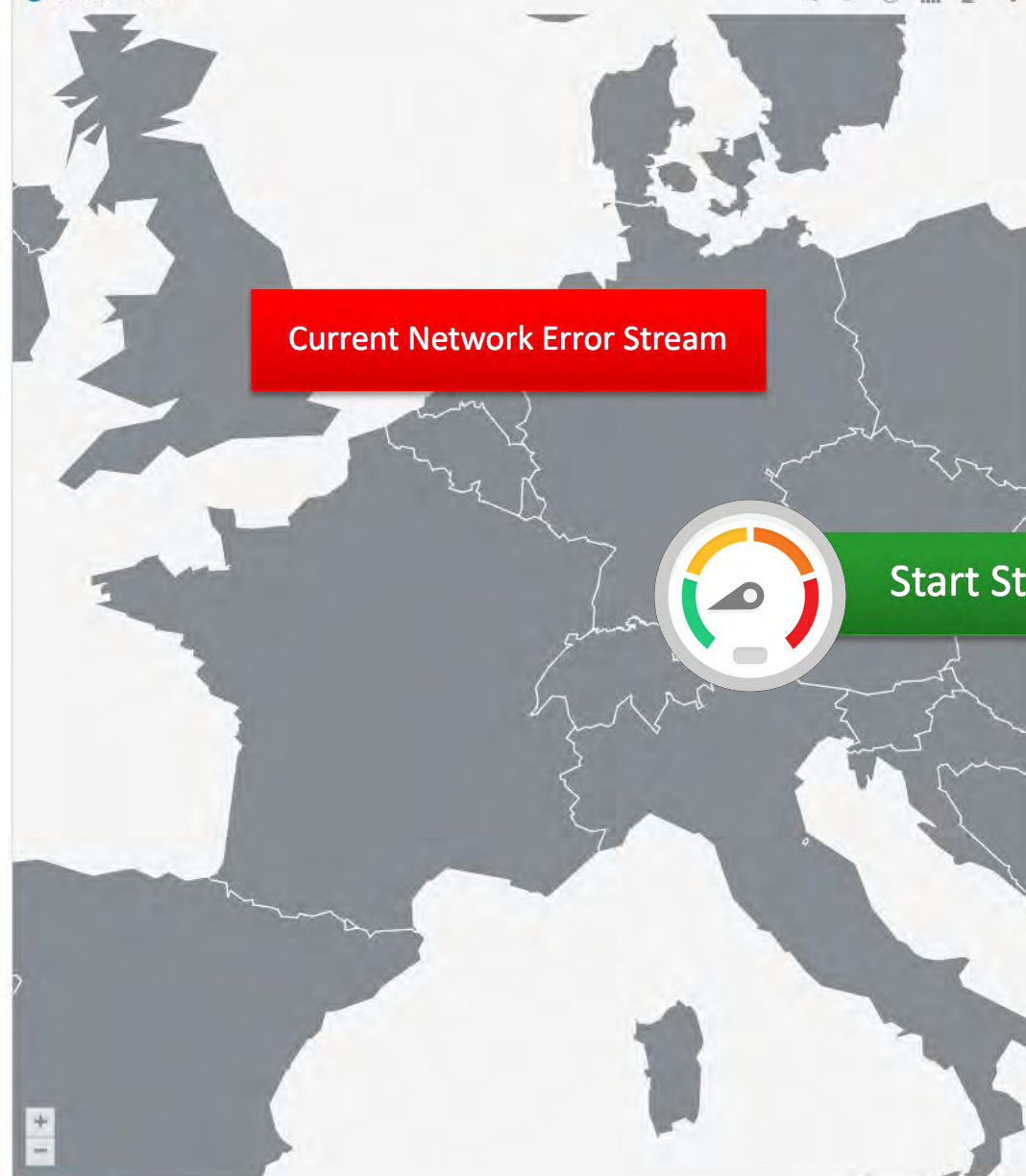


KAFKA_DEMO_TABLE

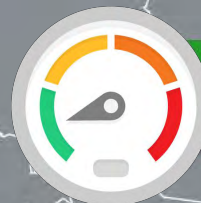
- Day
- Country
- Network Errors
- Hour
- Revenue Benchmark
- Revenue Stream

My Calculations
Value Labels

Network Errors

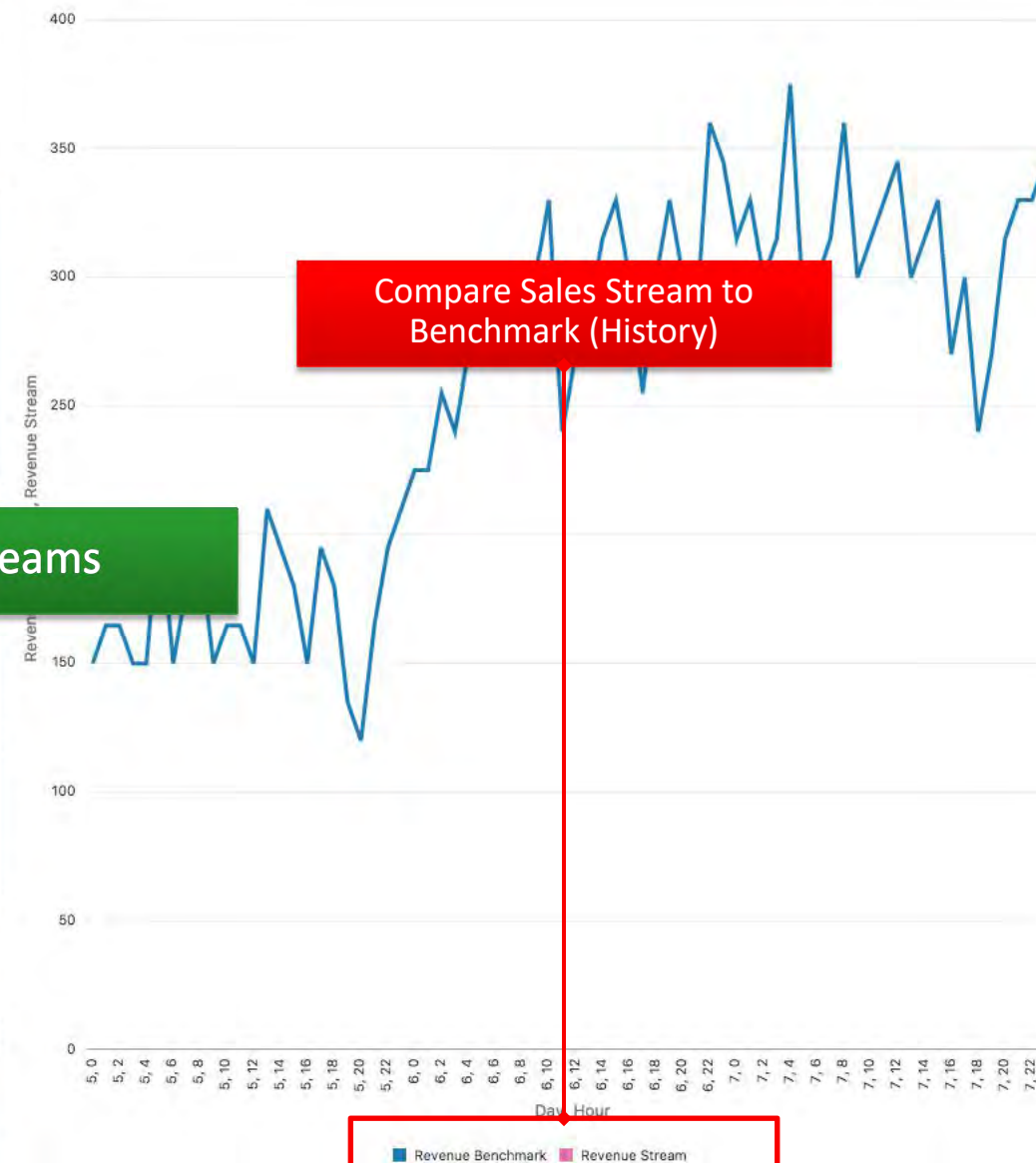


Current Network Error Stream



Start Streams

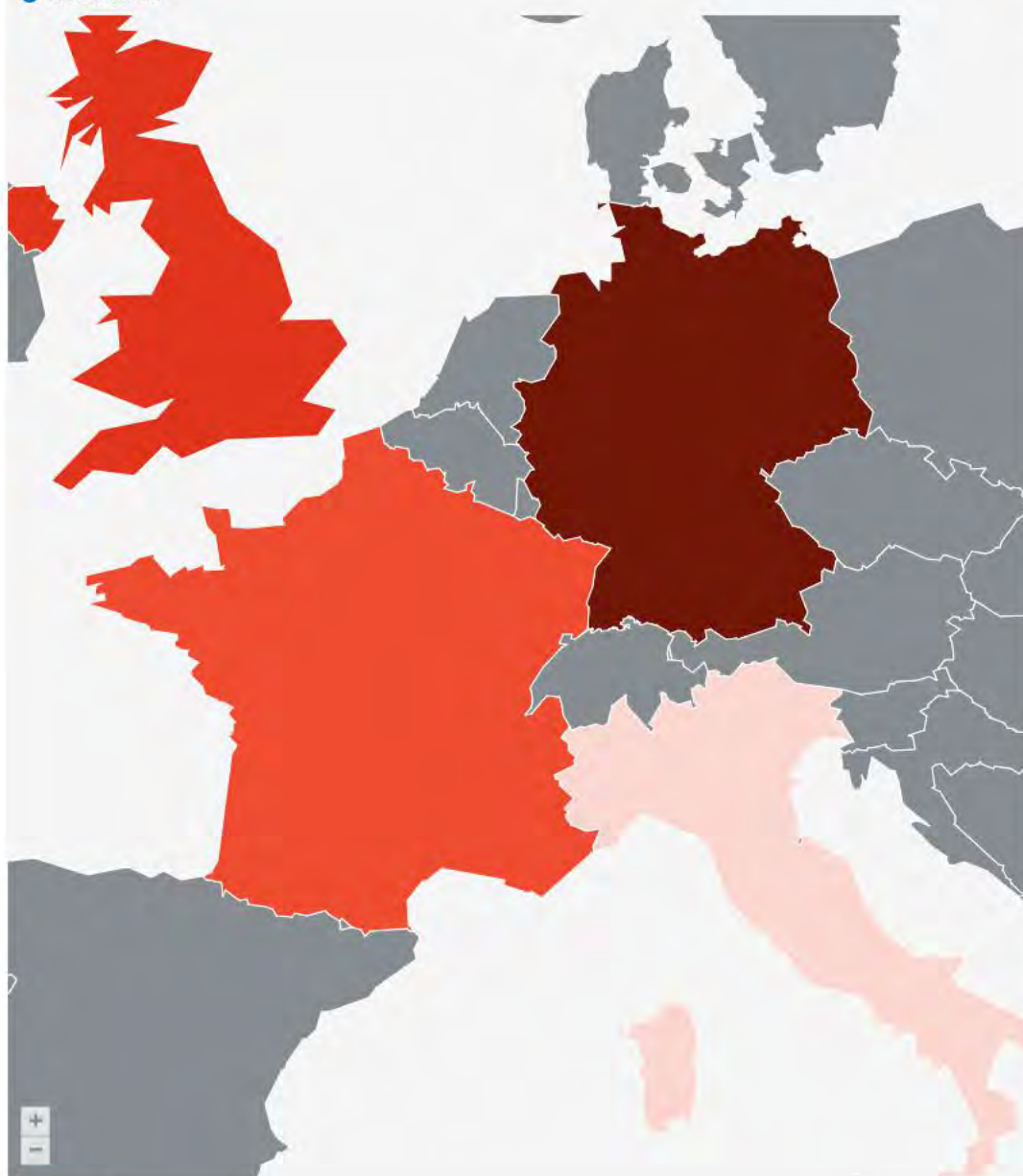
Revenue Stream vs Benchmark



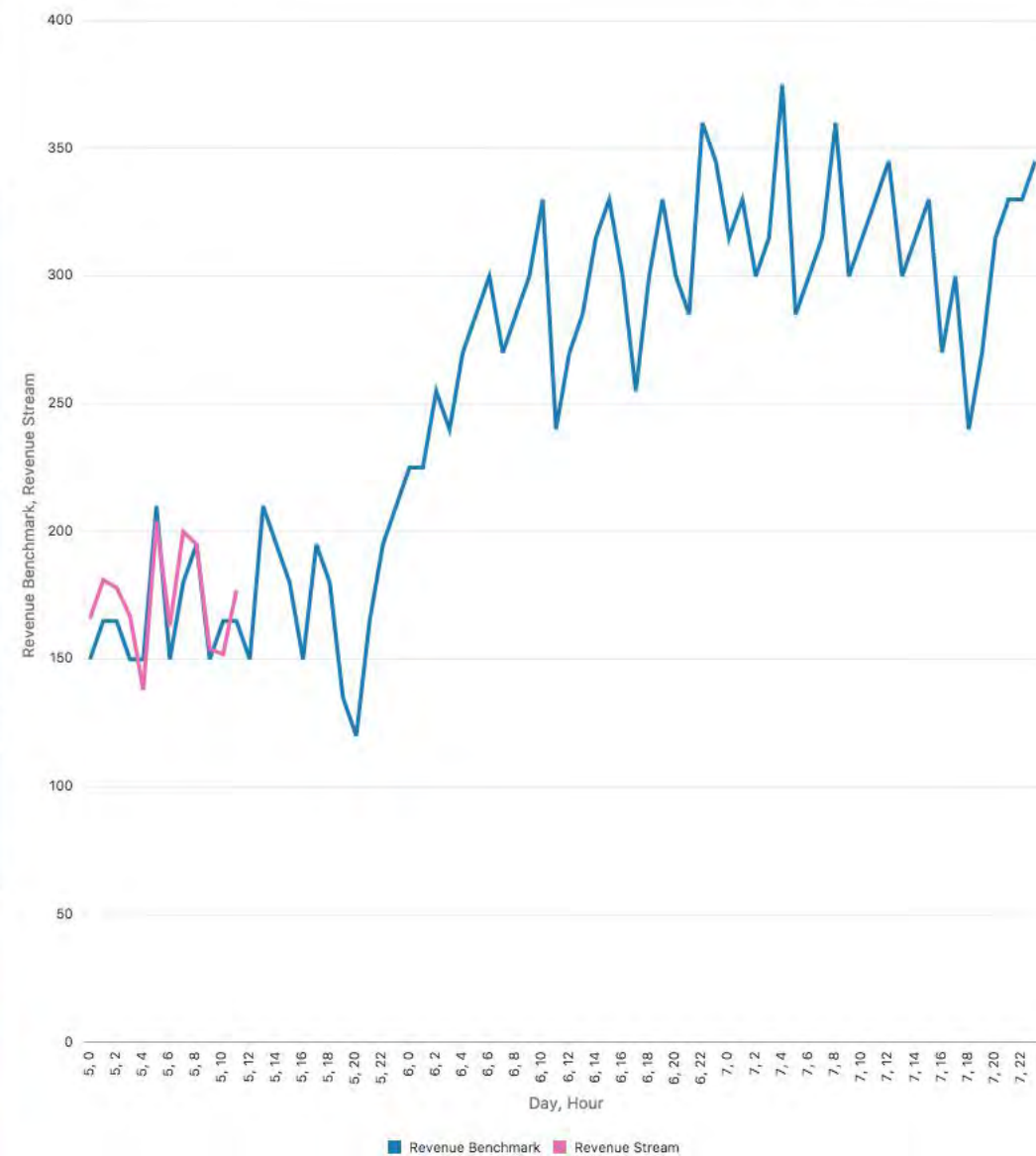
Compare Sales Stream to Benchmark (History)

- Day
- Country
- Network Errors
- Hour
- Revenue Benchmark
- Revenue Stream
- My Calculations
- Value Labels

Network Errors



Revenue Stream vs Benchmark



- ## Network Errors



The chart displays two data series over time. The Revenue Benchmark (blue line) starts at 150, dips to 120 in early May, then rises steadily to 330 by July 22. The Revenue Stream (pink line) starts at 150, peaks at 210 in mid-May, dips to 130 in early May, then rises to 340 by July 22.

Day, Hour	Revenue Benchmark	Revenue Stream
5, 0	150	150
5, 2	165	180
5, 4	150	140
5, 6	180	210
5, 8	195	200
5, 10	165	150
5, 12	165	175
5, 14	210	210
5, 16	150	170
5, 18	195	180
5, 20	120	130
5, 22	200	200
6, 0	225	225
6, 2	255	255
6, 4	270	270
6, 6	300	300
6, 8	270	270
6, 10	330	330
6, 12	240	240
6, 14	285	285
6, 16	330	330
6, 18	255	255
6, 20	330	330
6, 22	285	285
7, 0	360	360
7, 2	315	315
7, 4	375	375
7, 6	285	285
7, 8	320	320
7, 10	360	360
7, 12	345	345
7, 14	300	300
7, 16	330	330
7, 18	240	240
7, 20	315	315
7, 22	330	340

Search

KAFKA_DEMO_TABLE

Day

Country

Network Errors

Hour

Revenue Benchmark

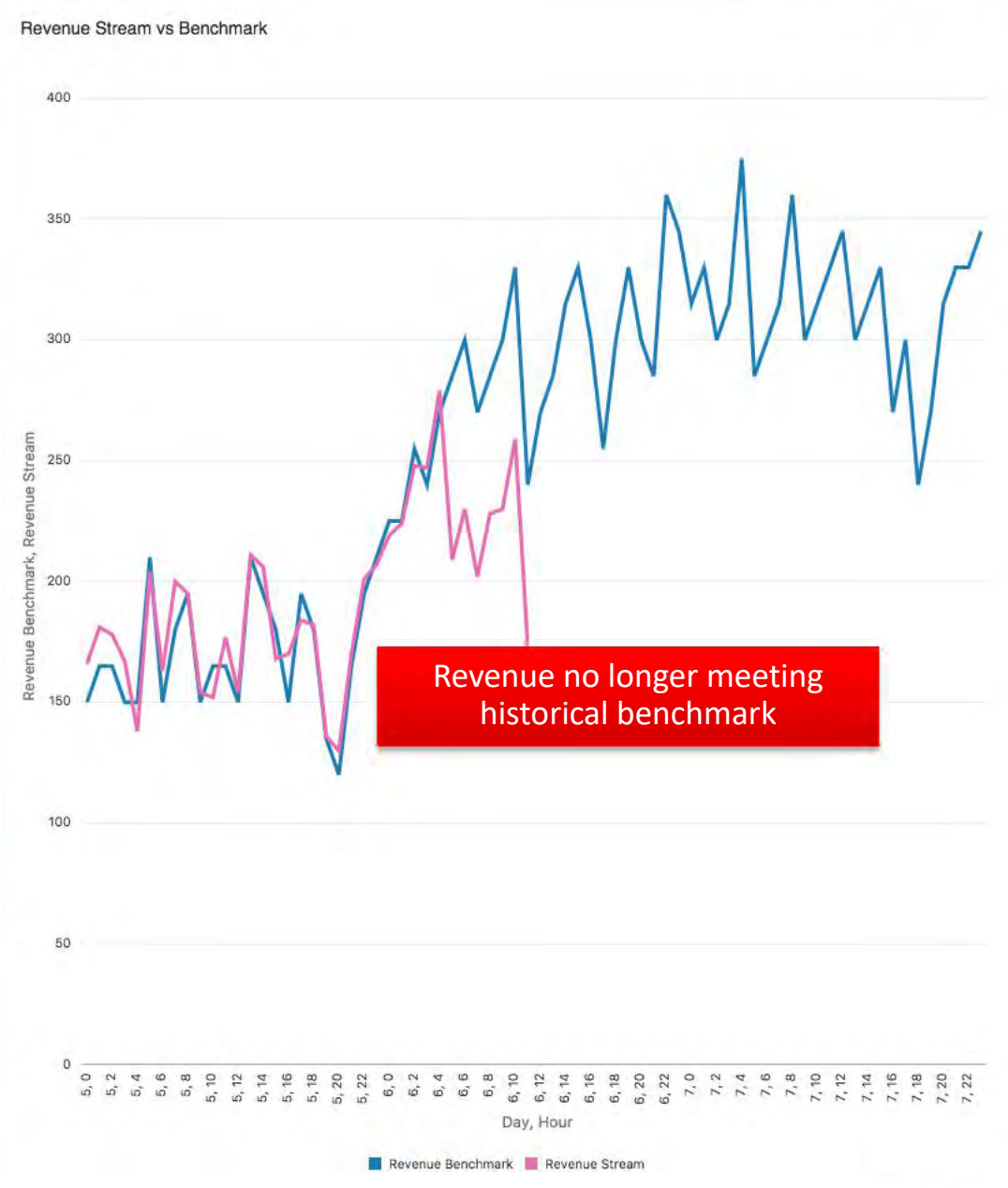
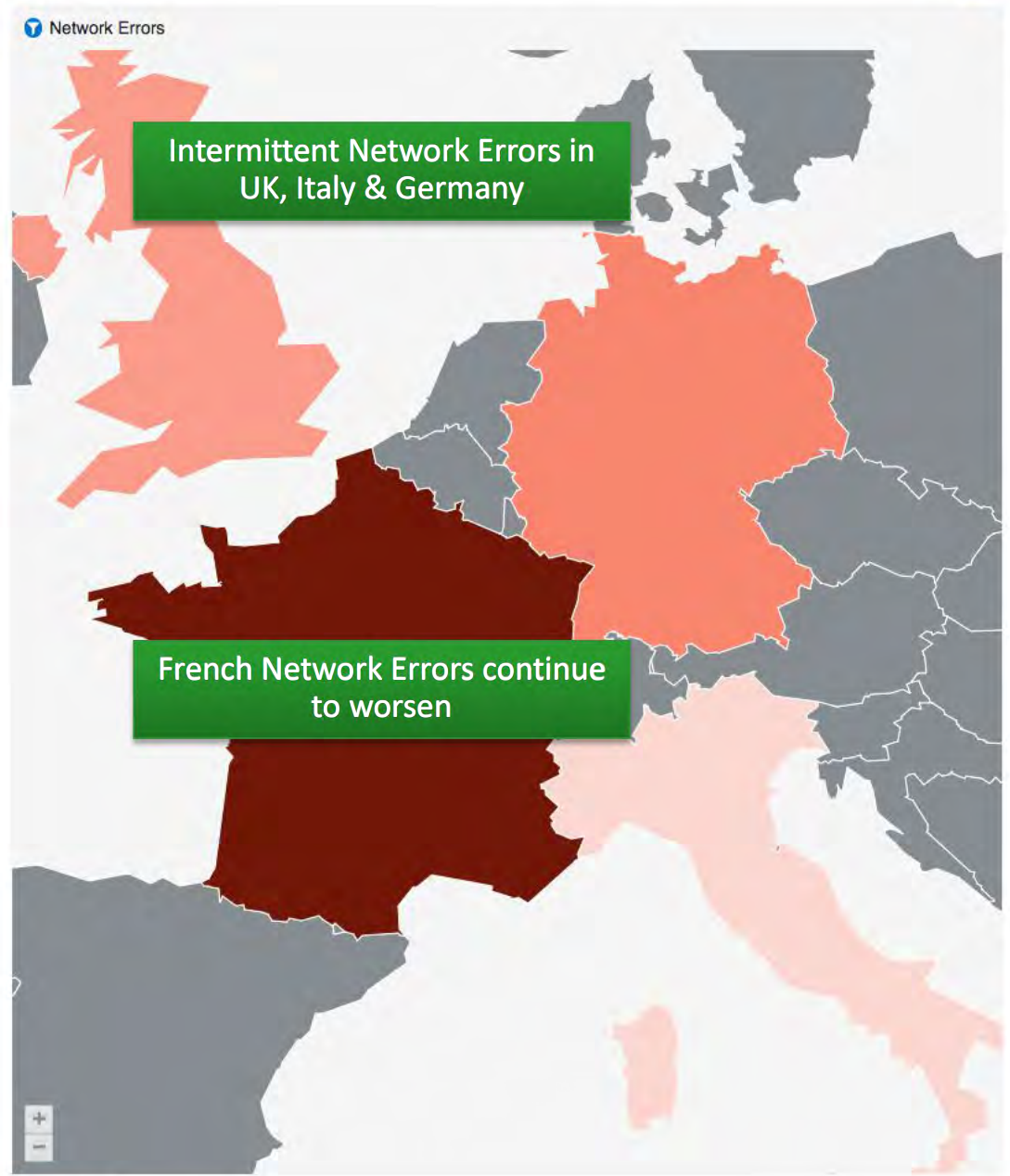
Revenue Stream

My Calculations

Value Labels

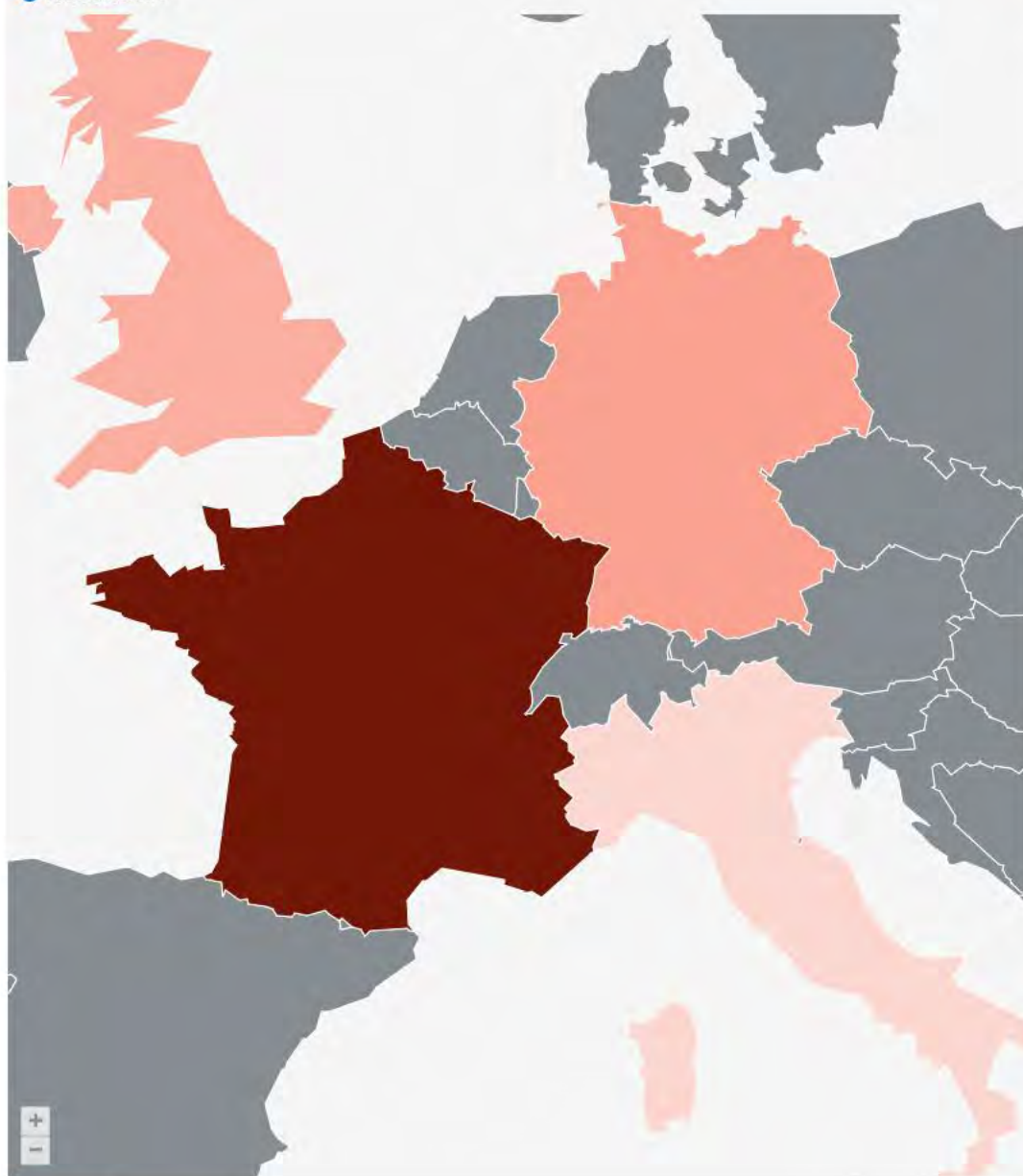
Streaming Demo
Project

Prepare Visualize Narrate

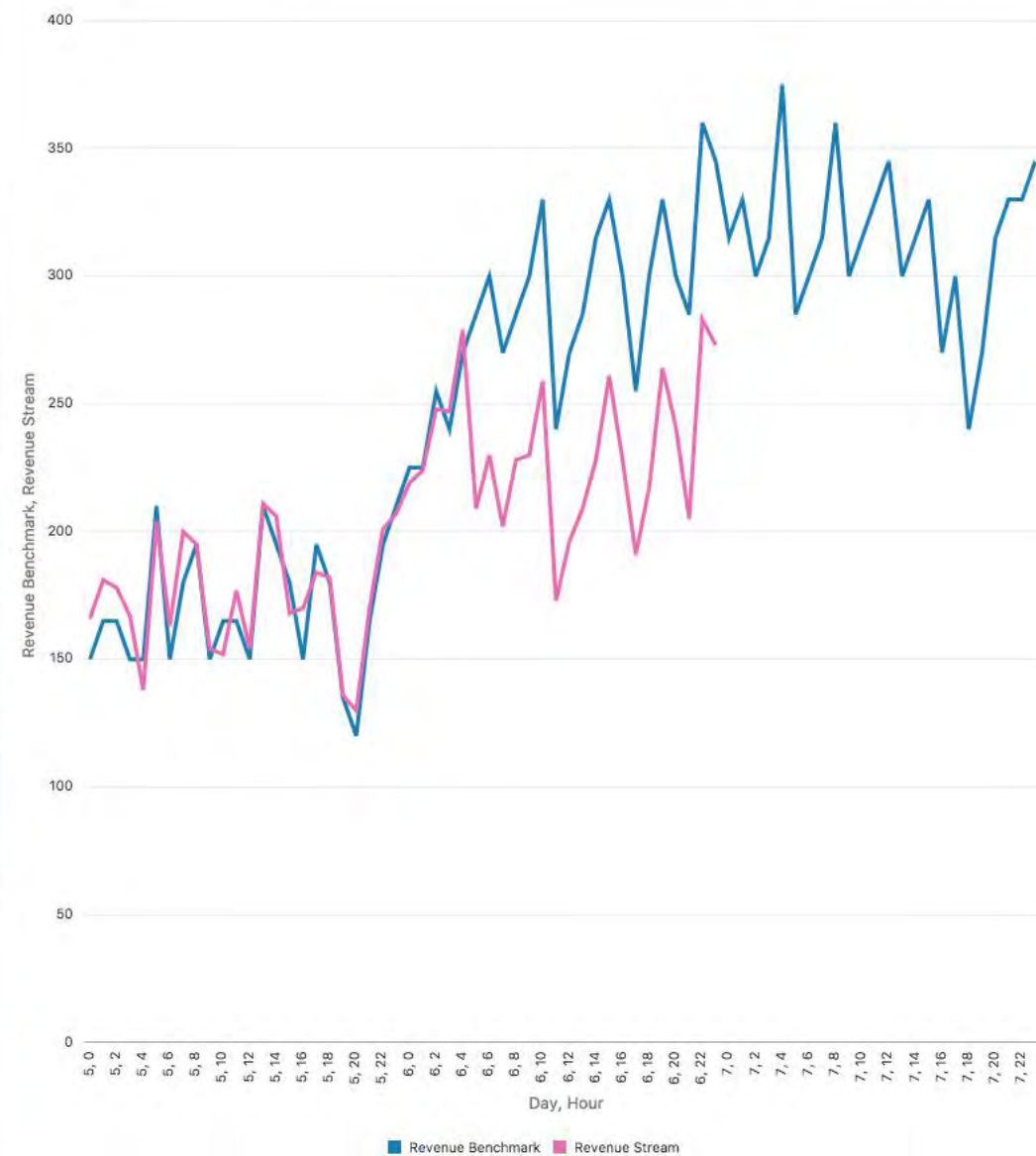


- Day
- Country
- Network Errors
- Hour
- Revenue Benchmark
- Revenue Stream
- My Calculations
- Value Labels

Network Errors



Revenue Stream vs Benchmark



- ## Network Errors



The chart displays two data series: 'Revenue Benchmark' (blue line) and 'Revenue Stream' (pink line). The x-axis represents time in 'Day, Hour' format, ranging from 5,0 to 7,22. The y-axis represents 'Revenue Benchmark, Revenue Stream' values from 0 to 400. The Revenue Benchmark line shows a general upward trend with some fluctuations, starting at 150 and ending at 340. The Revenue Stream line also shows an upward trend but with more volatility, starting at 165 and ending at 310. Both lines show a significant dip around May 20 and a peak around May 14 and June 4.

Day, Hour	Revenue Benchmark	Revenue Stream
5,0	150	165
5,2	165	180
5,4	150	160
5,6	210	140
5,8	150	200
5,10	165	150
5,12	150	175
5,14	210	210
5,16	150	170
5,18	195	185
5,20	125	135
5,22	200	200
6,0	225	225
6,2	255	250
6,4	280	275
6,6	270	210
6,8	295	230
6,10	330	260
6,12	240	175
6,14	290	210
6,16	330	260
6,18	255	190
6,20	330	265
6,22	285	210
7,0	360	285
7,2	315	245
7,4	375	310
7,6	285	220
7,8	360	280
7,10	300	235
7,12	345	240
7,14	300	
7,16	330	
7,18	270	
7,20	315	
7,22	340	310

Search

KAFKA_DEMO_TABLE

Day

Country

Network Errors

Hour

Revenue Benchmark

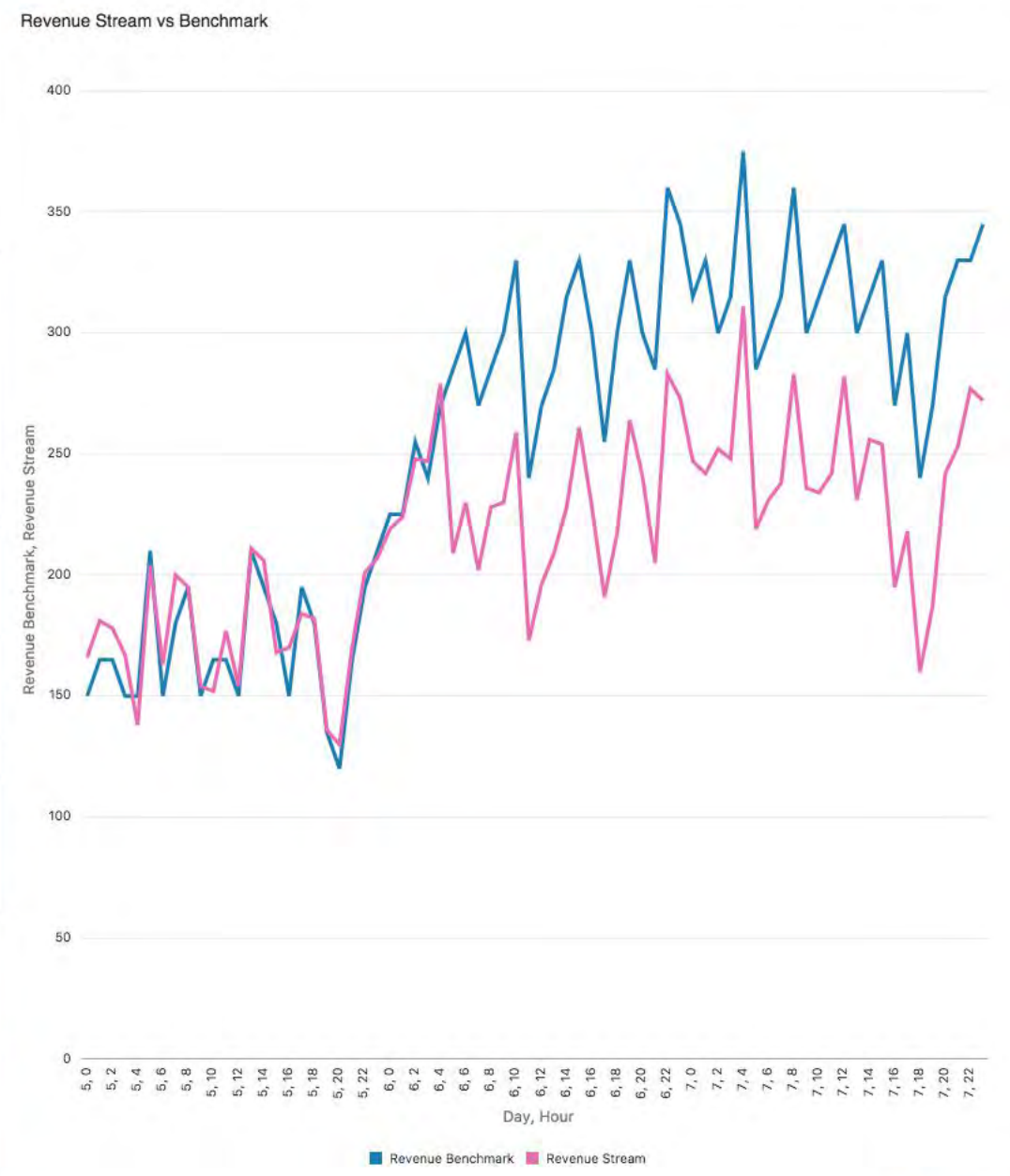
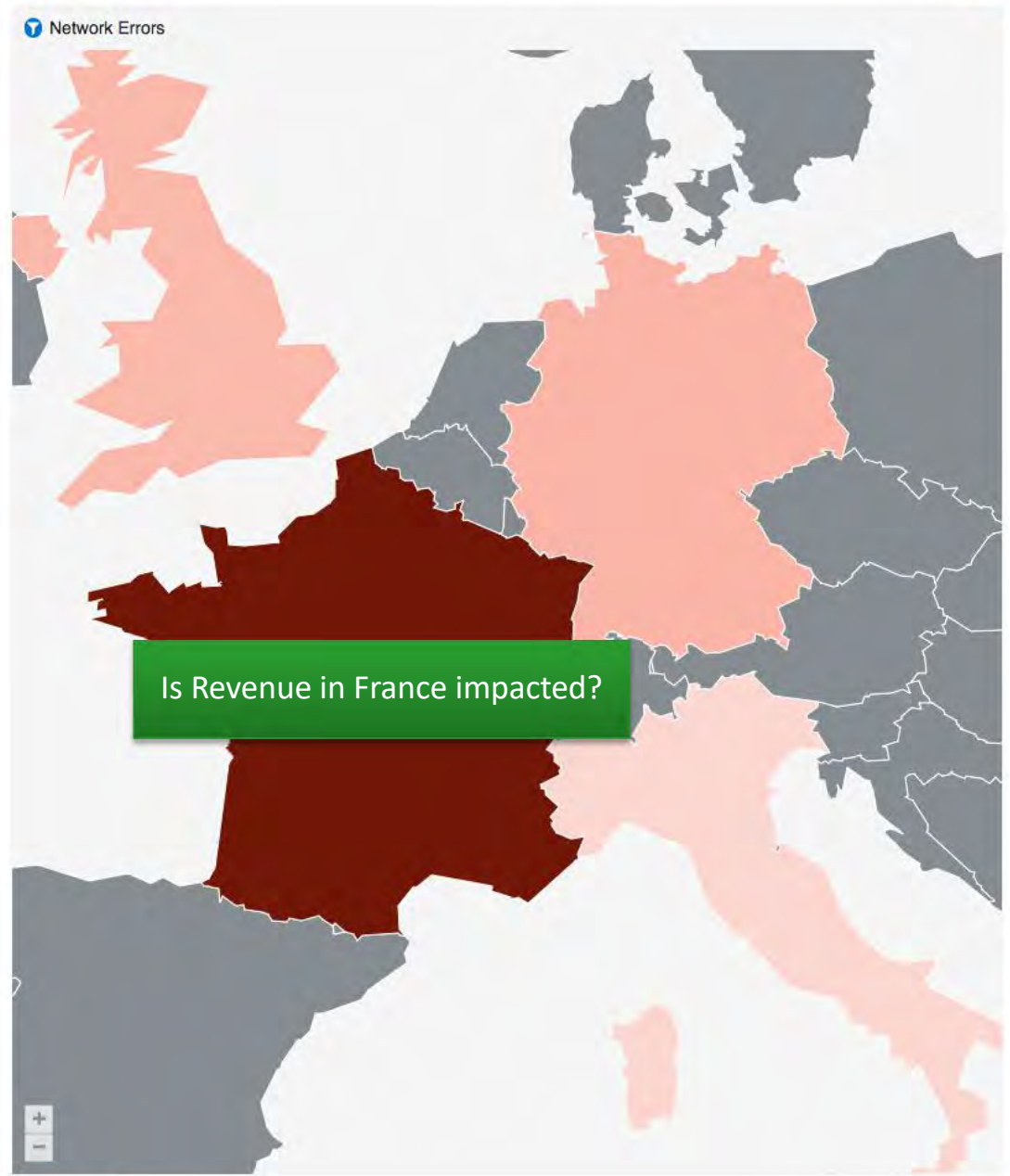
Revenue Stream

My Calculations

Value Labels

Streaming Demo
Project

Prepare Visualize Narrate



KAFKA_DEMO_TABLE

- Day
- Country
- Network Errors
- Hour
- Revenue Benchmark
- Revenue Stream
- My Calculations
- Value Labels

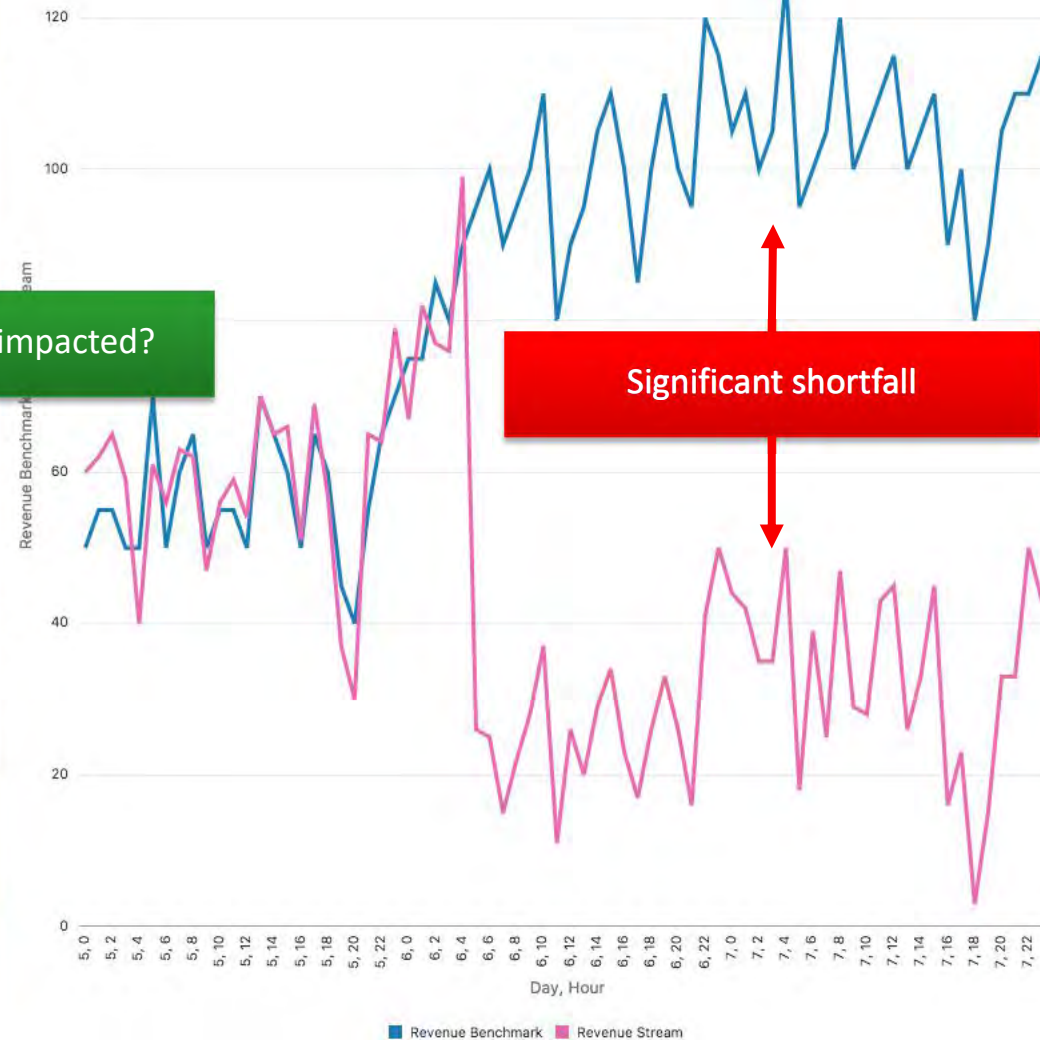
Network Errors



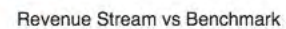
Is Revenue in UK impacted?

Revenue Stream vs Benchmark

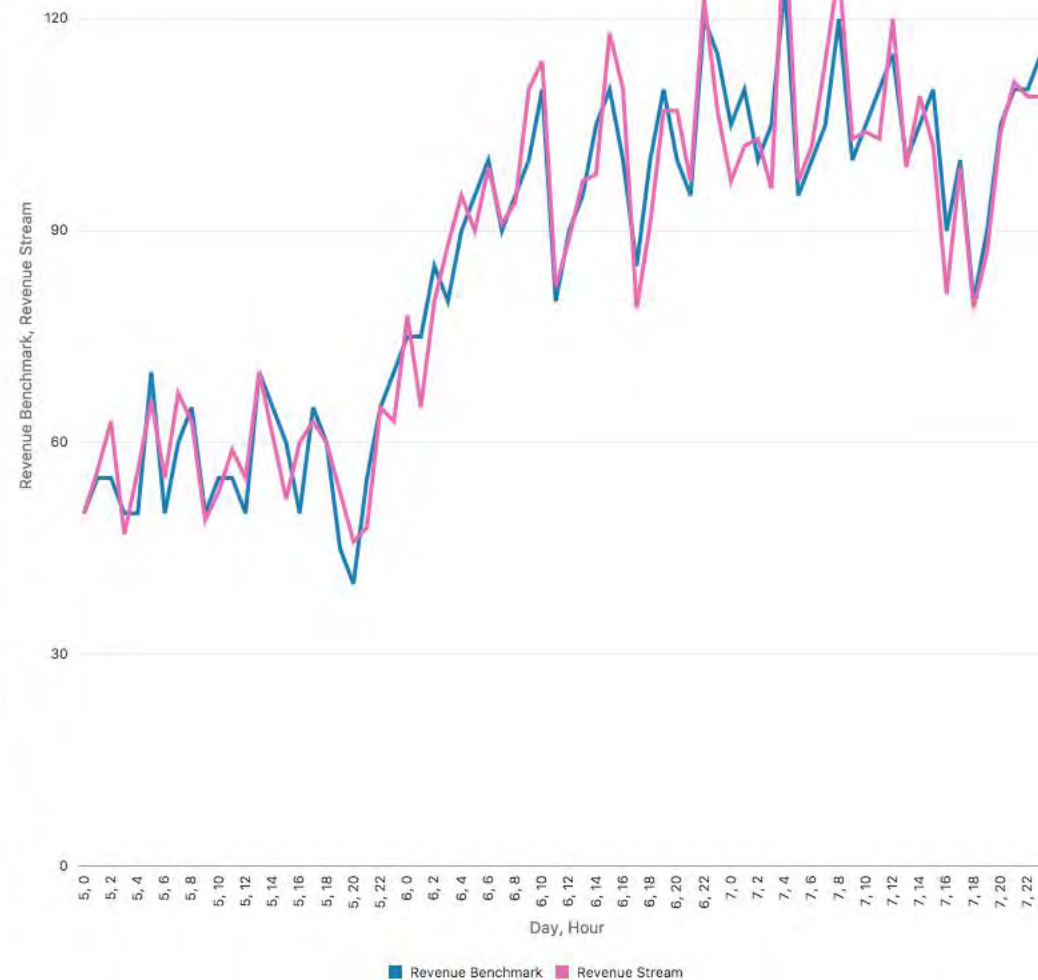
France



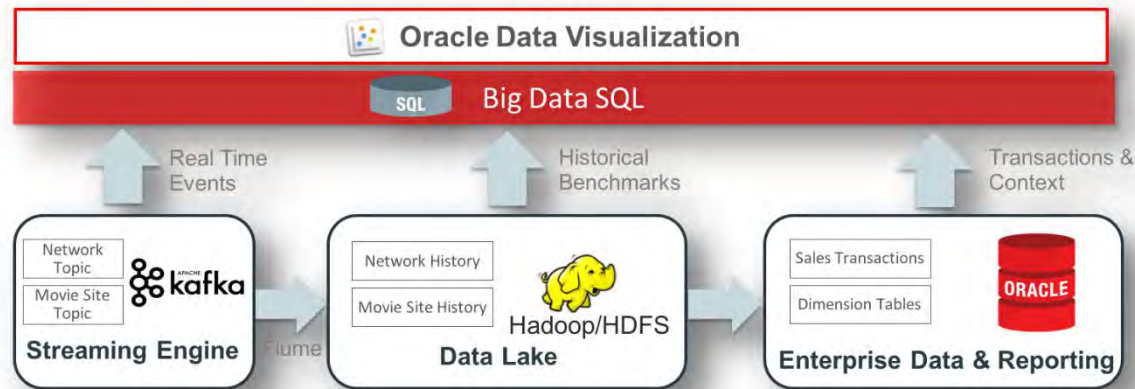
- ### Network Errors



United Kingdom

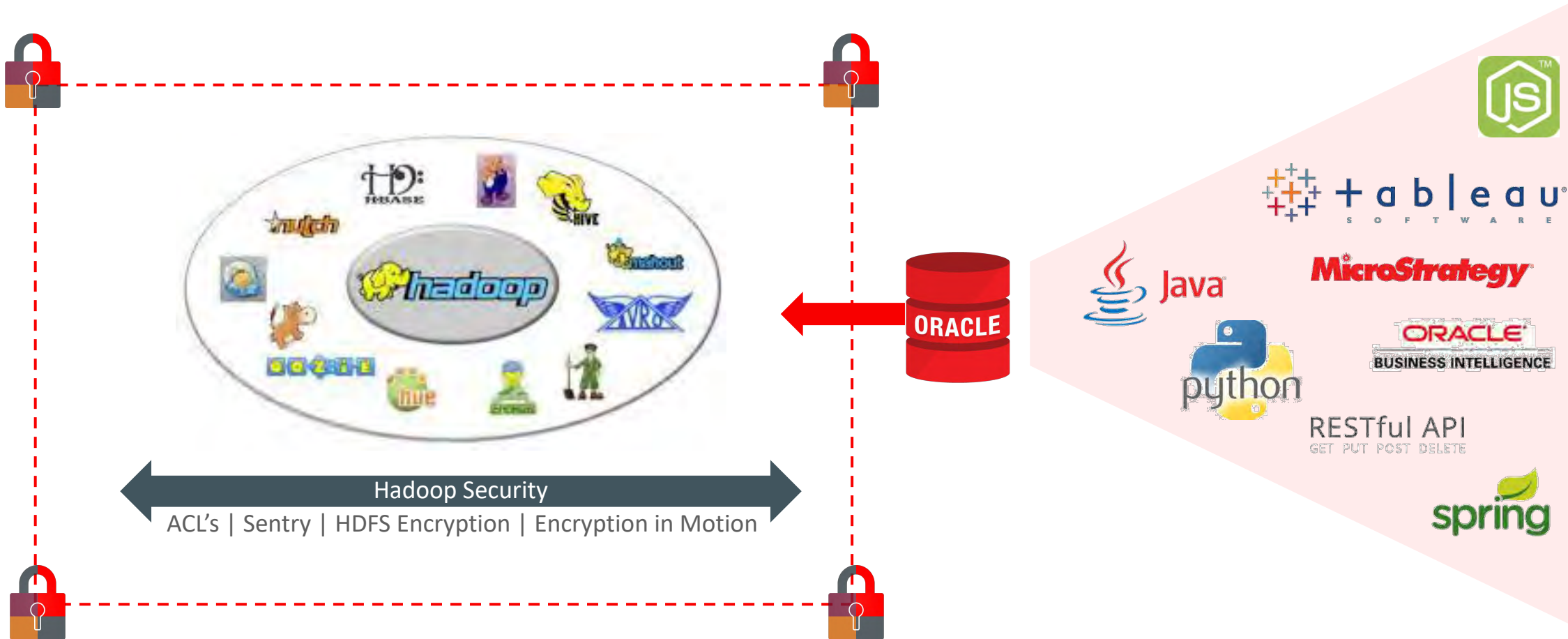


Insights Achieved with Simplicity

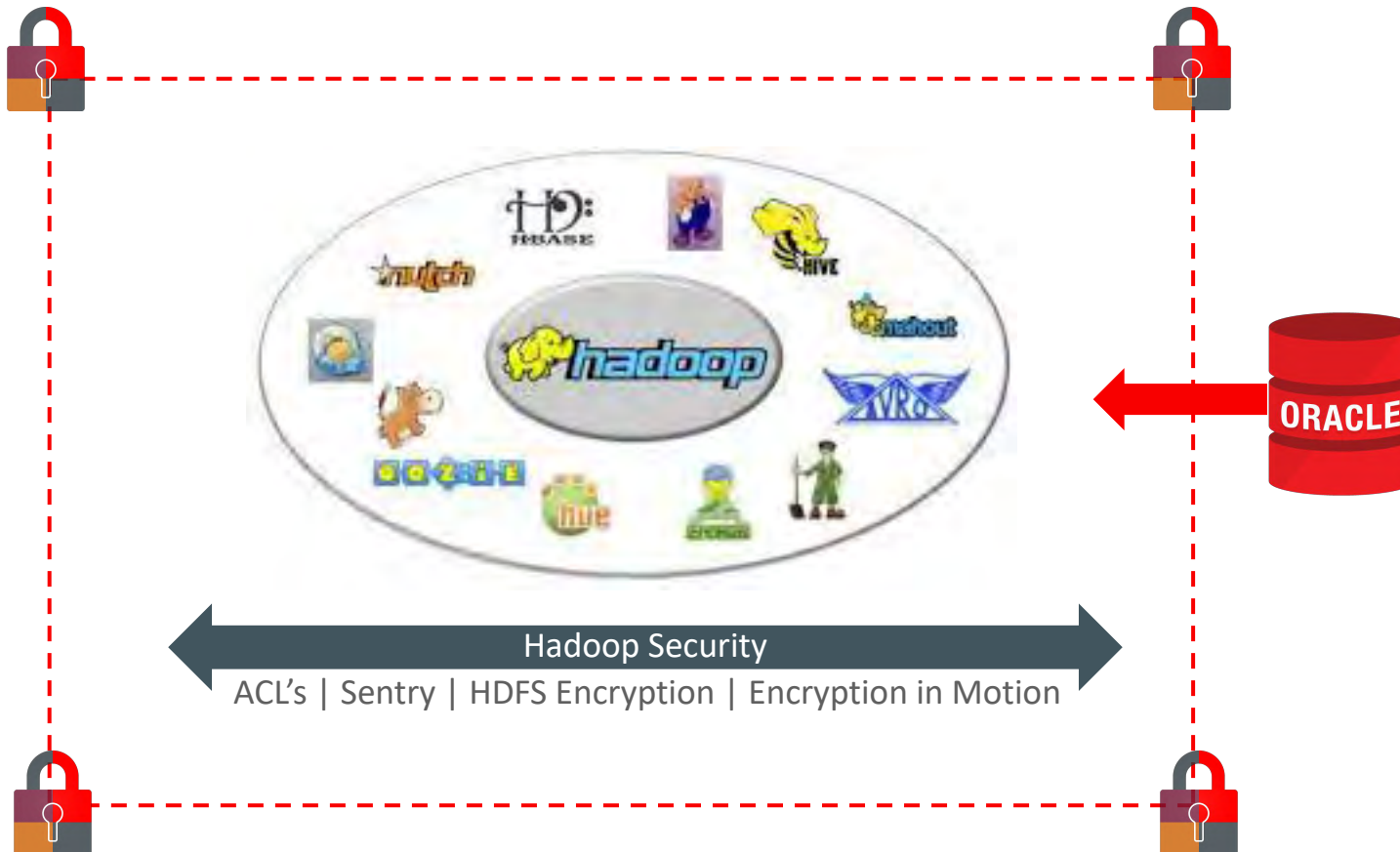


- Easily blend real time streams with history, benchmarks and context
 - Are we running at peak performance?
 - What is the opportunity cost of our current network latency?
- Any application realizes benefit
 - Use Oracle SQL and APIs over all data
- Ensure data is secure
 - Leverage Oracle advanced security

Big Data SQL Security Features



Big Data SQL Security Features



- Same security models apply to a wider range of data stores
- Advanced features such as data redaction can now be applied enabling joins between disparate sources
- Oracle security layers on top of existing Hadoop functionality

Big Data SQL Summary



Easily access any data
across big data stores



Provide a unified security
model across the sources



Analyze all data using
Oracle's rich SQL dialect



Fast performance using Big
Data SQL Smart Scan

More Information



- OTN: Big Data Lite Virtual Machine (a free sandbox environment to get started):
<http://www.oracle.com/technetwork/database/bigdata-appliance/oracle-bigdatalite-2104726.html>
- Oracle.com:
<https://www.oracle.com/big-data/index.html>
- Blog: (technical examples and tips):
<https://blogs.oracle.com/datawarehousing/>