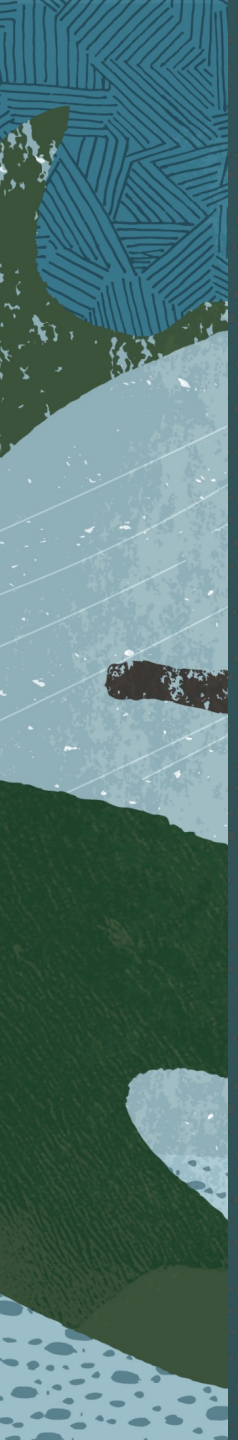
ORACLE

Tech: Project Kit

Using Java with MySQL Heatwave & Lakehouse

Lightning Talk

Tony Darnell
Principal Solutions Engineer
MySQL



MySQL: An extremely popular database

MySQL is the #1 Open Source Database

Rank Mar 2024	Rank Feb 2024	Rank Mar 2023	DBMS	Database Model	Mar 2024
1.	1.	1.	Oracle	Relational, Multi-model	1221.06
2.	2.	2.	MySQL	Relational, Multi-model	1101.50
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	845.81
4.	4.	4.	PostgreSQL	Relational, Multi-model	634.91
5.	5.	5.	MongoDB	Document, Multi-model	424.53

DB-ENGINES

Innovative organizations across many industries run MySQL – Do you?

Social

E-Commerce

Tech

Finance

Manufacturing

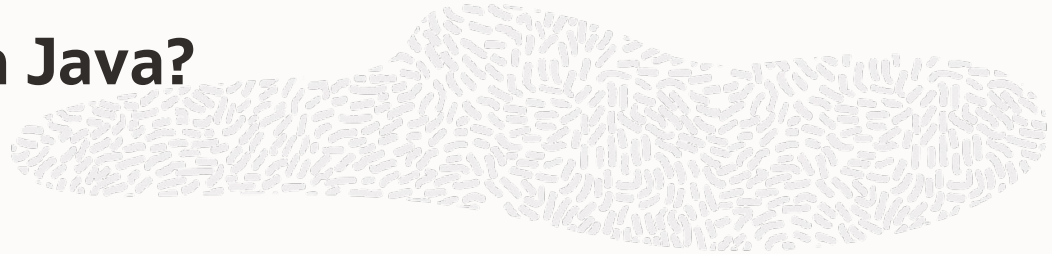
MySQL powers Open Source applications

Custom Apps Development

Content management and eCommerce

Learning platforms

What do I need to connect to MySQL with Java?



MySQL Java Connector – Connector/J

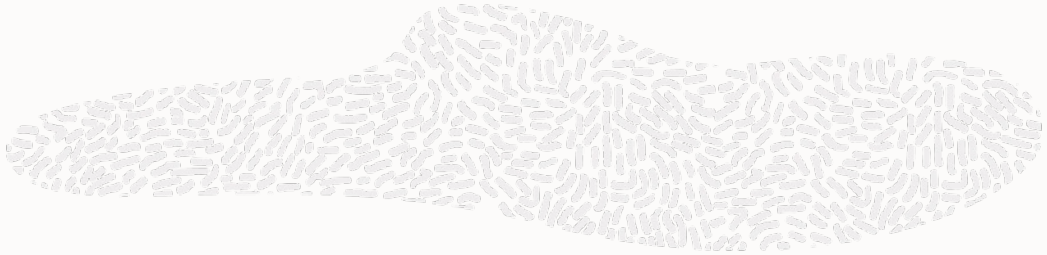
Download at MySQL Community Downloads page:

- https://dev.mysql.com/downloads/connector/j/

Add connector location to CLASSPATH in your config file:

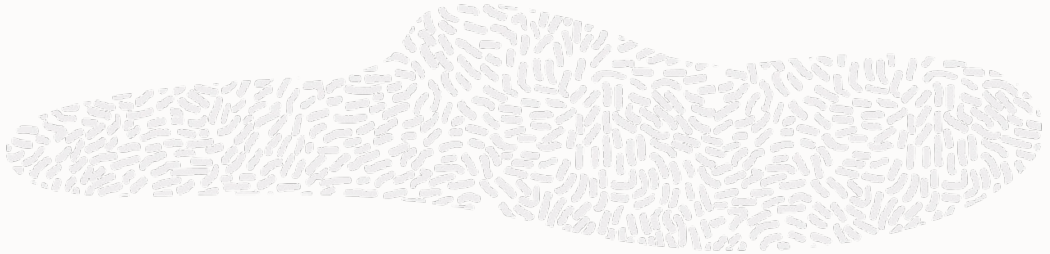
- export CLASSPATH=/usr/local/mysql/java/**mysql-connector-j-8.3.0.jar**:$CLASSPATH

Example Java – Select Records



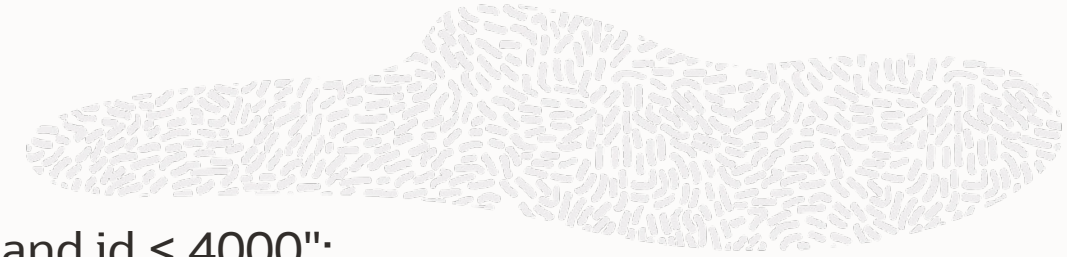
```
import java.sql.*;

public class JdbcSelectTest {
   public static void main(String[] args) {
      try (
         Connection conn = DriverManager.getConnection(
              "jdbc:mysql://localhost:3306/database", "user", "password");
         Statement stmt = conn.createStatement();
      ) {
         String strSelect = "select id_employee, name_first, name_last from employee";
         System.out.println("The SQL statement is: " + strSelect + "\n");
         ResultSet rset = stmt.executeQuery(strSelect);
```

Example Java – Select Records



```
System.out.println("The records selected are:");
      int rowCount = 0;
      while(rset.next()) {
        String id_employee = rset.getString("id_employee");
        String name_first = rset.getString("name_first");
        String name_last   = rset.getString("name_last");
        System.out.println(id_employee + ", " + name_first + ", " + name_last);
        ++rowCount;
      }
      System.out.println("Total number of records = " + rowCount);
    } catch(SQLException ex) {
      ex.printStackTrace();
    } } }
```

Other DML Commands - Examples



String **sqlDelete** = "delete from books where id >= 3000 and id < 4000";

String **sqlInsert** = "insert into books values (3001, 'Gone Fishing', 'Kumar', 11.11, 11)";

sqlInsert = "insert into books values "
+ "(3002, 'Gone Fishing 2', 'Kumar', 22.22, 22),"
+ "(3003, 'Gone Fishing 3', 'Kumar', 33.33, 33)";

sqlInsert = "insert into books (id, title, author) values (3004, 'Fishing 101', 'Kumar')";

String **strUpdate** = "update books set price = price*1.07, qty = qty+1 where id = 1001";

Challenges for developers and DBAs…

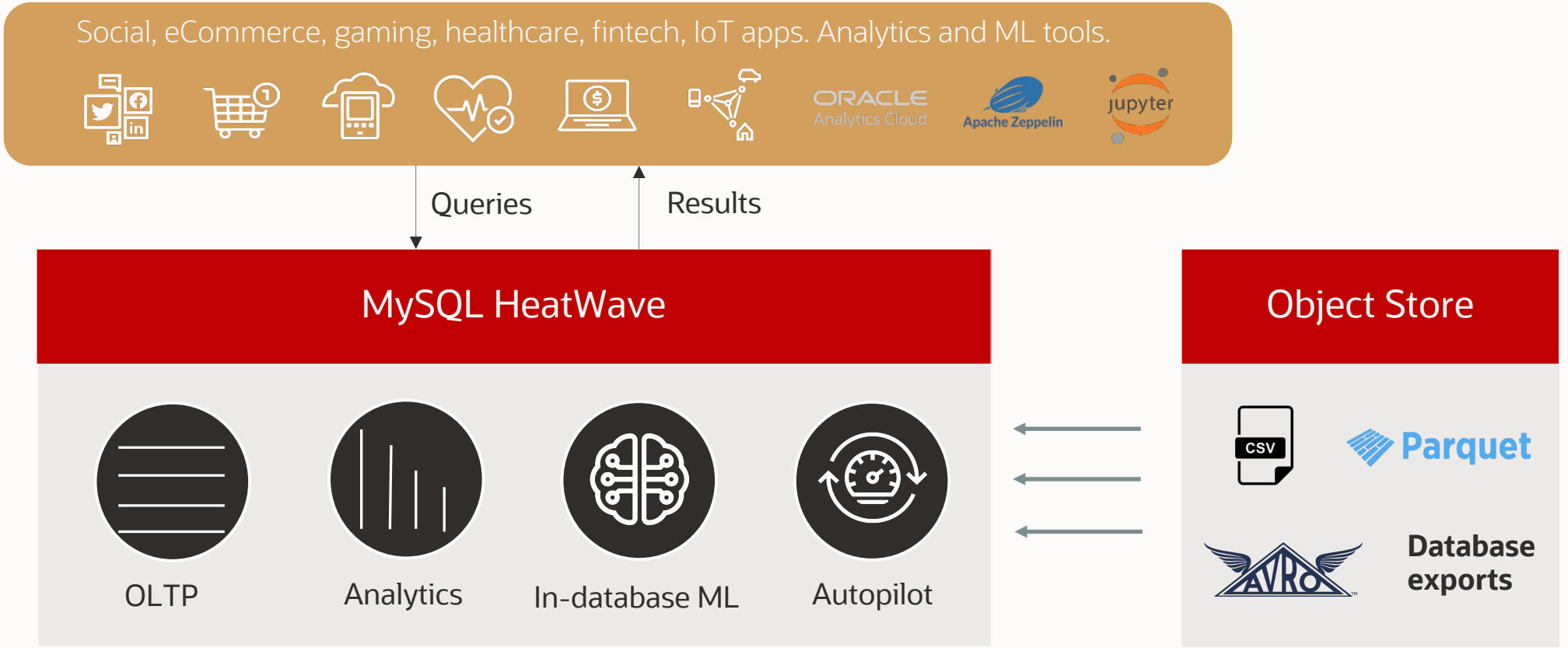
…And how MySQL HeatWave uniquely addresses them

What we've heard from customers

Complex and costly to use separate systems for transactions and analytics	We provide one MySQL cloud database service for OLTP and real-time analytics across data warehouses and data lakes—without ETL duplication
Want to leverage ML and generative AI on all their data	We provide automated in-database ML with an explanation of models and results, without ETL, plus generative AI with Vector store (private preview)
Spend too much time on manual management tasks	We provide a fully managed database service with machine learning-powered automation and built-in advanced security features
Want to use multiple clouds	MySQL HeatWave is available on OCI, AWS, and Azure.

MySQL HeatWave overview

Transactions, real-time analytics across data warehouse and data lake, and machine learning in one database service

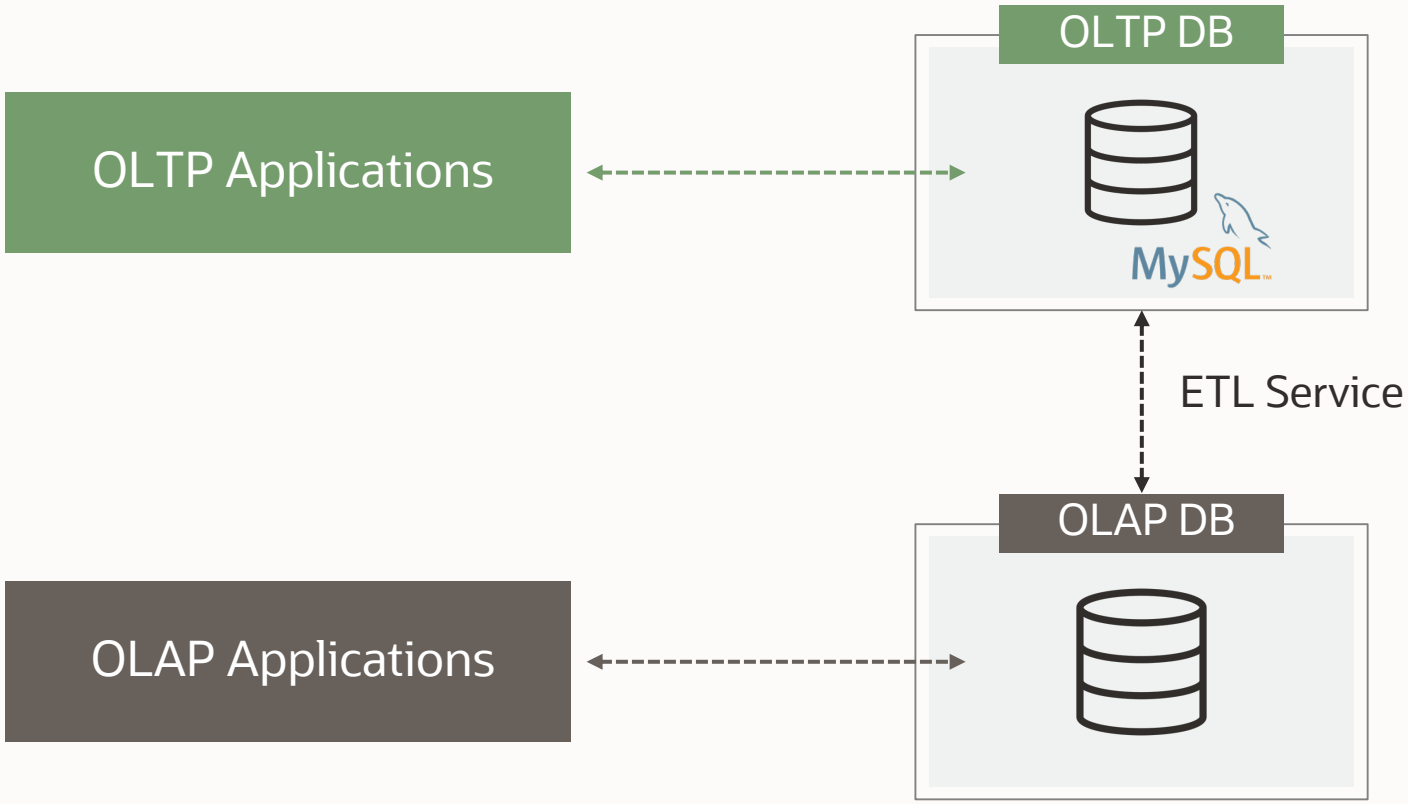


Data remains in object store, processing is done in HeatWave

For both non-MySQL and MySQL workloads

Challenge: Organizations need to use separate systems for transactions and analytics

MySQL is optimized for OLTP, not designed for analytic processing



Separate analytics database

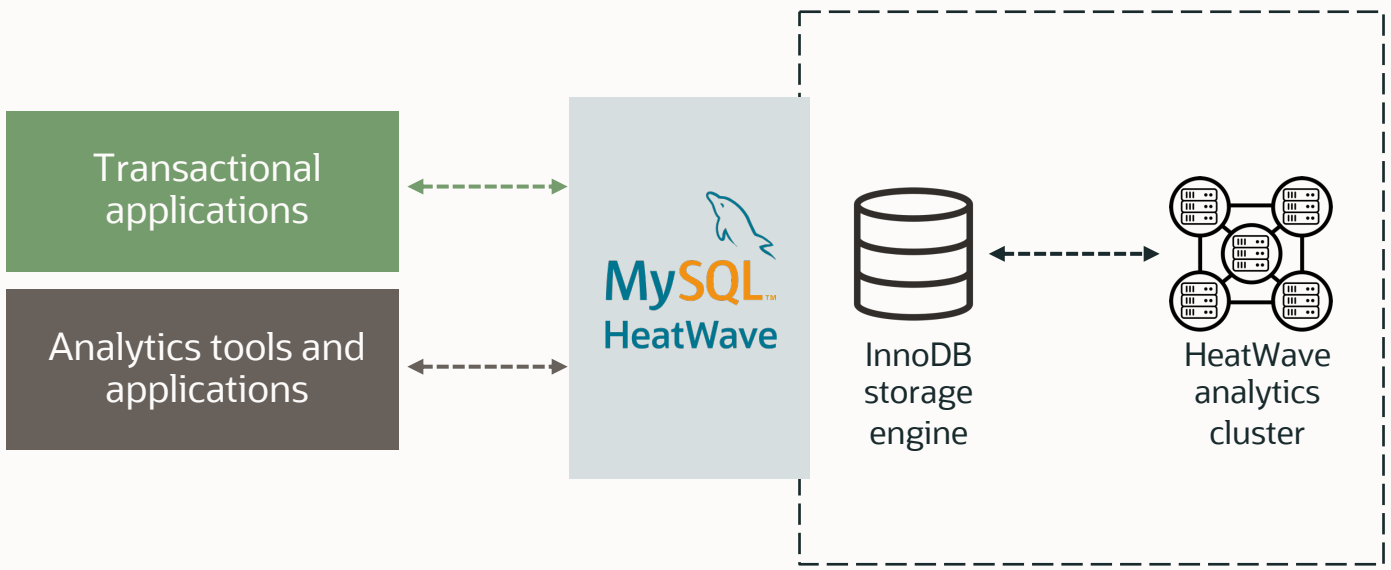
Complex ETL

No real-time analytics

Security & compliance risks

Increased costs

One database is better than two



MySQL

1>2 with MySQL HeatWave

- One service for OTLP & OLAP
- No ETL duplication
- Unmatched performance, at a fraction of the cost
- Real-time analytics
- Improved security
- Applications work without changes

Best performance in industry for data warehouse

TPC-H 10TB

Faster time to insights = faster business response to market trends

4X faster than Redshift	4X faster than Snowflake	9X faster than BigQuery	11X faster than Databricks
10X ra3.4xlarge	X-Large Cluster	800 slots	Large Cluster

Get answers in hours, not days

According to 10 TB TPC-H benchmarks as of May 23, 2023. Redshift, Snowflake, Databricks and BigQuery numbers for 10TB TPC-H numbers are provided by a third party.
Benchmark queries are derived from the TPC-H benchmarks, but results are not comparable to published TPC-H benchmark results since these do not comply with the TPC-H specifications.

Lowest cost in industry for data warehouse

Price performance comparison 10TB TPC-H

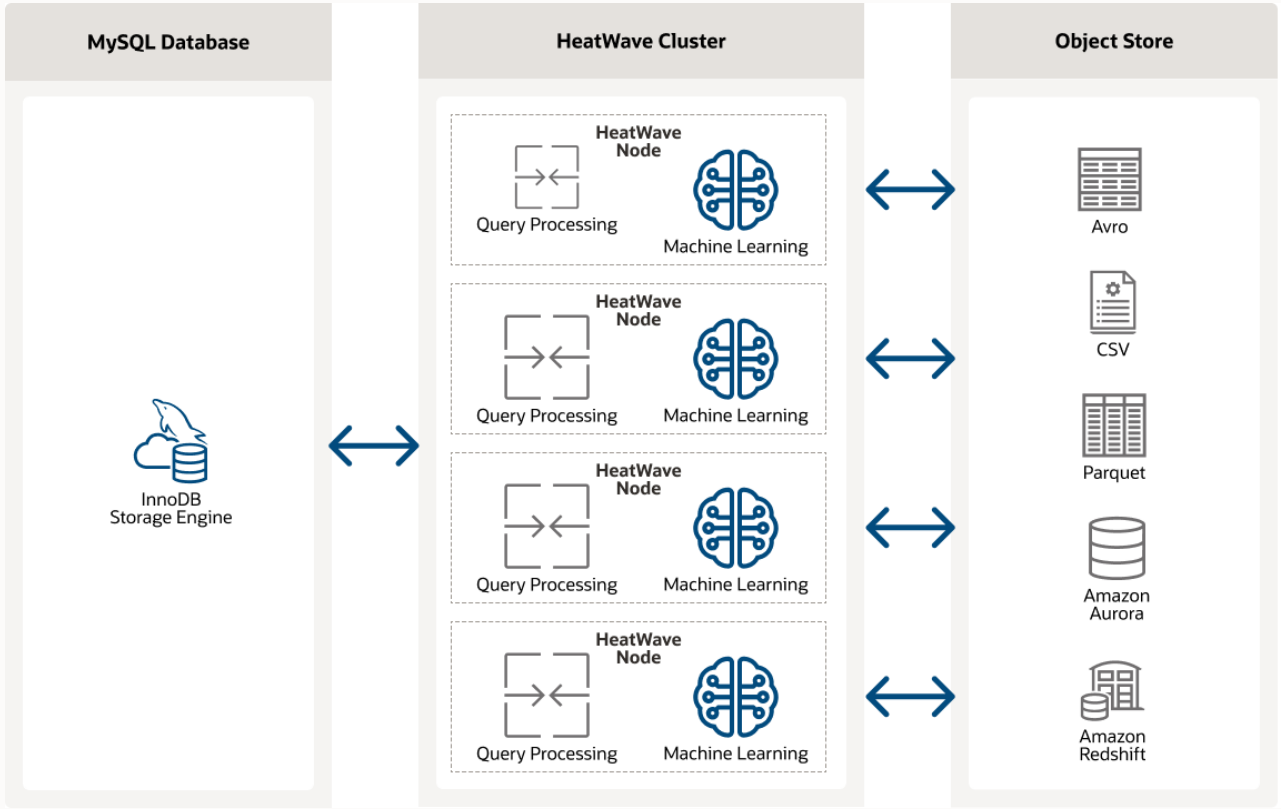
10X	15X	20X	37X
better than Redshift	better than Snowflake	better than BigQuery	better than Databricks
1 year reserved, paid upfront	Standard Edition	1 year reserved	1 year reserved

Much less expensive

According to 10 TB TPC-H benchmarks as of May 23, 2023. Redshift, Snowflake, Databricks and BigQuery numbers for 10TB TPC-H numbers are provided by a third party.
Benchmark queries are derived from the TPC-H benchmarks, but results are not comparable to published TPC-H benchmark results since these do not comply with the TPC-H specifications.

MySQL HeatWave Lakehouse

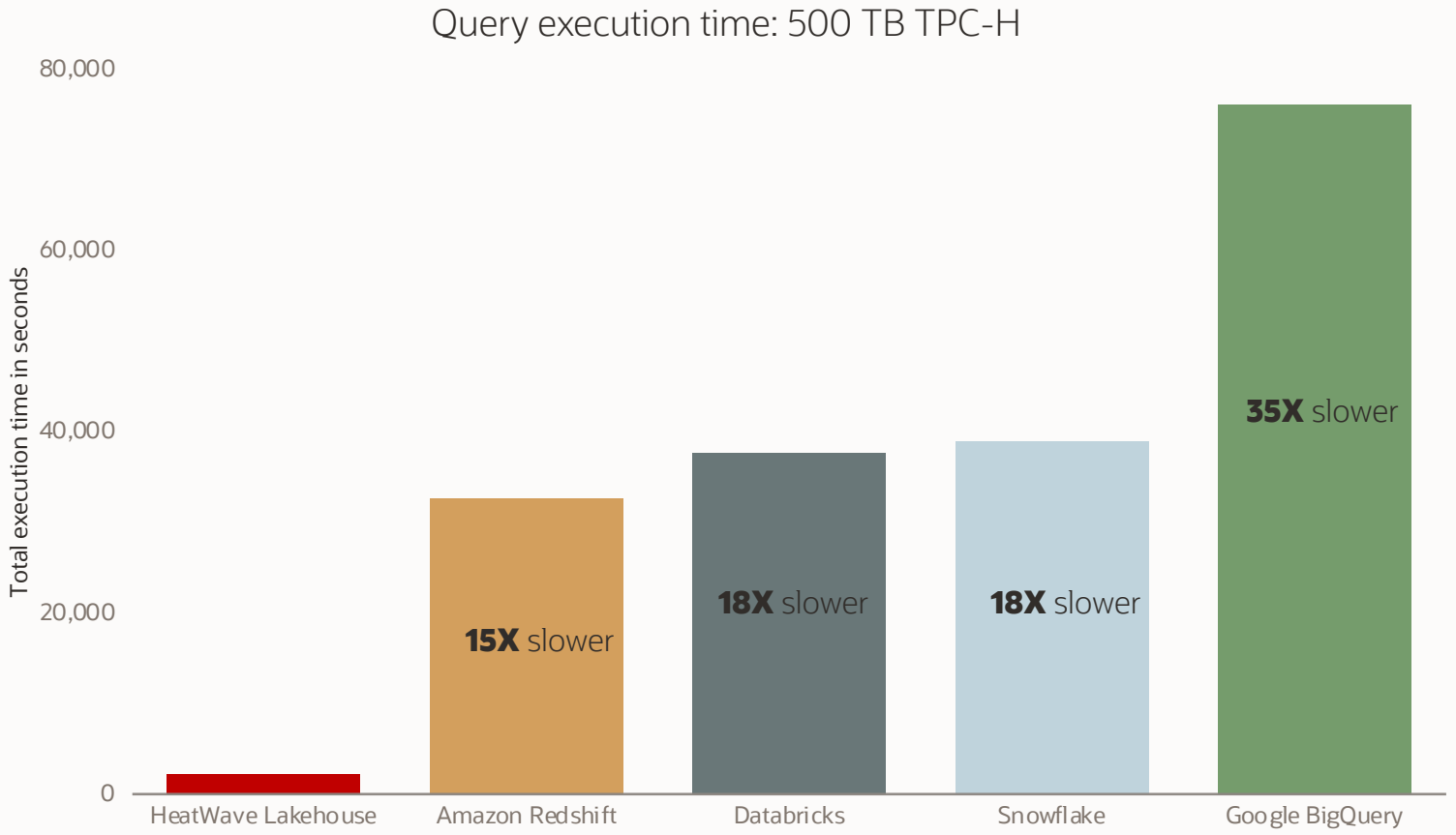
Query half a PB data in the object store—in a variety of file formats



- Query data in MySQL, in the object store, or across both—using standard SQL syntax
- Up to 500 TB of data—the HeatWave cluster scales to 512 nodes
- Querying the data in the object store is as fast as querying the database – **an industry first!**
- Scale out data processing in the object store, data is not copied to the MySQL Database: for both MySQL and non-MySQL workloads

Query performance of HeatWave Lakehouse

15X faster than Redshift, 18X faster than Snowflake, 18X faster than Databricks, 35X faster than BigQuery

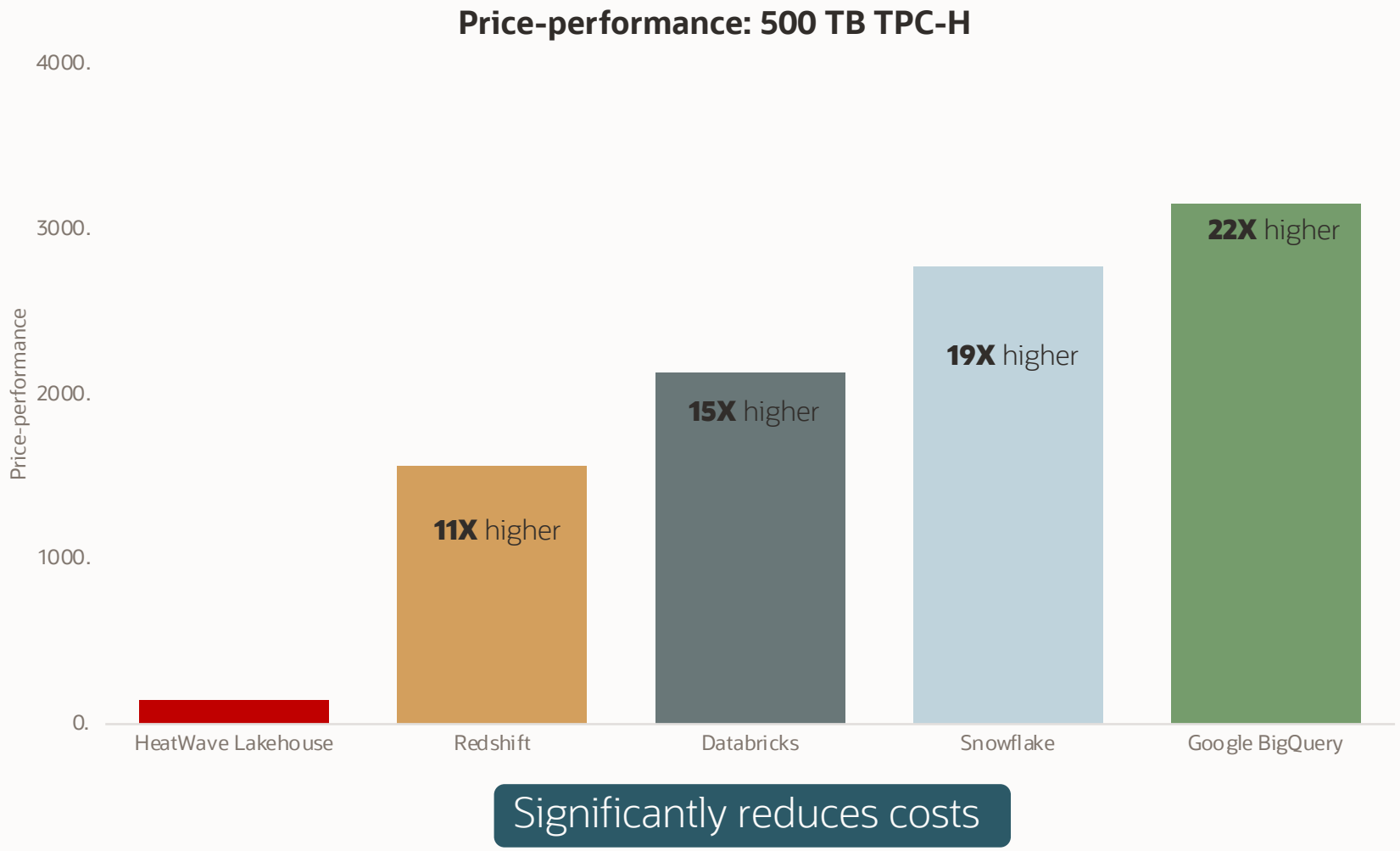


Significantly reduces time-to-insights

Configuration: MySQL HeatWave Lakehouse: 512 nodes; Snowflake: 4X-Large Cluster; Databricks: 3X-Large Cluster; Amazon Redshift: 20-ra3.16xlarge; Google BigQuery: 6400 slots
Benchmark queries are derived from the TPC-H benchmarks, but results are not comparable to published TPC-H benchmark results since these do not comply with the TPC-H specifications.

Query price-performance of HeatWave Lakehouse

11X better than Redshift, 15X better than Databricks, 19X better than Snowflake, 22X faster than BigQuery



Significantly reduces costs

Configuration: MySQL HeatWave Lakehouse: 512 nodes; Snowflake: 4X-Large Cluster; Databricks: 3X-Large Cluster; Amazon Redshift: 20-ra3.16xlarge; Google BigQuery: 6400 slots

Benchmark queries are derived from the TPC-H benchmarks, but results are not comparable to published TPC-H benchmark results since these do not comply with the TPC-H specifications.

Thanks.