

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



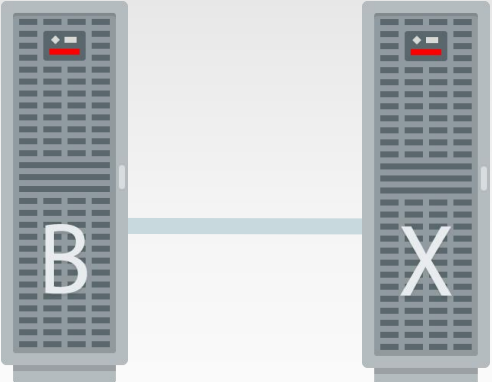
Analyze all data using
Oracle's rich SQL dialect



Fast performance using Big
Data SQL Smart Scan

Flexible Deployment Options

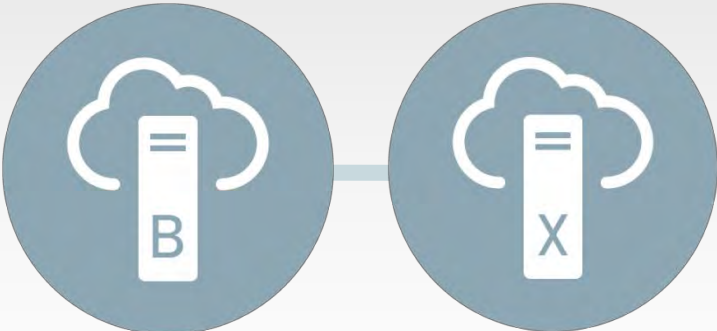
Engineered Systems



cloudera

12^c

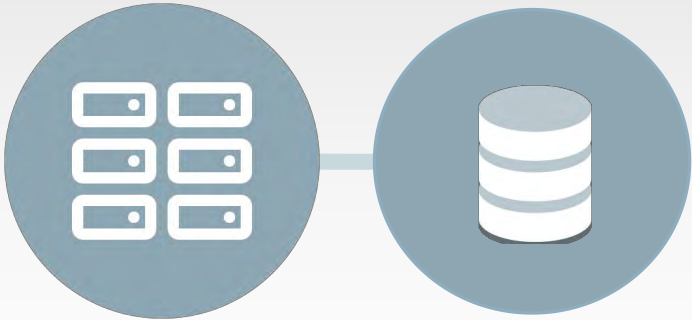
Oracle Cloud



cloudera

12^c

Commodity

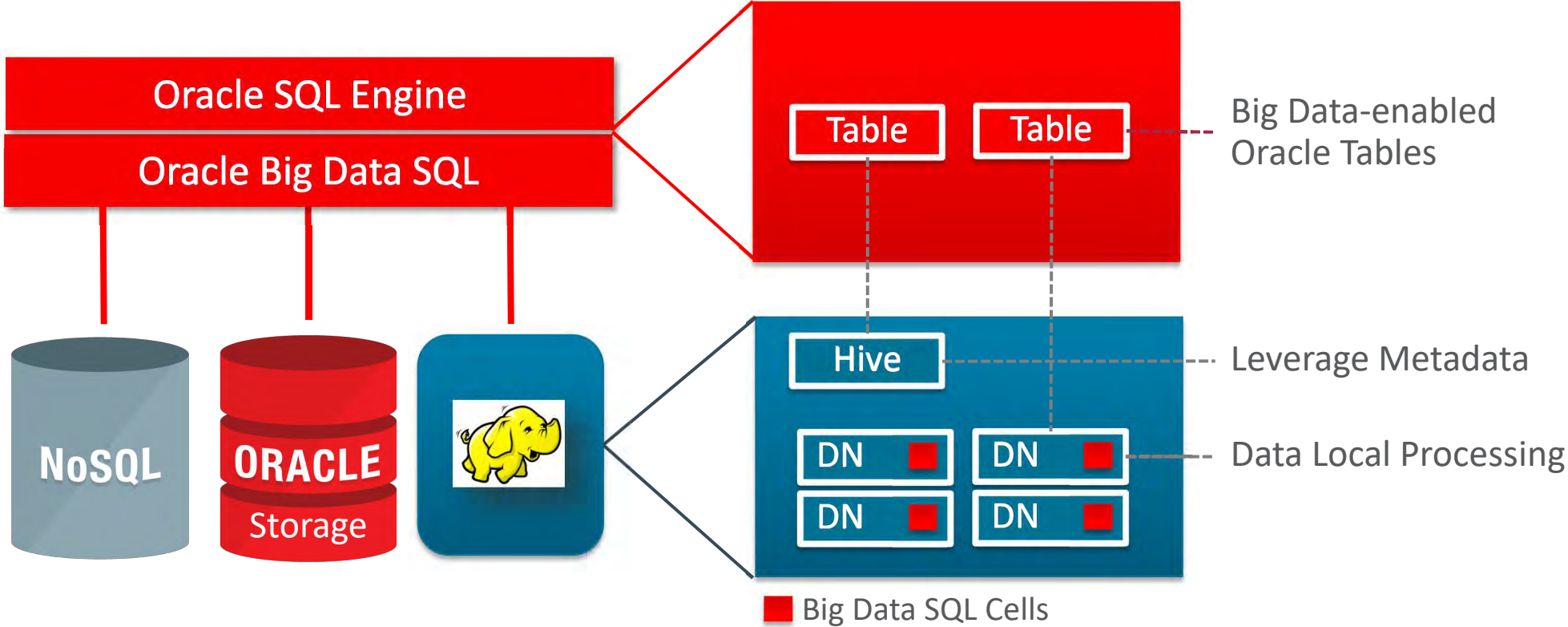


cloudera

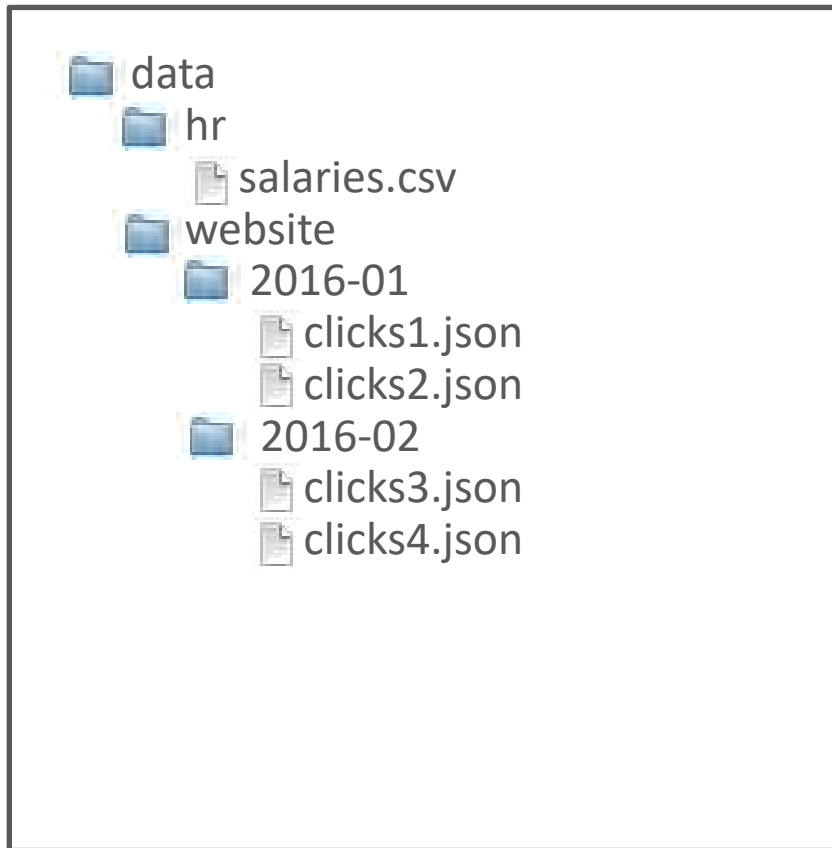


12^c

Big Data SQL Architecture

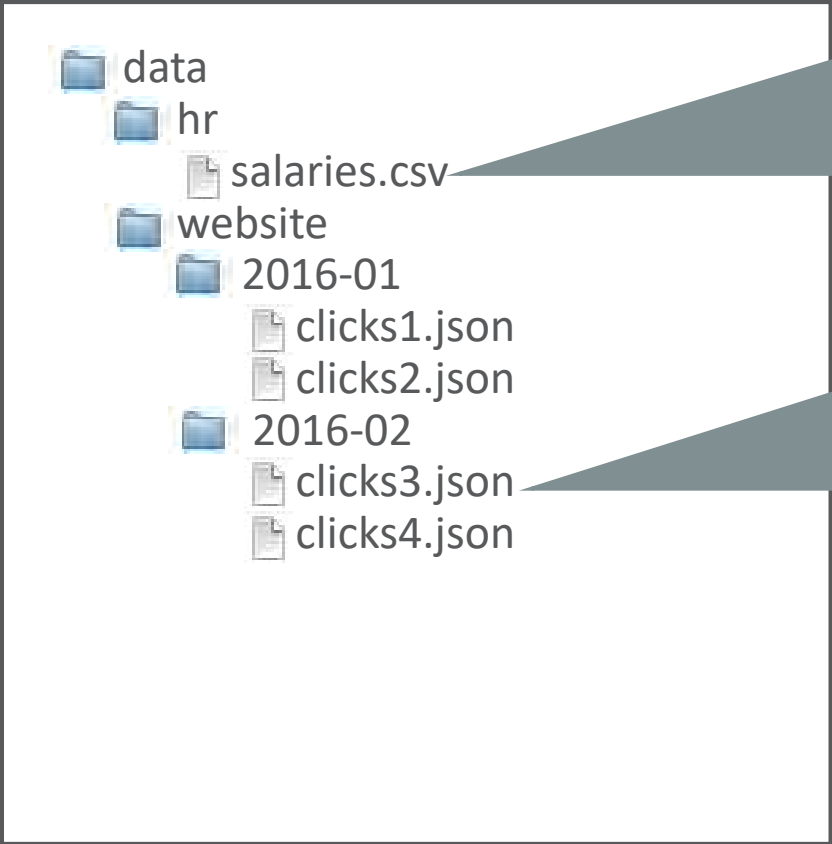


How Data is Stored in HDFS



- 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

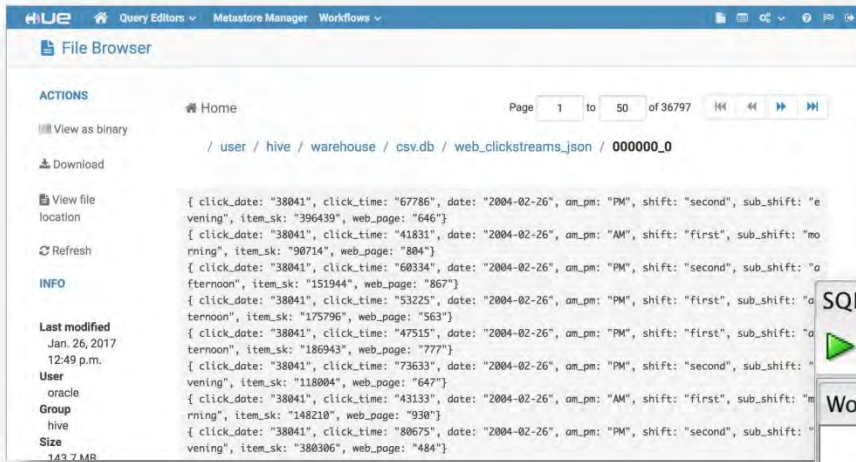
How Data is Stored in HDFS



Hanks,Spielberg,1000000
Spielberg,Cameron,2500000
Cameron,Oprah,125000
Oprah,Boss,54000000

```
{"custId":1354924,"movieId":1948,"genreId":9,"time":"2012-07-01:00:00:22"}  
{"custId":1083711,"movieId":null,"genreId":null,"time":"2012-07-01:00:00:26"}  
{"custId":1234182,"movieId":11547,"genreId":44,"time":"2012-07-01:00:00:32"}  
{"custId":1010220,"movieId":11547,"genreId":44,"time":"2012-07-01:00:00:42"}
```


Demonstration: Accessing Data in HDFS



The screenshot shows the Oracle HIVE File Browser interface. The path is `/user/hive/warehouse/csv.db/web_clickstreams_json/000000_0`. The main area displays a list of JSON files with their metadata, including click_date, click_time, date, am_pm, shift, sub_shift, item_sk, and web_page. The files are sorted by date and time.

File Name	click_date	click_time	date	am_pm	shift	sub_shift	item_sk	web_page
000000_0	"38041"	"67786"	"2004-02-26"	"PM"	"second"	"evening"	"396439"	"646"
000000_1	"38041"	"41831"	"2004-02-26"	"AM"	"first"	"morning"	"90714"	"804"
000000_2	"38041"	"60334"	"2004-02-26"	"PM"	"second"	"afternoon"	"151944"	"867"
000000_3	"38041"	"53225"	"2004-02-26"	"PM"	"first"	"afternoon"	"175796"	"563"
000000_4	"38041"	"47515"	"2004-02-26"	"PM"	"first"	"afternoon"	"186943"	"777"
000000_5	"38041"	"73633"	"2004-02-26"	"PM"	"second"	"evening"	"118004"	"647"
000000_6	"38041"	"43133"	"2004-02-26"	"AM"	"first"	"morning"	"148210"	"930"
000000_7	"38041"	"80675"	"2004-02-26"	"PM"	"second"	"evening"	"380306"	"484"

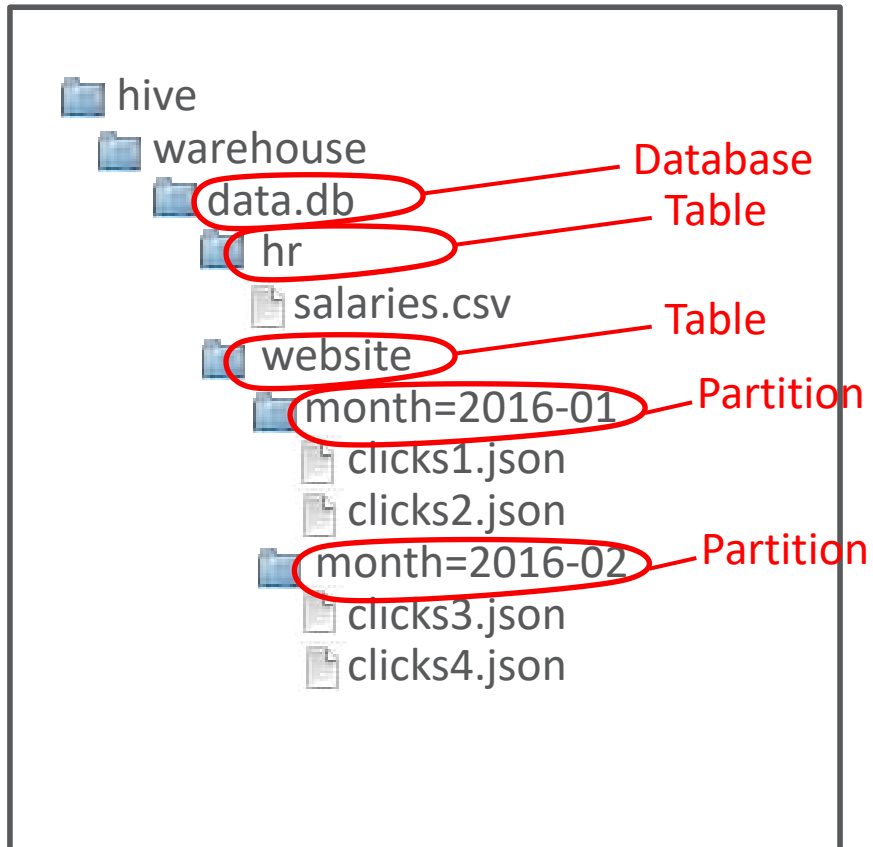


The screenshot shows the Oracle SQL Worksheet interface. The main area displays a SQL statement to create an external table:

```
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 ;
```

The interface also shows a toolbar with various icons for execution and editing, and a 'Query Builder' tab.

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

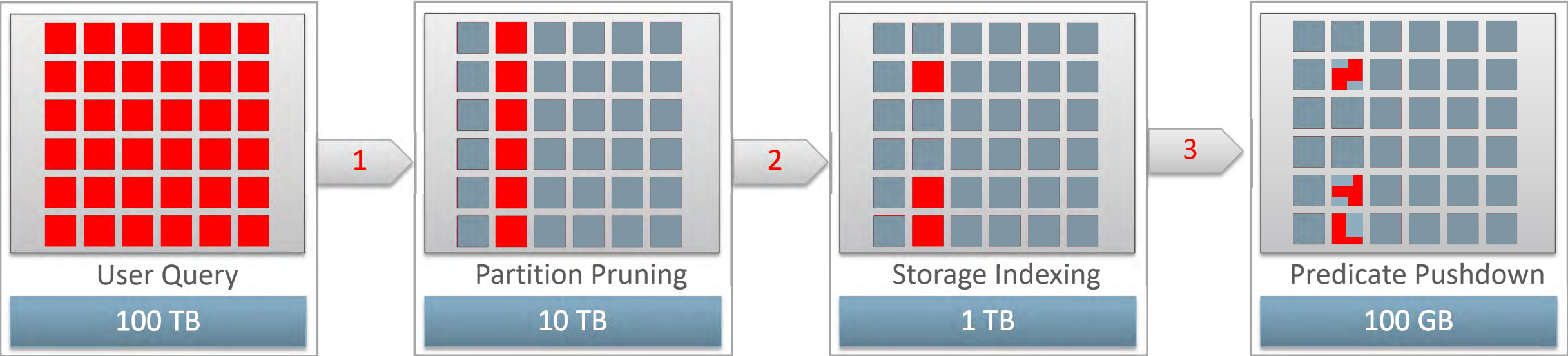
The screenshot shows the Hive Metastore Manager interface. On the left, a list of tables is visible, including 'customer', 'customer_address', 'customer_demographics', 'date_dim', 'emp_hbase', 'household_demographics', 'income_band', 'inventory', 'item', 'item_marketprices', 'mv_store_sales_cat_state', 'mv_store_sales_qtr_class', 'mv_store_sales_qtr_state_class2', 'product_reviews', 'promotion', 'reason', 'ship_mode', 'store', 'store_returns', and 'store_sales'. The main pane displays the definition for the 'createtab_stmt' table, which is a CREATE TABLE statement with various columns and their data types, including decimal, ROW FORMAT SERDE, STORED AS INPUTFORMAT, OUTPUTFORMAT, and LOCATION.

The screenshot shows the Oracle Connections window. It displays a tree view of connections and tables. The 'hive' connection is expanded, showing a list of tables including 'customer', 'customer_address', 'customer_demographics', 'date_dim', 'household_demographics', 'income_band', 'inventory', and 'item'. A context menu is open over the 'customer' table, with the option 'Use in Oracle Big Data SQL...' highlighted.



Big Data SQL Performance Features

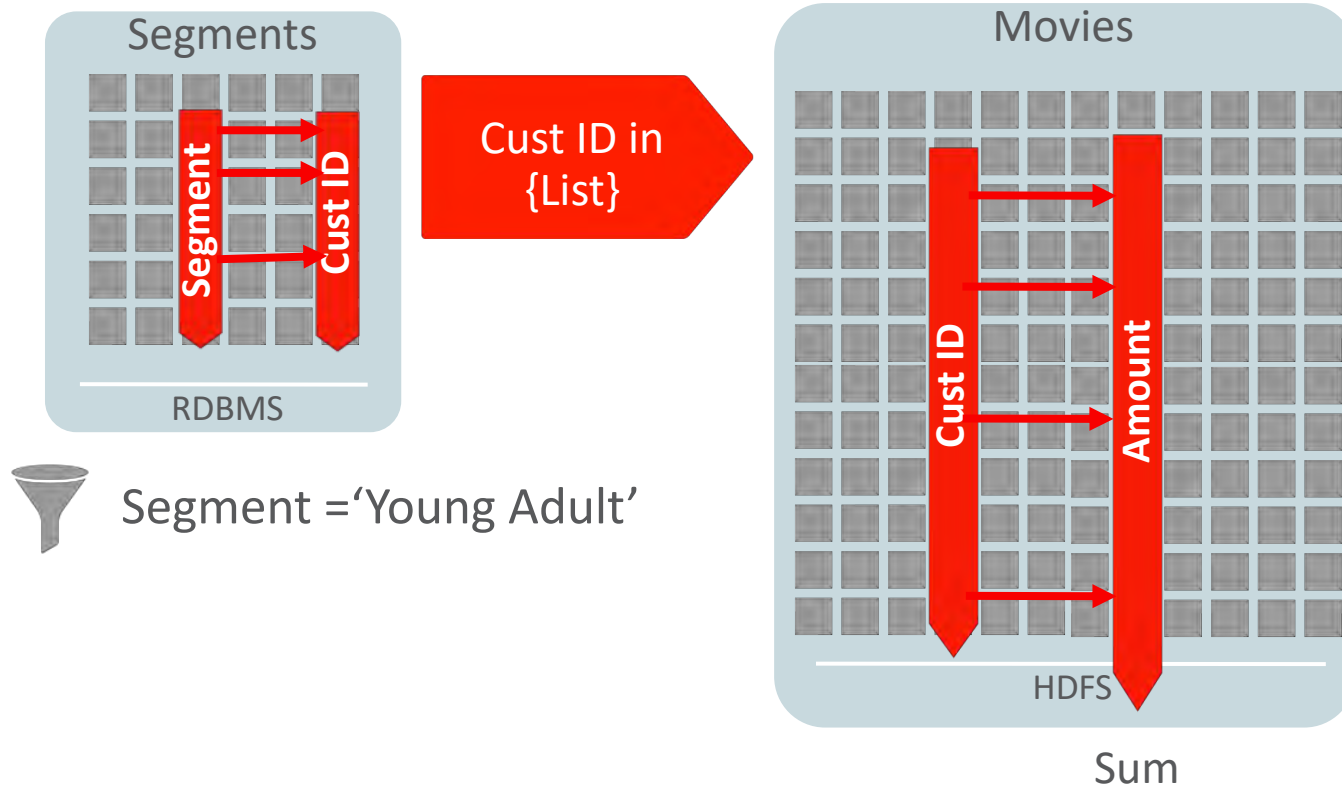
Compound IO Reduction thru Smart Scans



Big Data SQL Performance Features

Smart Scan – Execute Joins as Bloom Filters on Hadoop Nodes

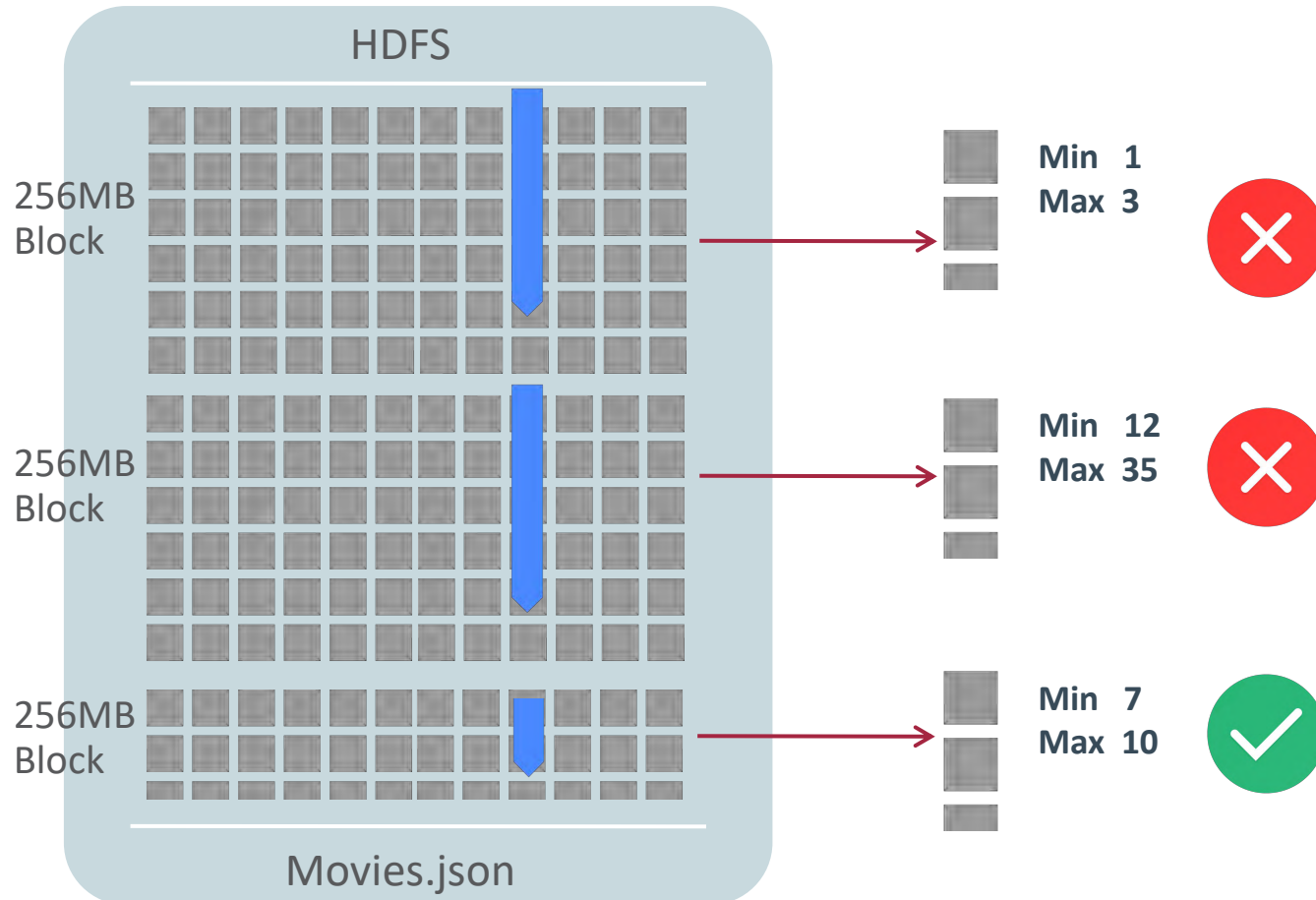
Example: Total movie sales for customer segment



- 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)

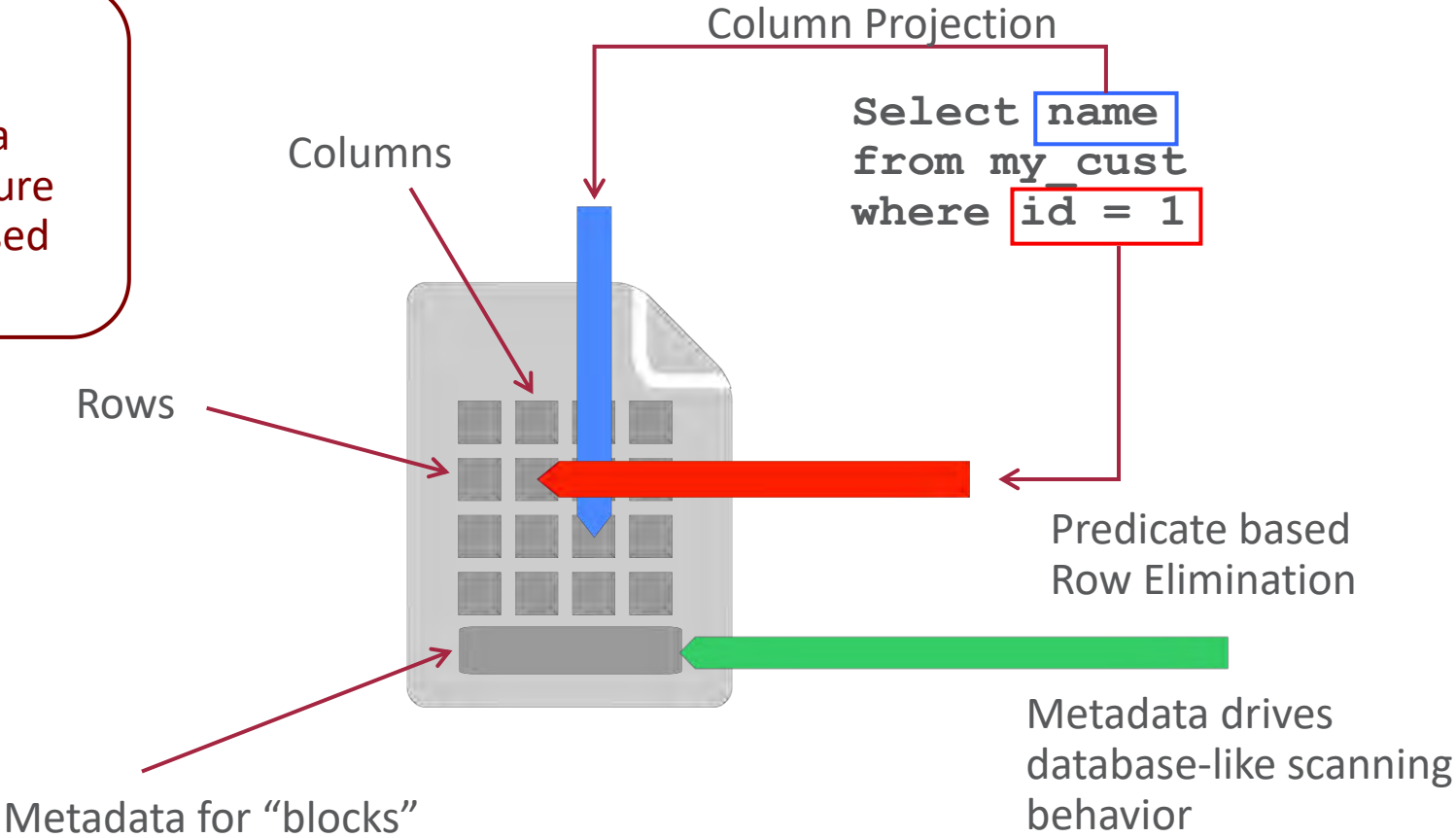


- Storage index provides query speed-up through transparent IO elimination of HDFS Blocks. It's a *negative index*
- Min / max value is recorded for columns included in a storage index (max # of columns = 32)
- Storage index provides partition pruning like performance for un-modeled data sets

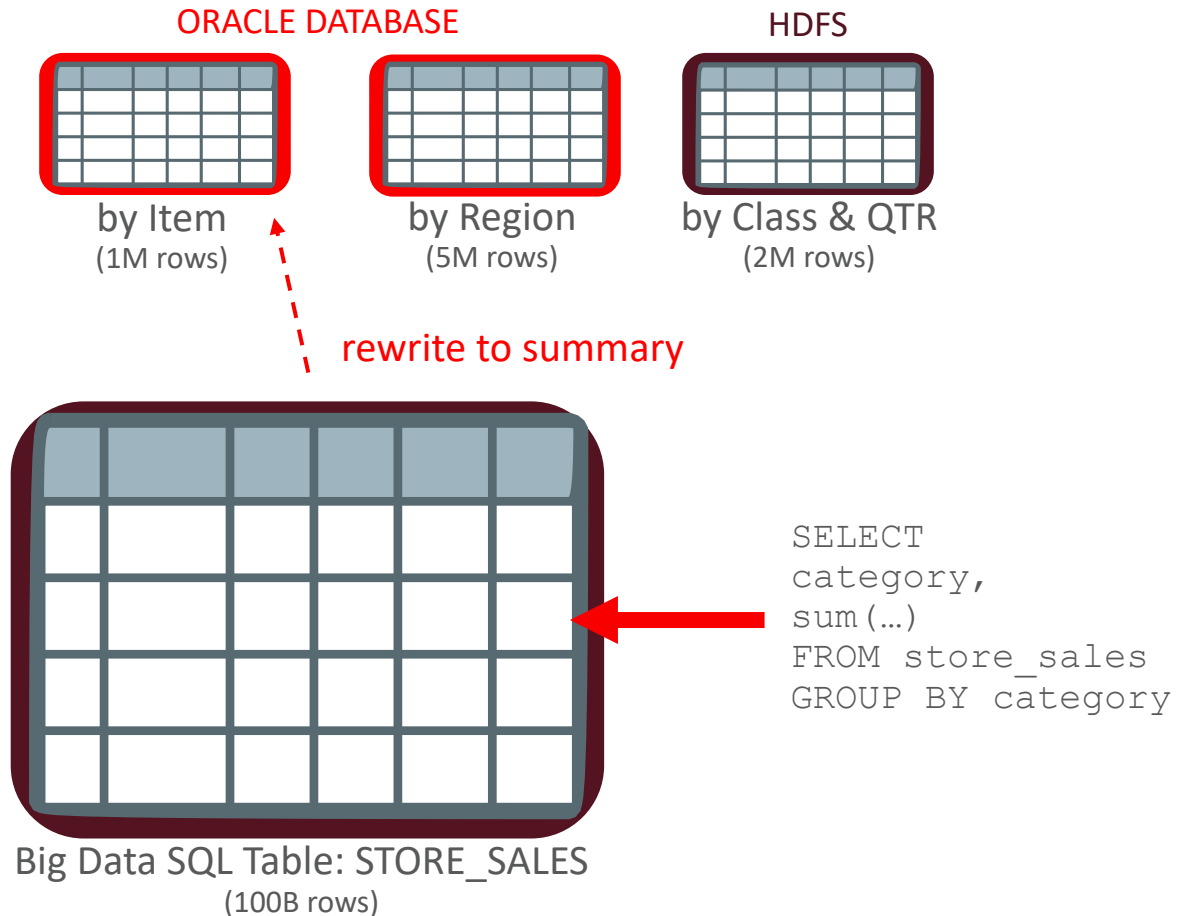
Parquet: Columnar Database File in HDFS

Big Data SQL Pushes Predicates to Parquet

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



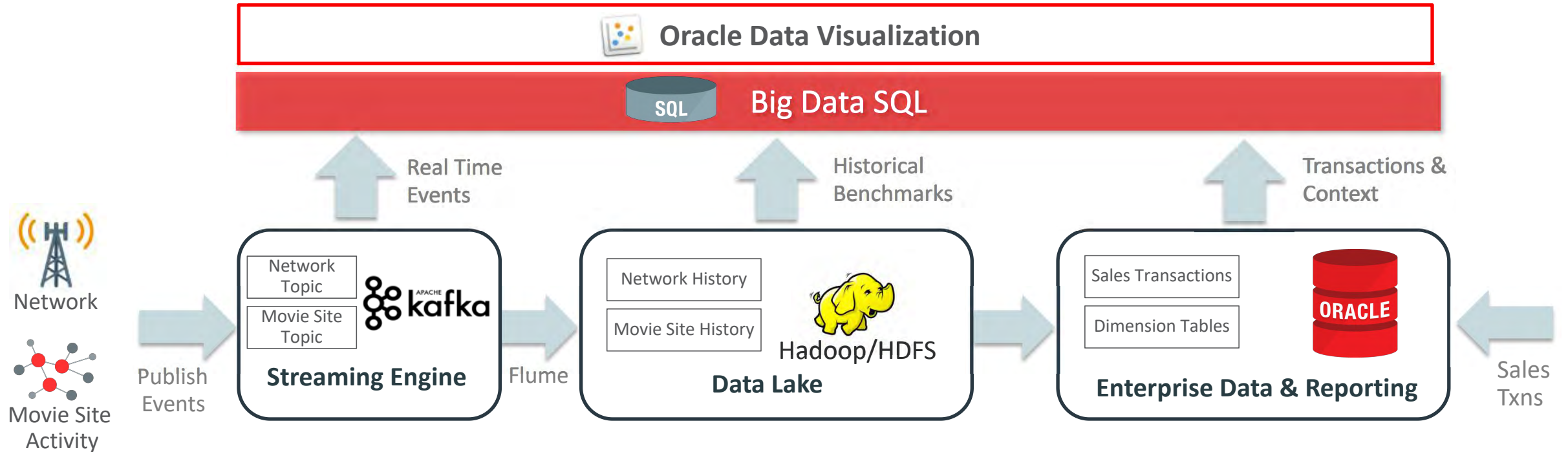
Enhance Performance with Automatic Query Rewrite



- **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

Demonstration Scenario

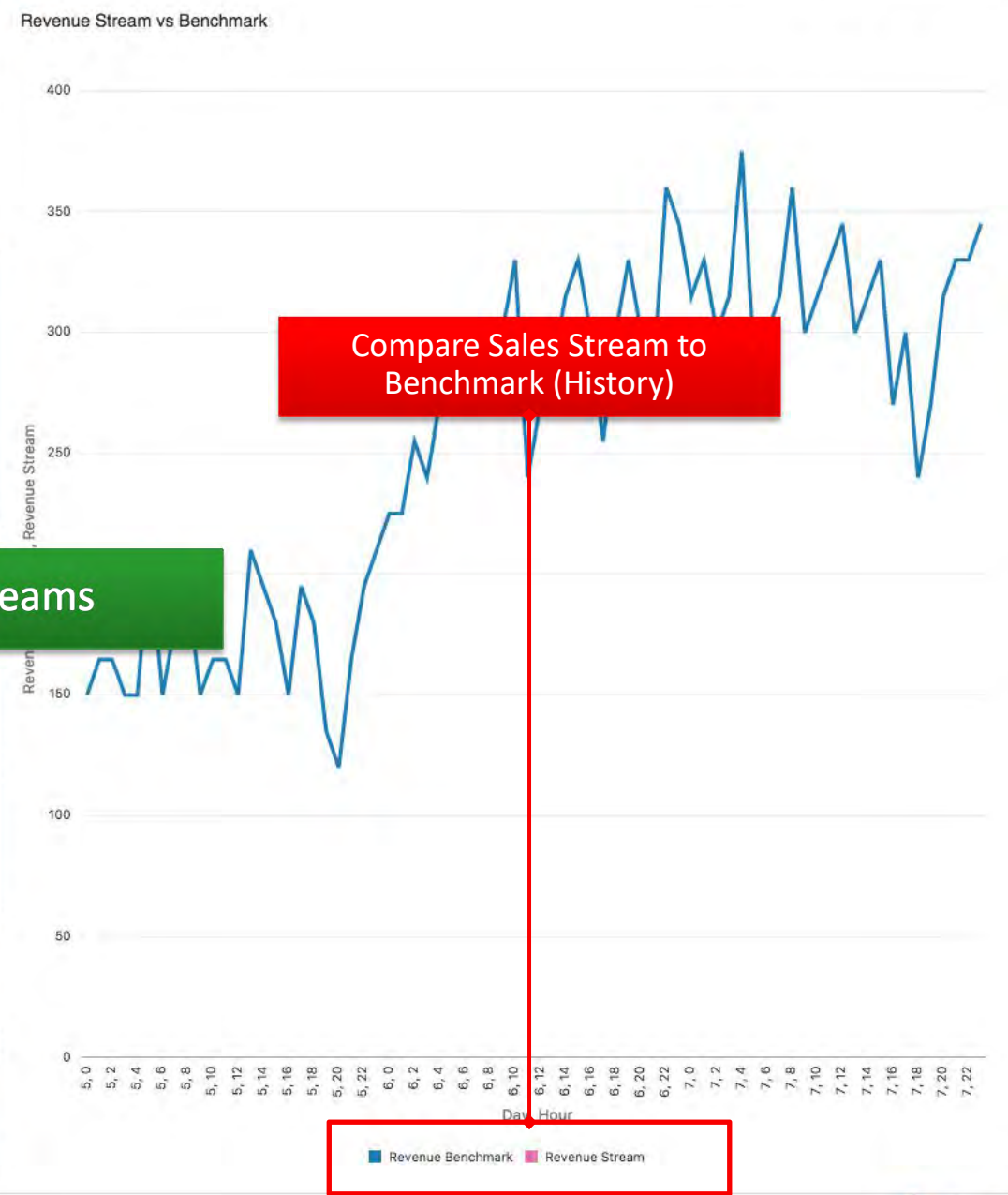
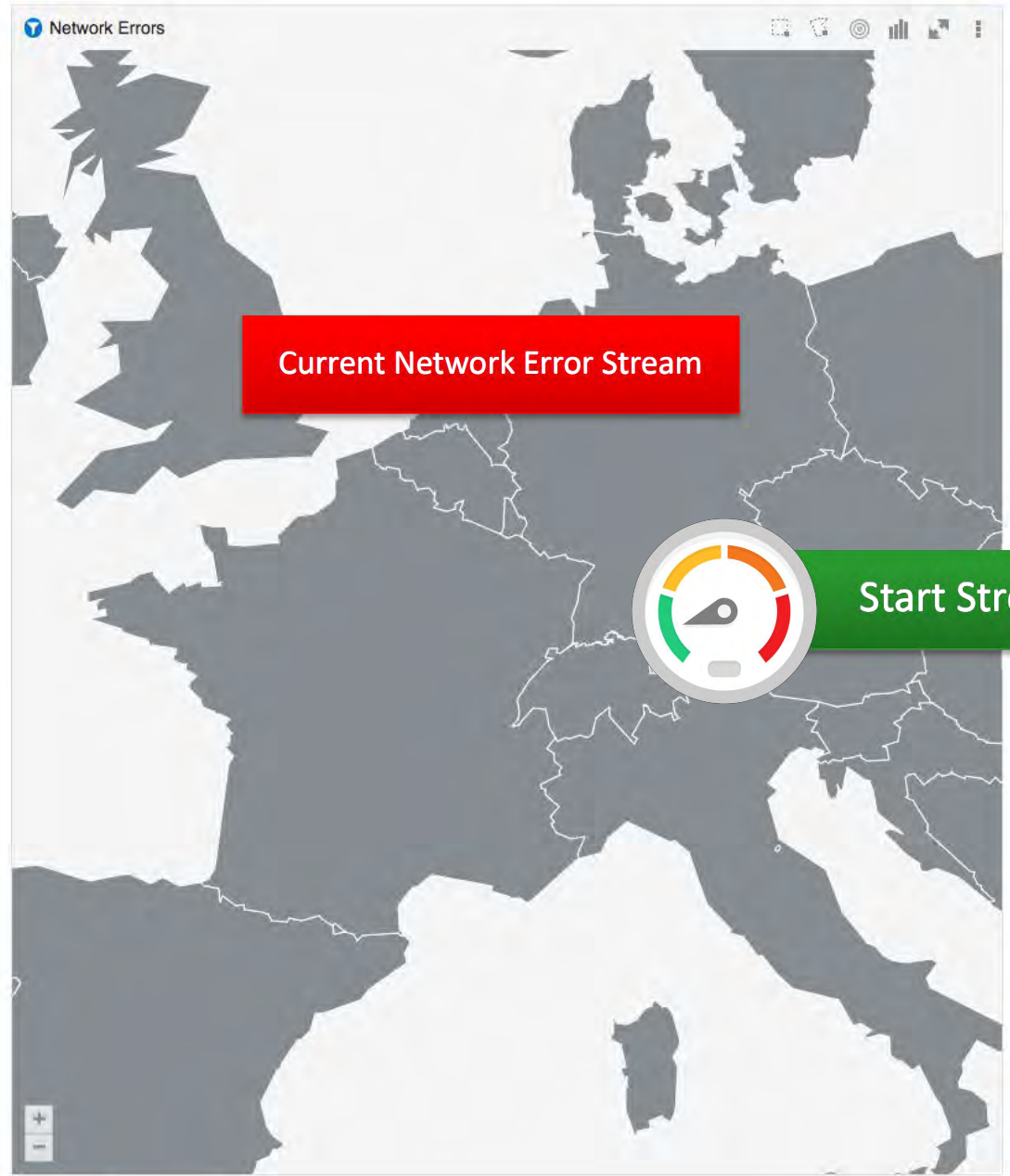
Analyzing Real-time Streams



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

Streaming Demo Project

Prepare Visualize Narrate



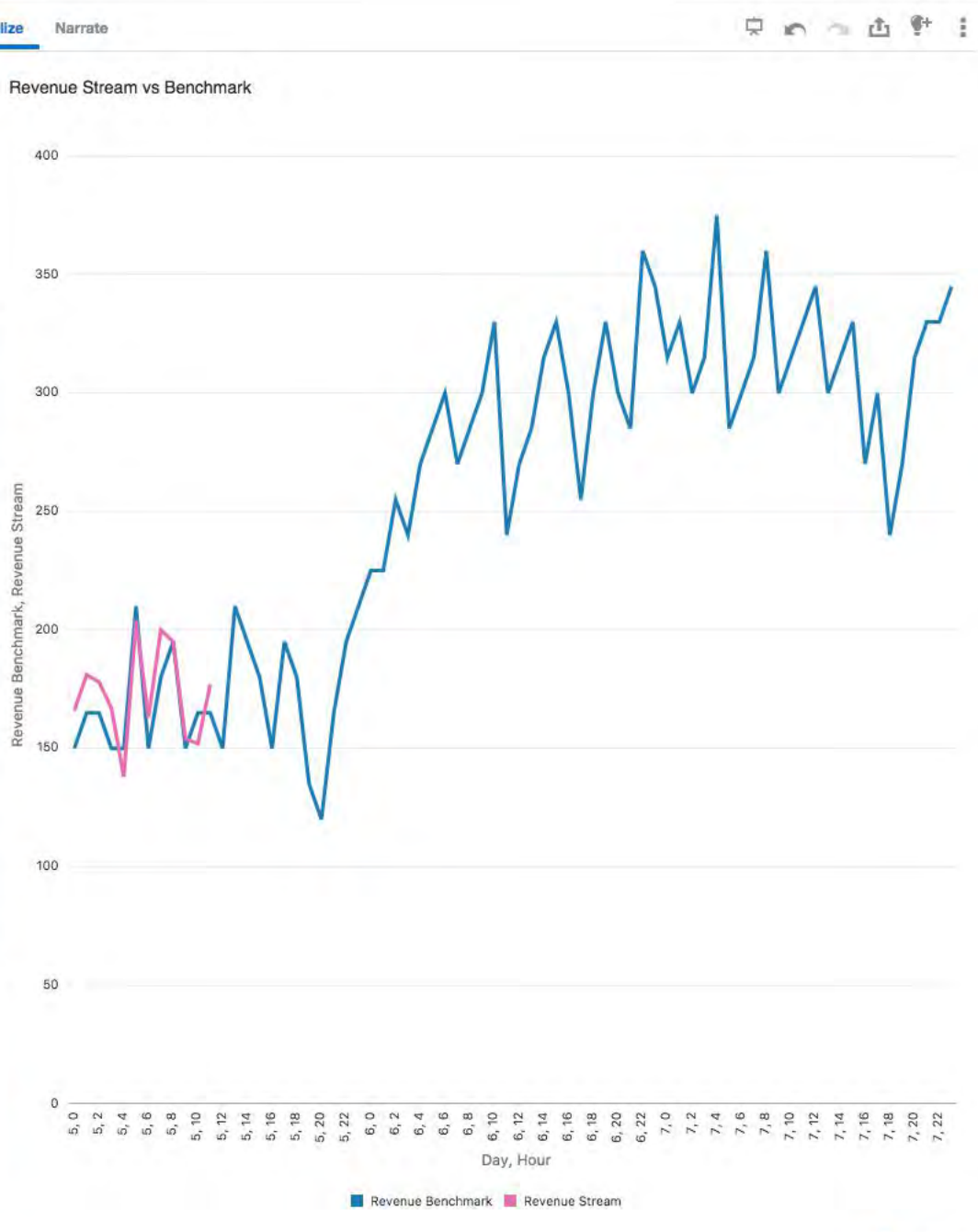
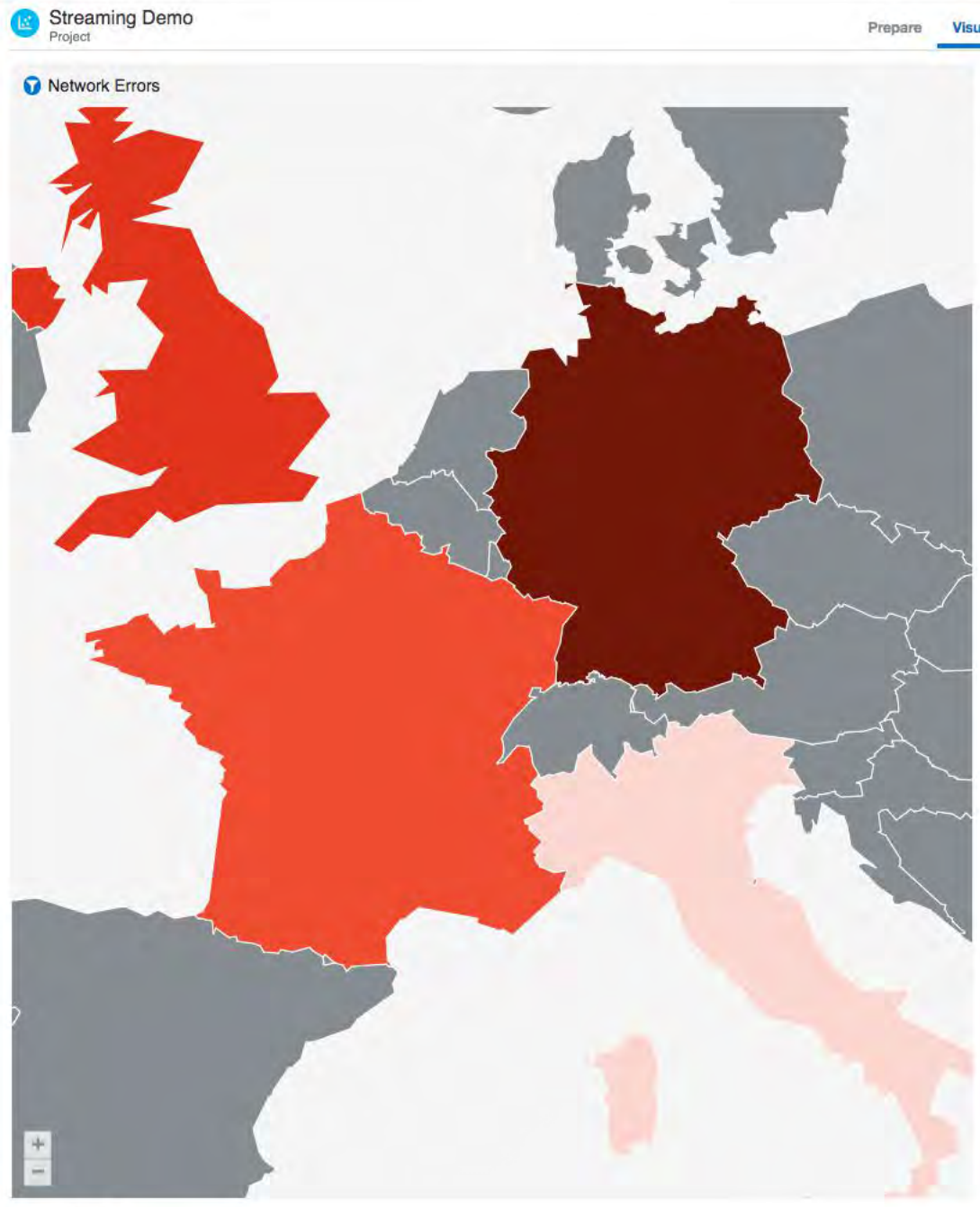
Search

KAFKA_DEMO_TABLE

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

My Calculations

Value Labels



Search

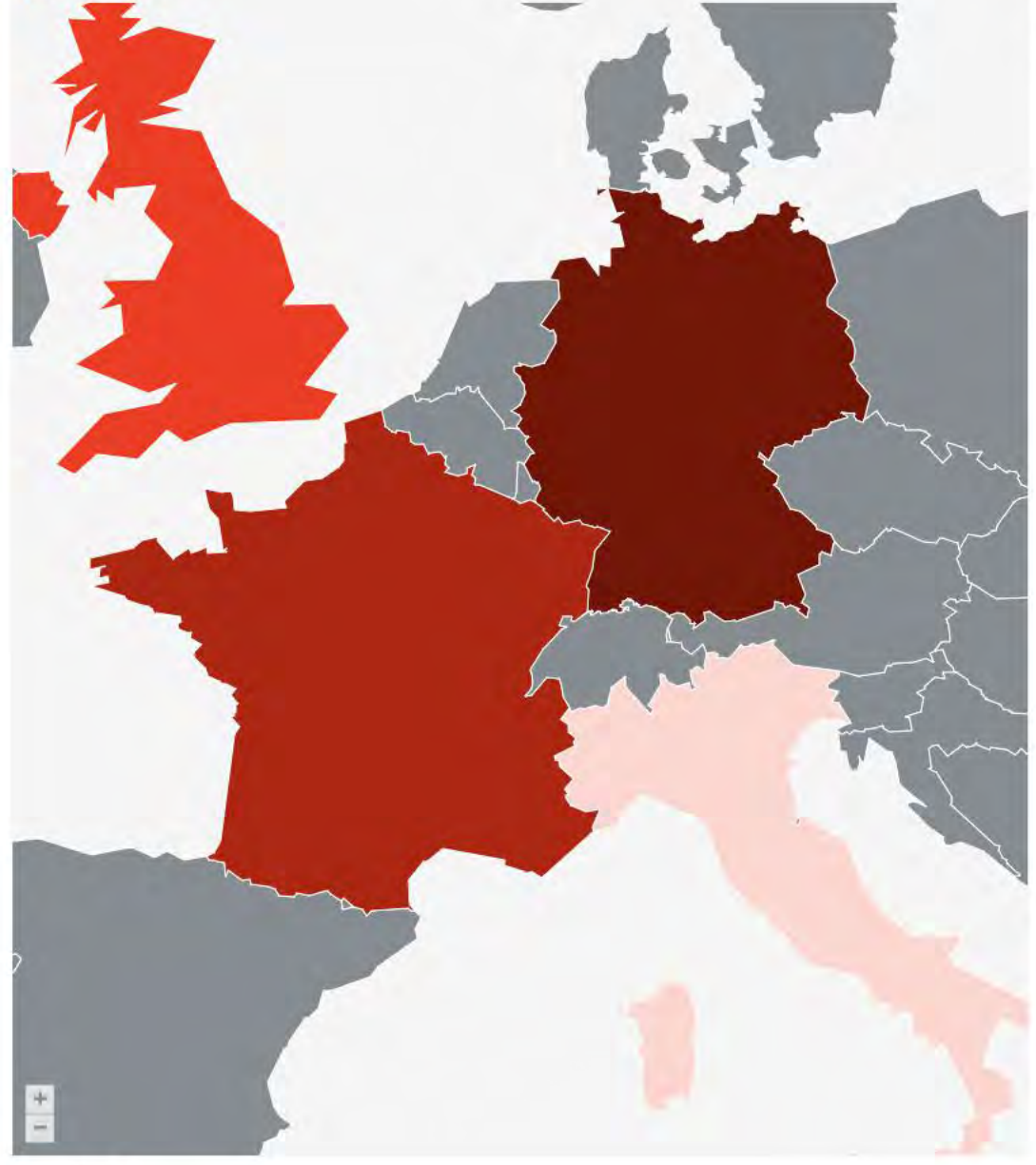
- KAFKA_DEMO_TABLE
 - Day
 - Country
 - Network Errors
 - Hour
 - Revenue Benchmark
 - Revenue Stream
- My Calculations
- Value Labels

Streaming Demo Project

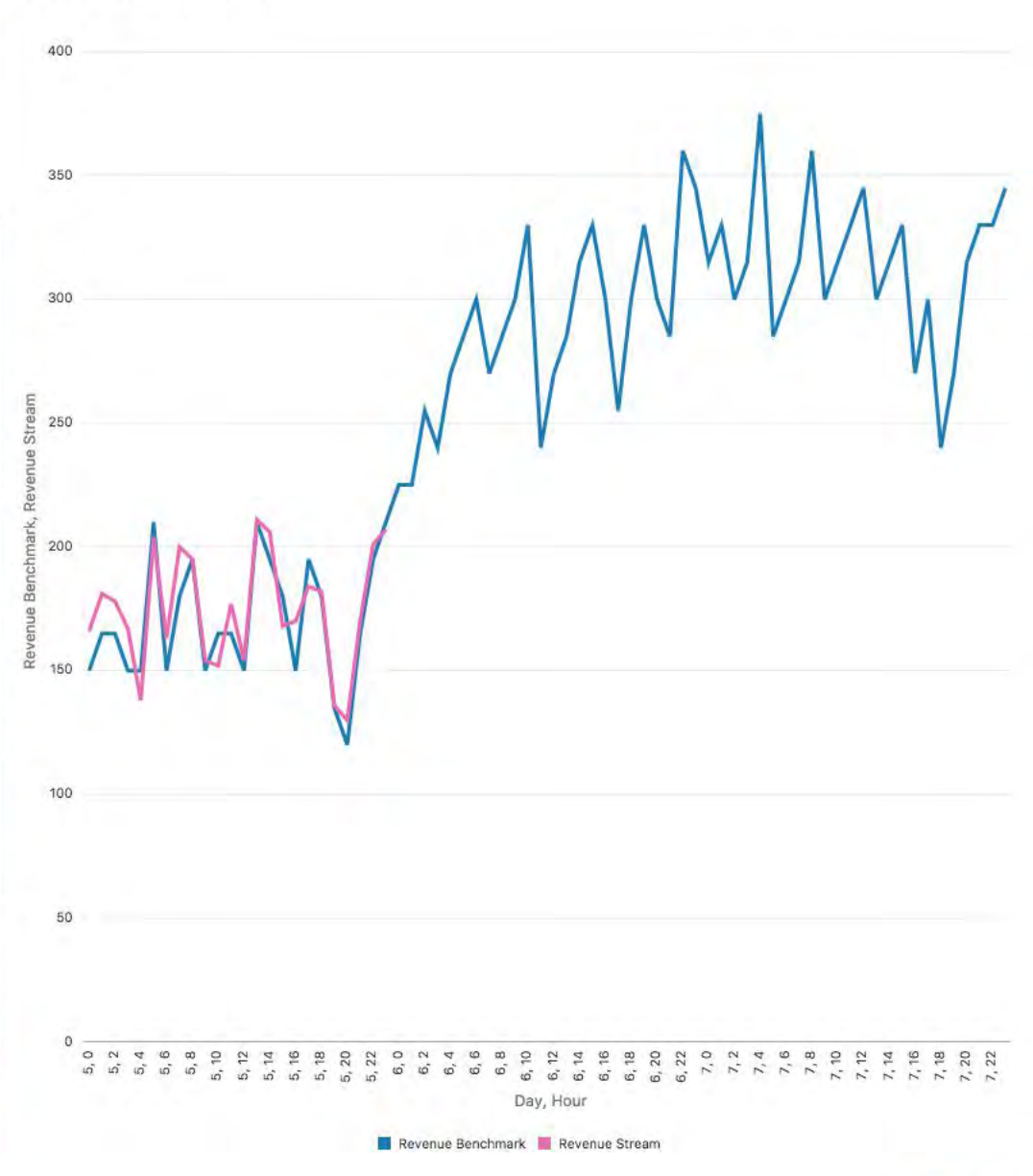
Prepare Visualize Narrate



Network Errors



Revenue Stream vs Benchmark

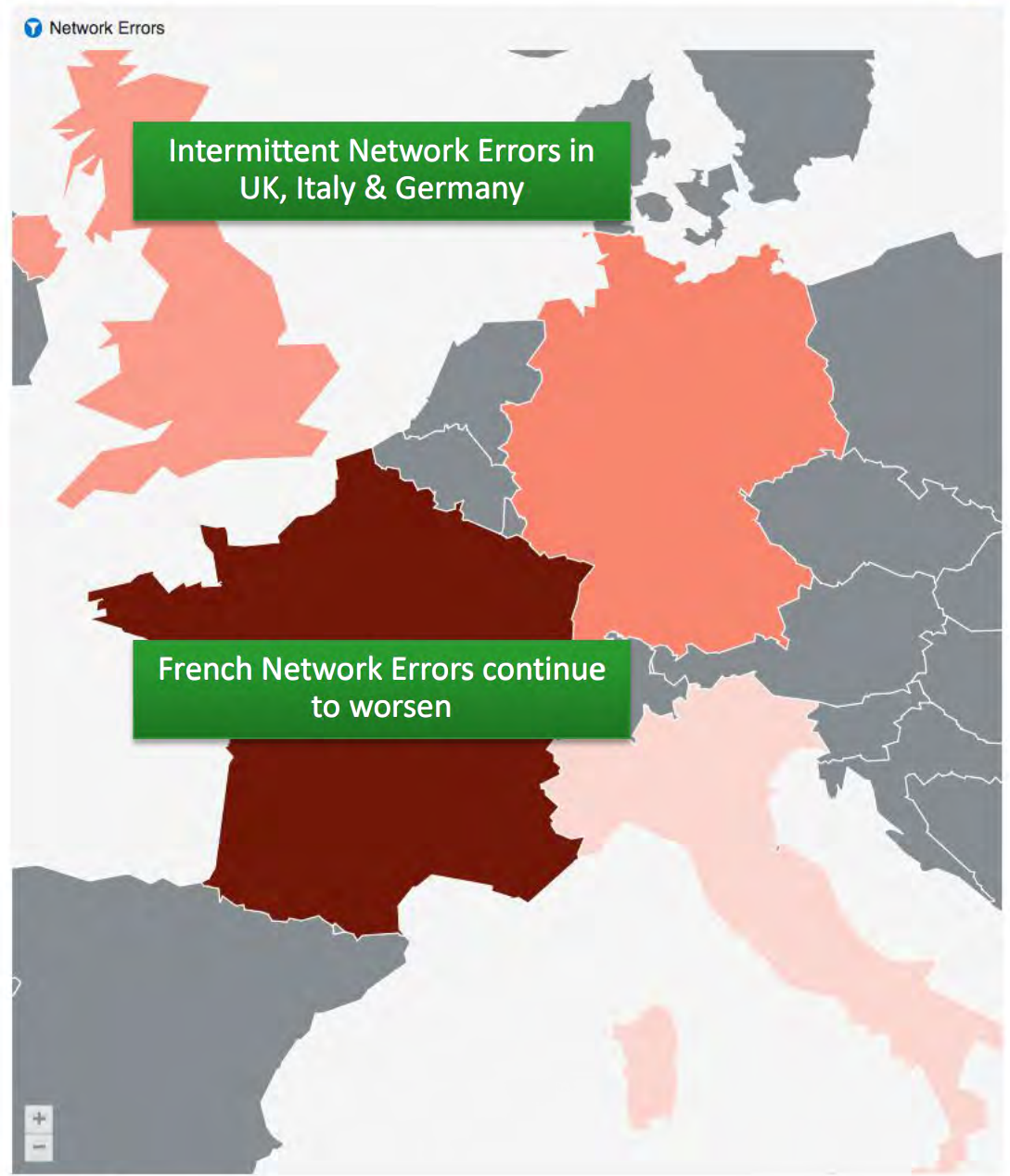


Search

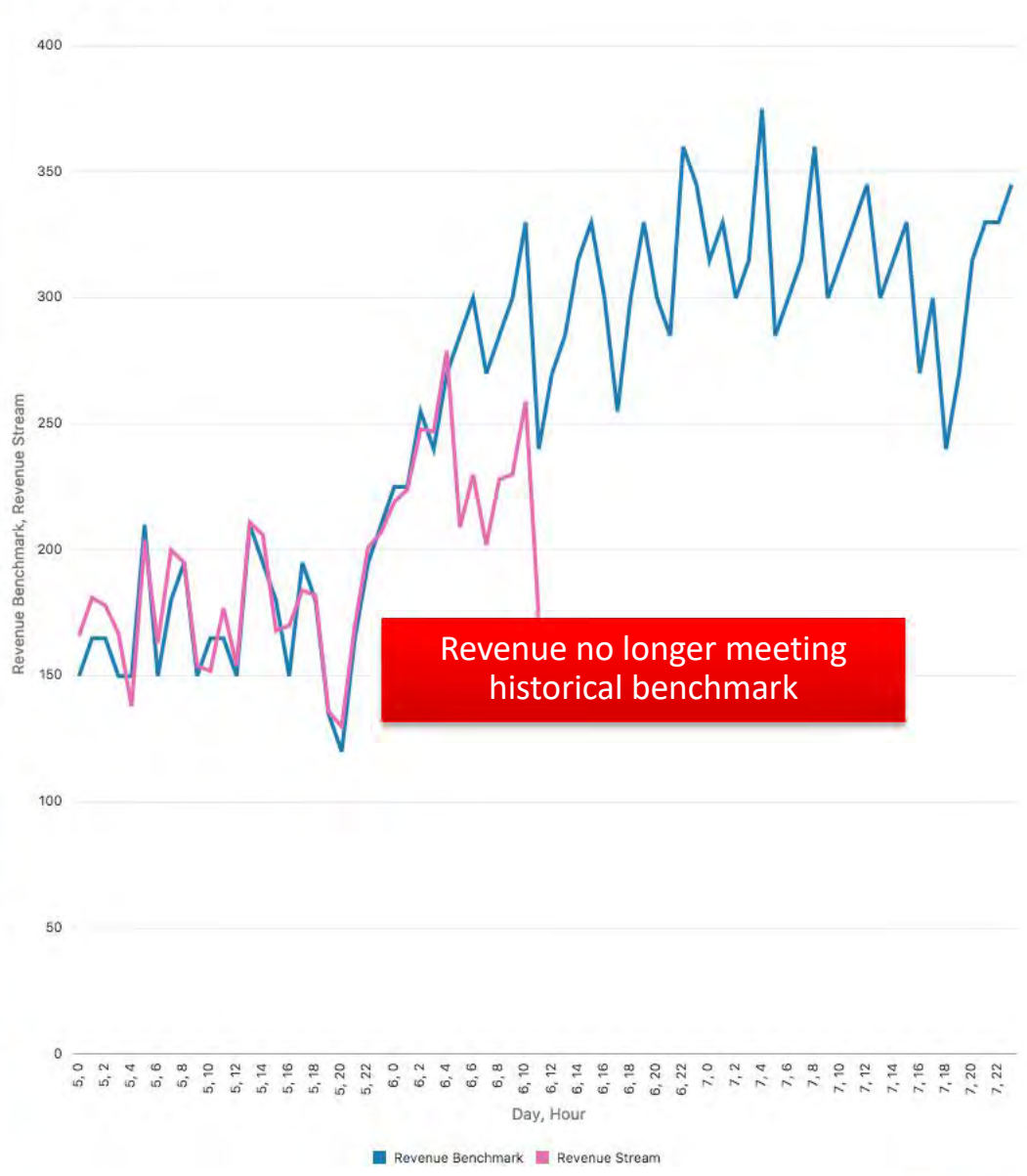
- KAFKA_DEMO_TABLE
 - Day
 - Country
 - Network Errors
 - Hour
 - Revenue Benchmark
 - Revenue Stream
- My Calculations
- Value Labels

Streaming Demo Project

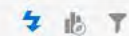
Prepare Visualize Narrate



Revenue Stream vs Benchmark

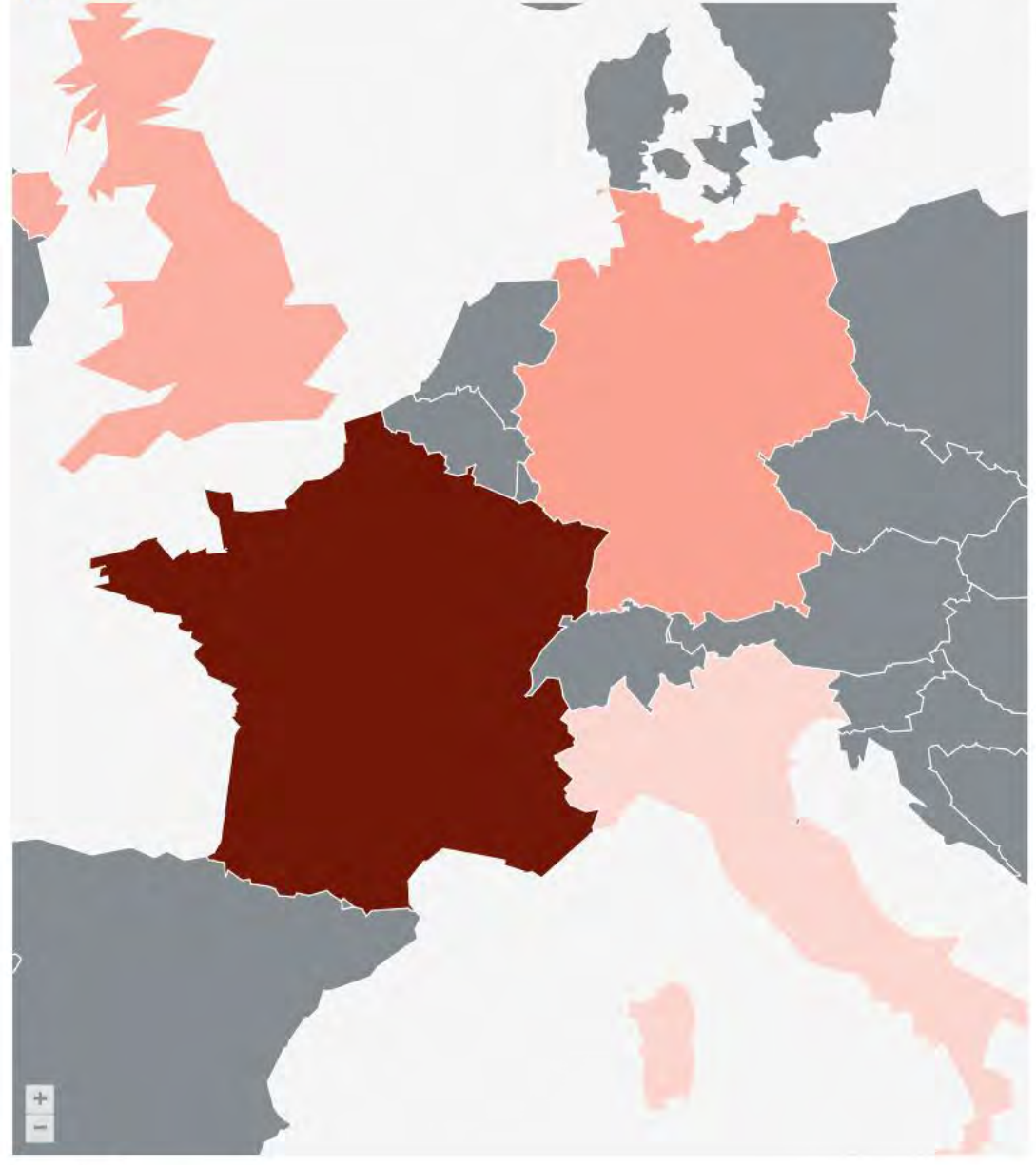


Revenue no longer meeting historical benchmark

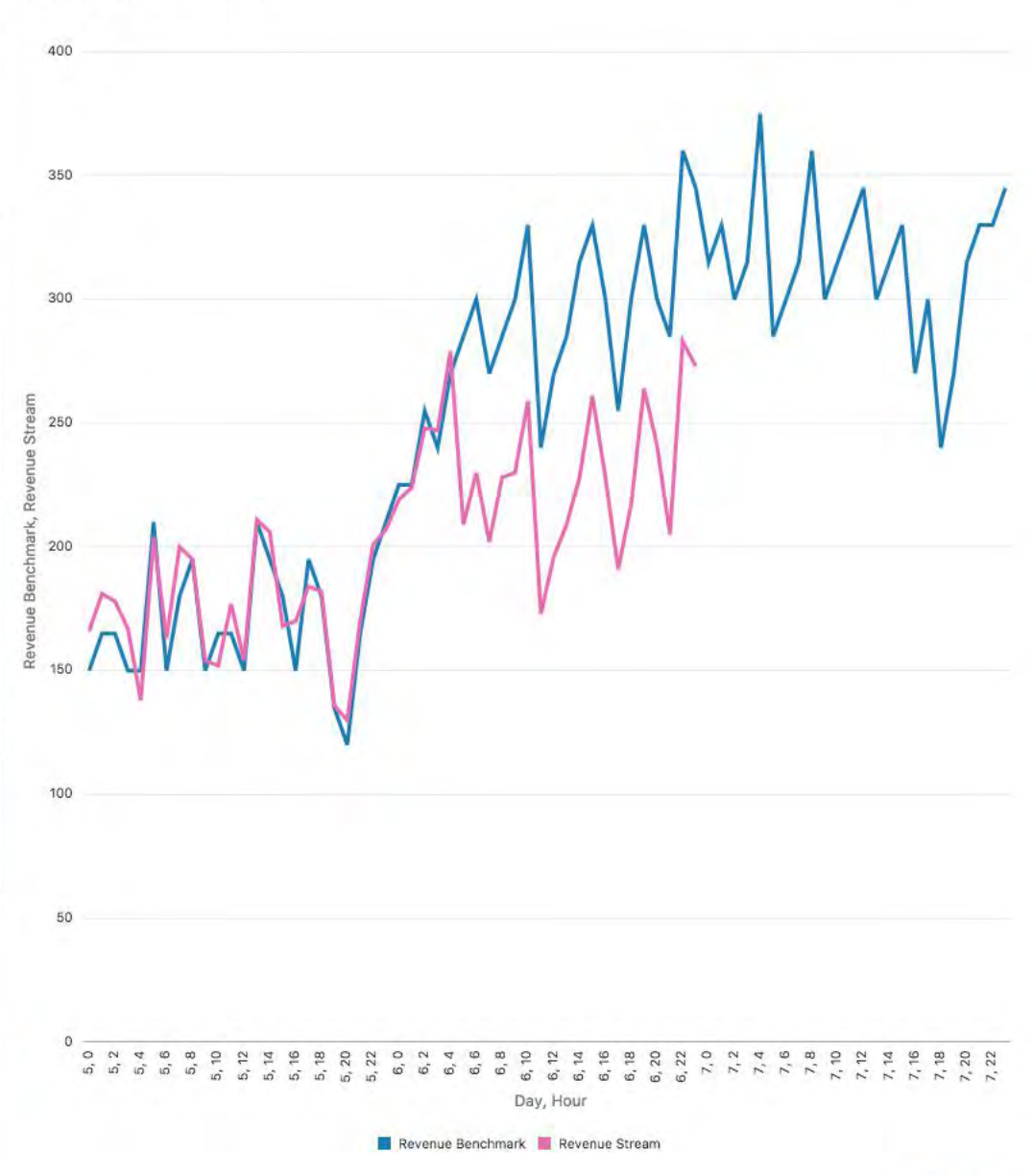


- 📁 KAFKA_DEMO_TABLE
 - 📅 Day
 - 🌐 Country
 - 🔴 Network Errors
 - 🕒 Hour
 - 🔴 Revenue Benchmark
 - 🔴 Revenue Stream
- 📁 My Calculations
 - 📄 Value Labels

📍 Network Errors



Revenue Stream vs Benchmark



Search

KAFKA_DEMO_TABLE

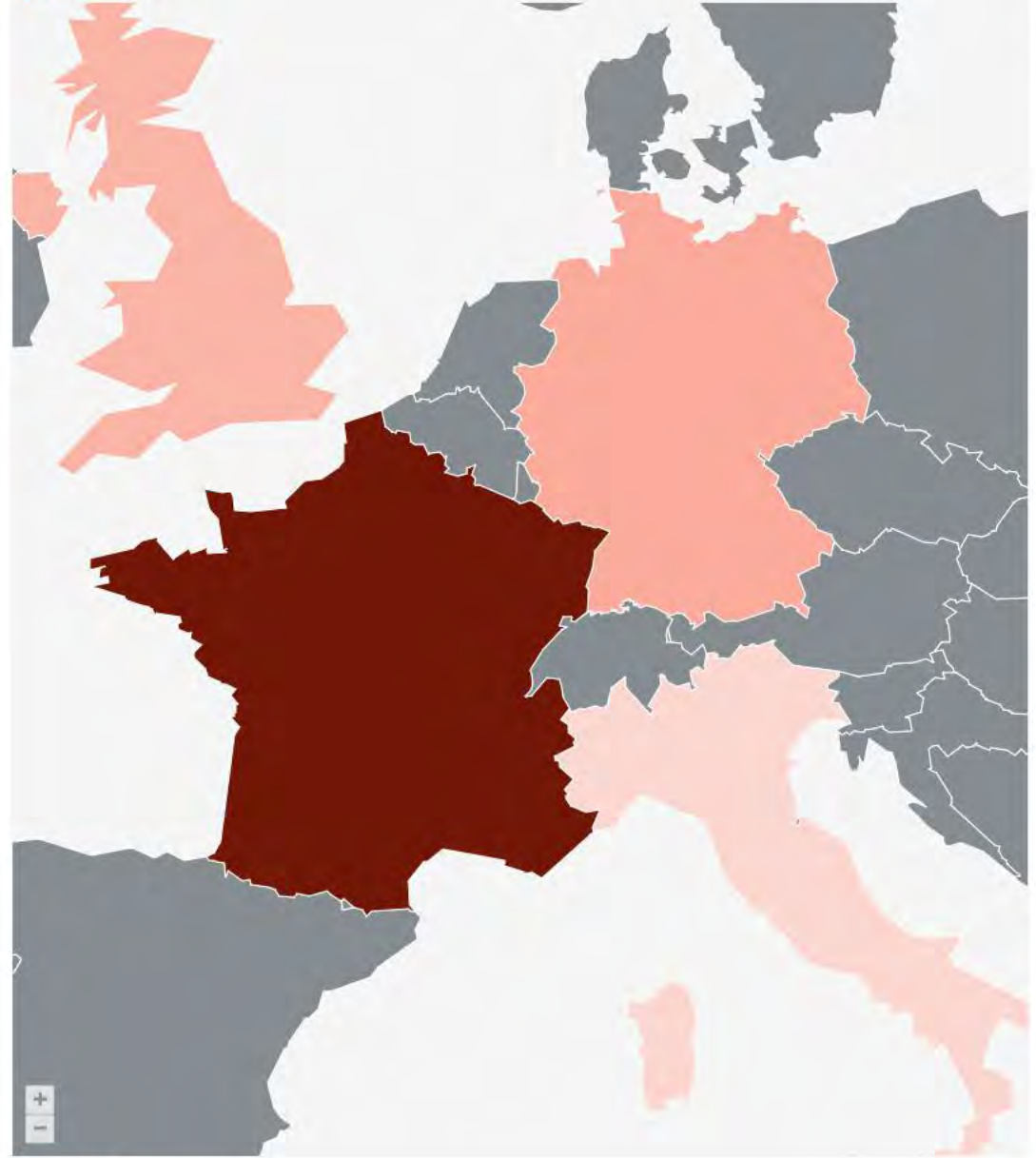
- Day
- Country
- Network Errors
- Hour
- Revenue Benchmark
- Revenue Stream

My Calculations

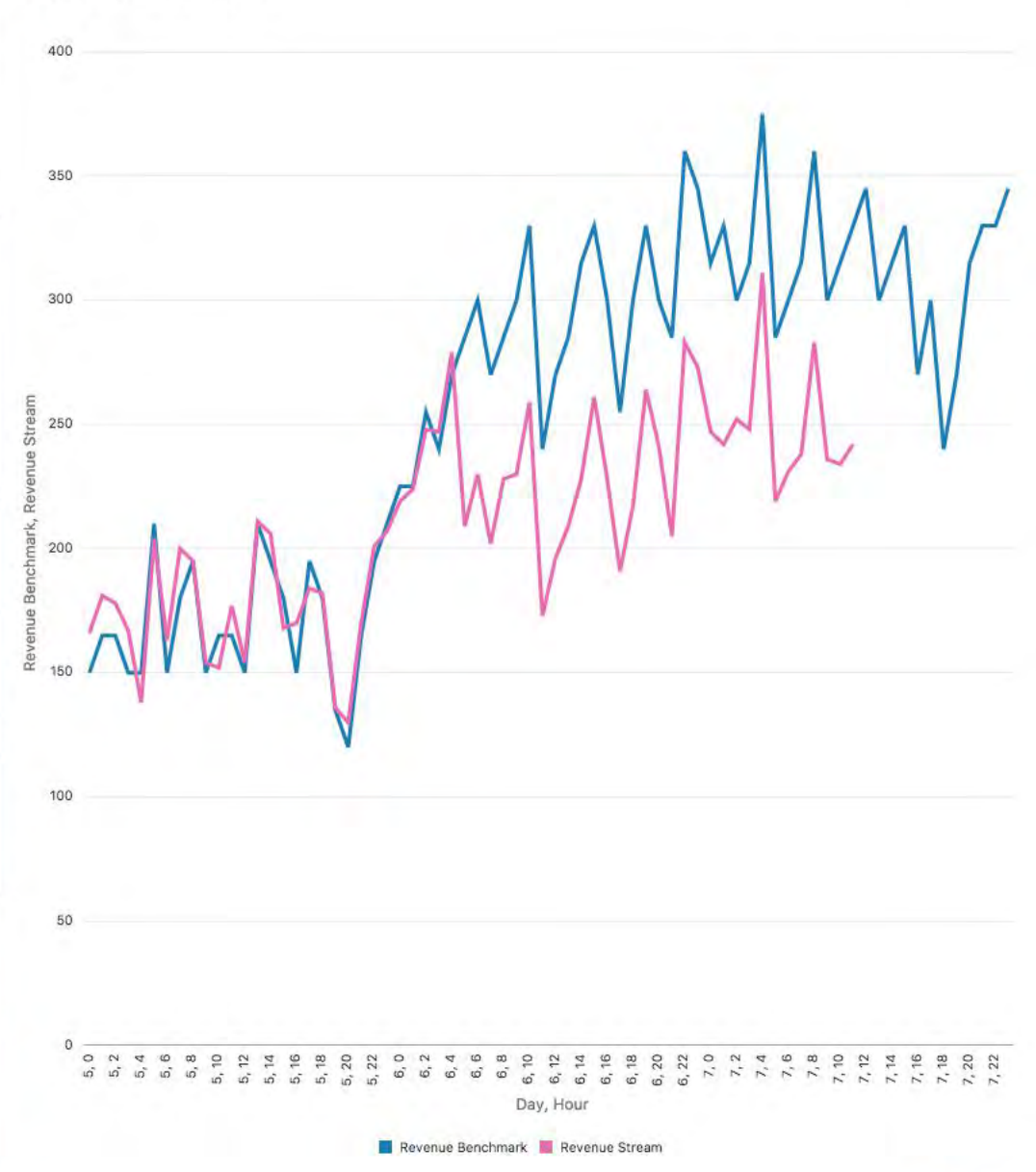
Value Labels

Streaming Demo Project

Network Errors



Revenue Stream vs Benchmark



Search

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

Streaming Demo Project

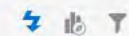
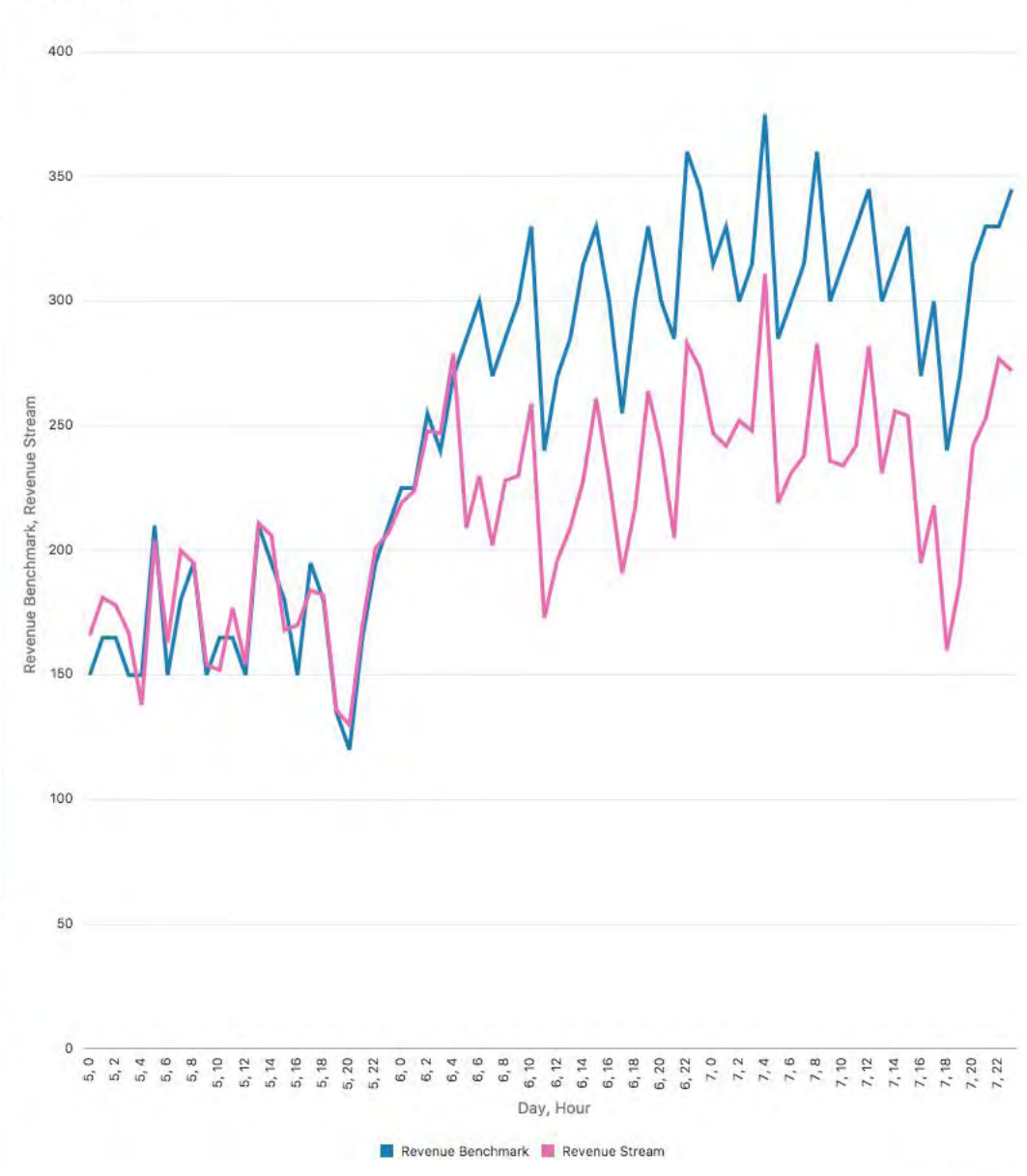
Prepare Visualize Narrate



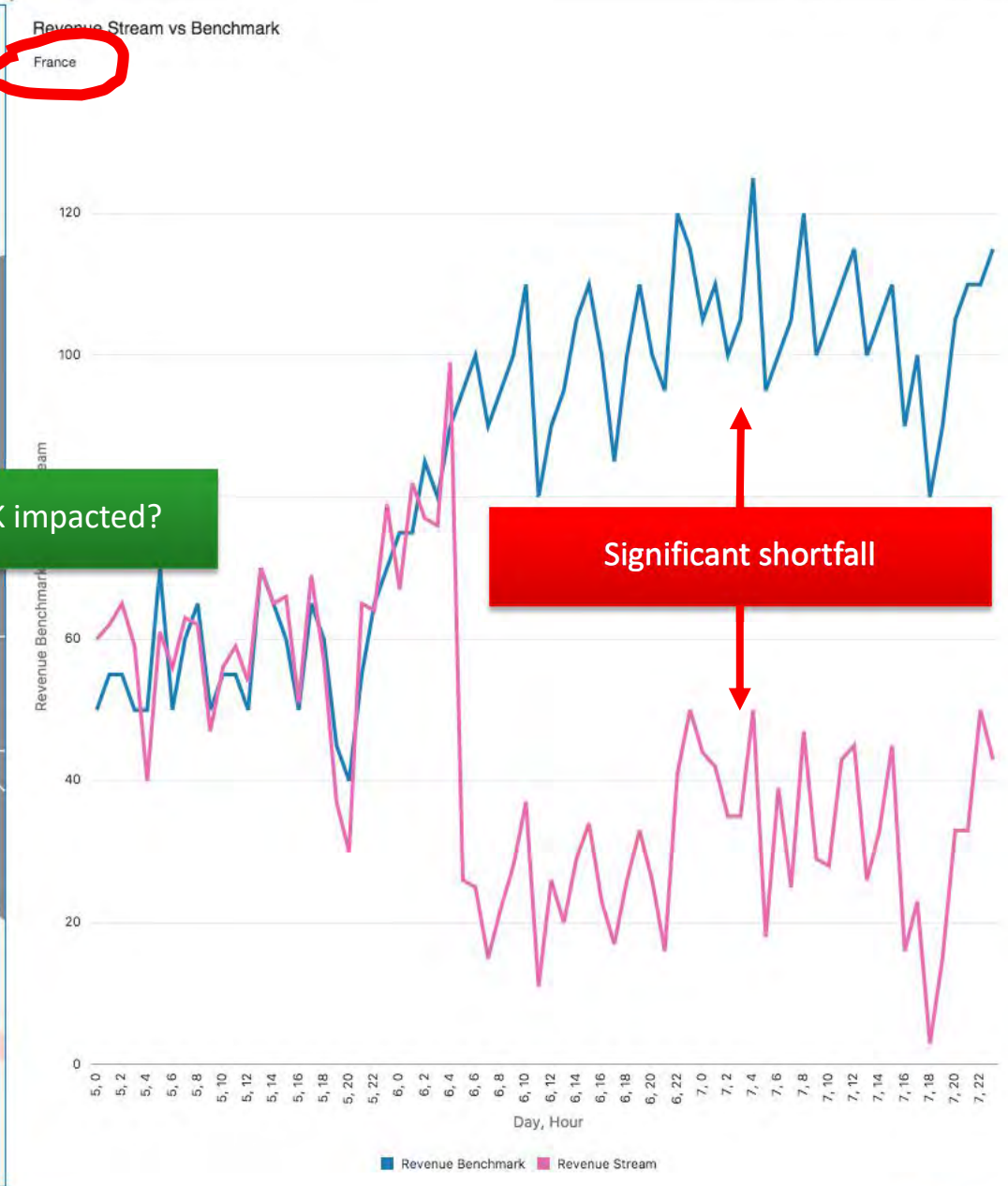
Network Errors



Revenue Stream vs Benchmark

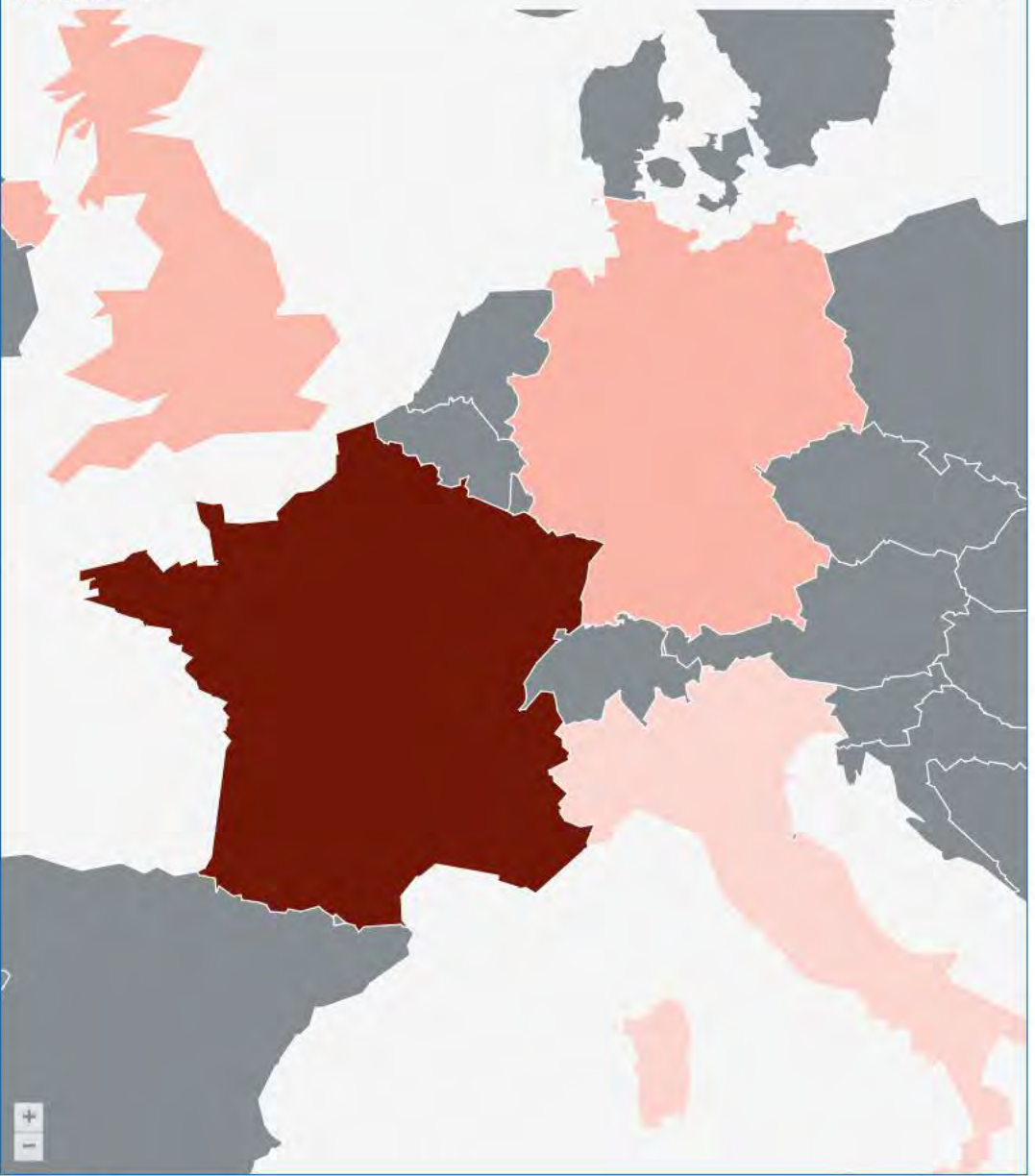


- Search
- KAFKA_DEMO_TABLE
 - Day
 - Country
 - Network Errors
 - Hour
 - Revenue Benchmark
 - Revenue Stream
- My Calculations
- Value Labels



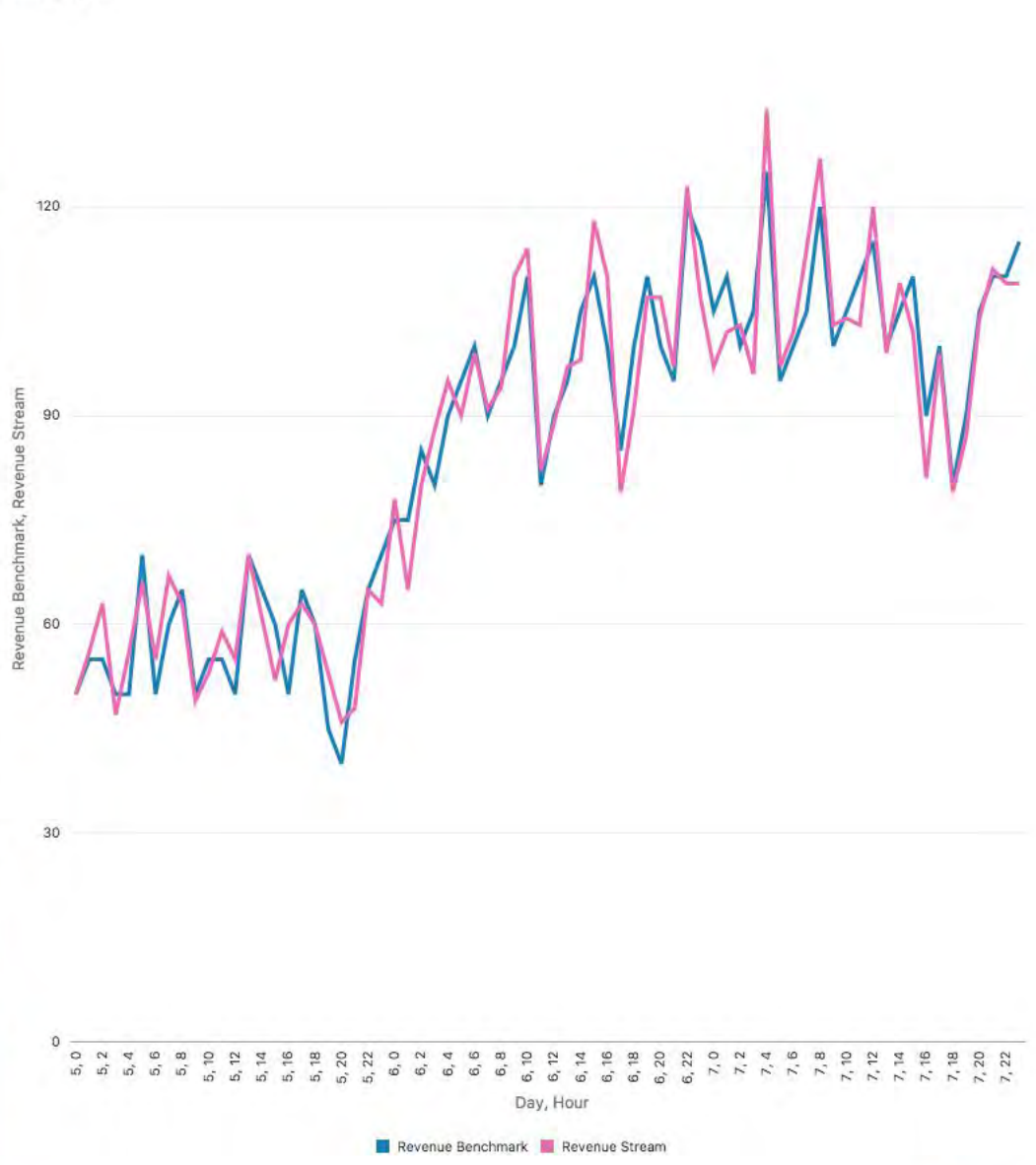
- KAFKA_DEMO_TABLE
 - Day
 - Country
 - Network Errors
 - Hour
 - Revenue Benchmark
 - Revenue Stream
- My Calculations
- Value Labels

Network Errors

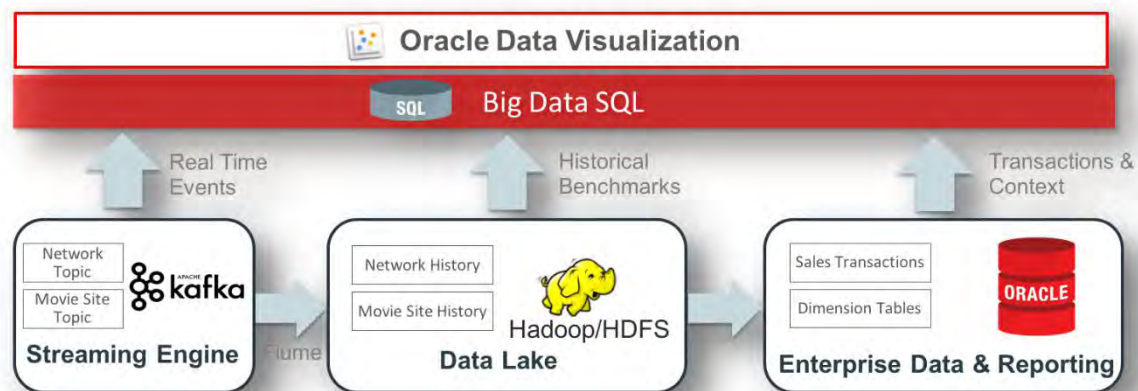


Revenue Stream vs Benchmark

United Kingdom

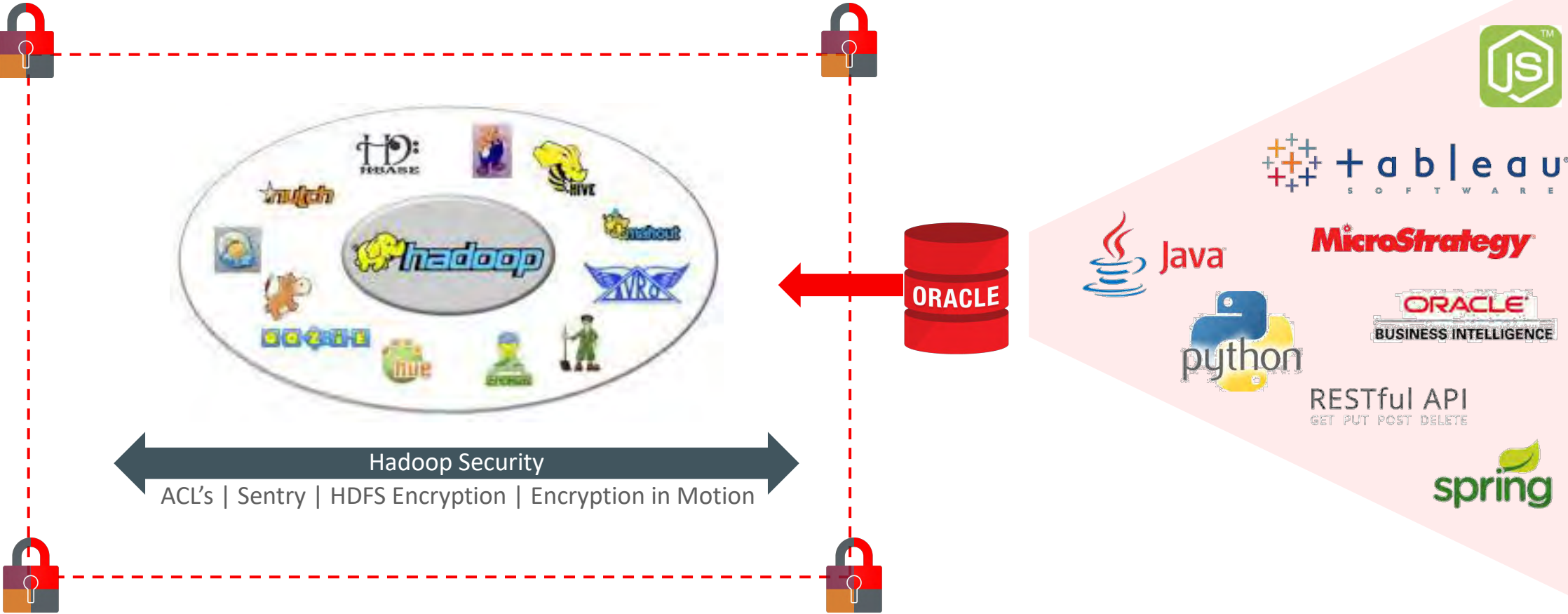


Insights Achieved with Simplicity

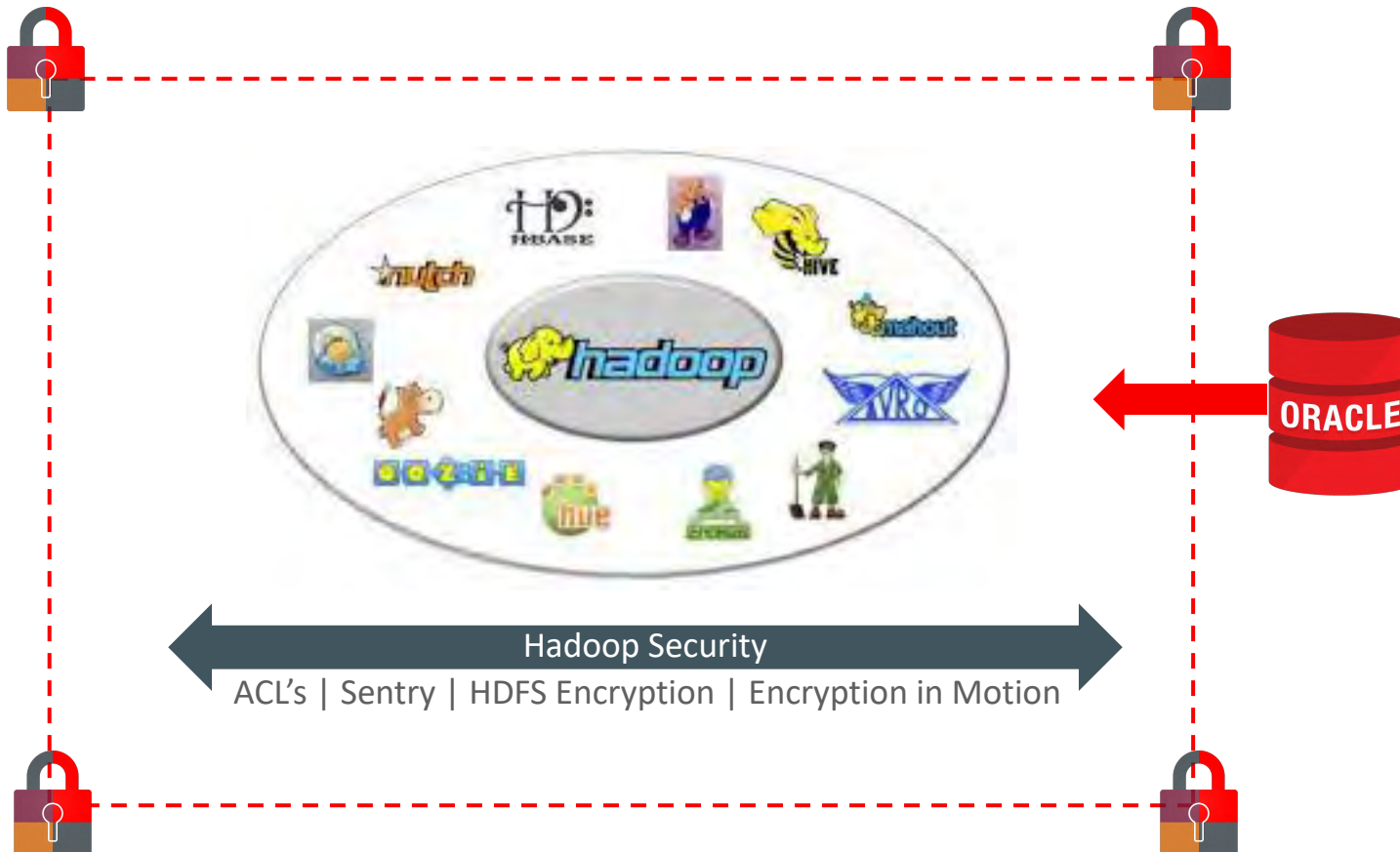


- Easily blend real time streams with history, benchmarks and context
 - Are we running at peak performance?
 - What is the opportunity cost of our current network latency?
- Any application realizes benefit
 - Use Oracle SQL and APIs over all data
- Ensure data is secure
 - Leverage Oracle advanced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