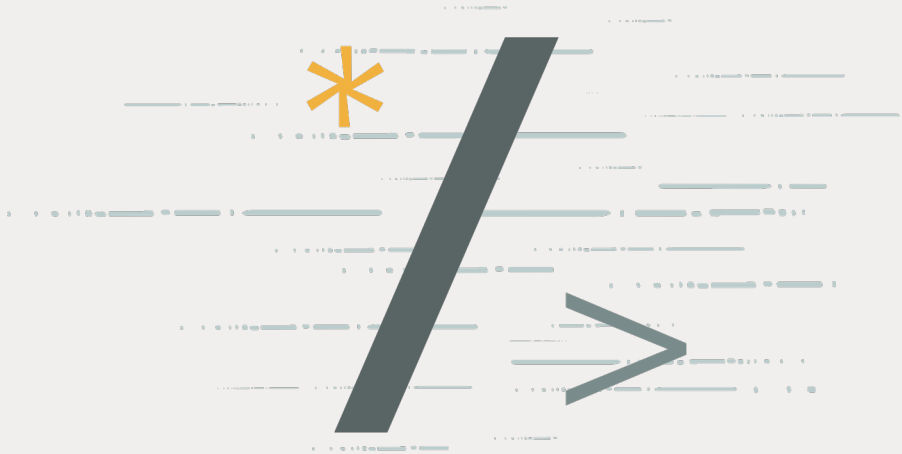
Data Warehouse and Lakehouse Analytics at the Speed of Thought with MySQL HeatWave

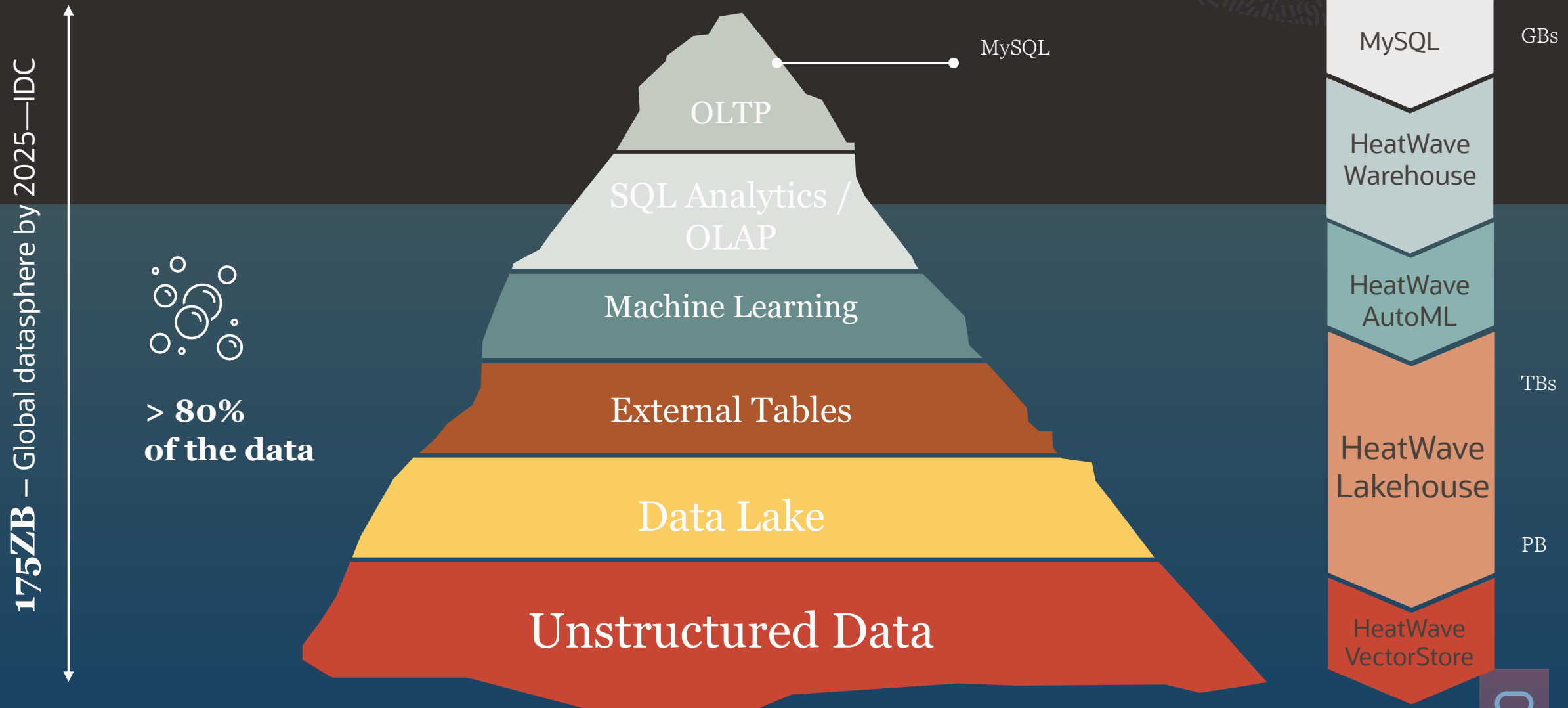
Gaurav Chadha
Senior Development Manager
MySQL HeatWave
May 1, 2024

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, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

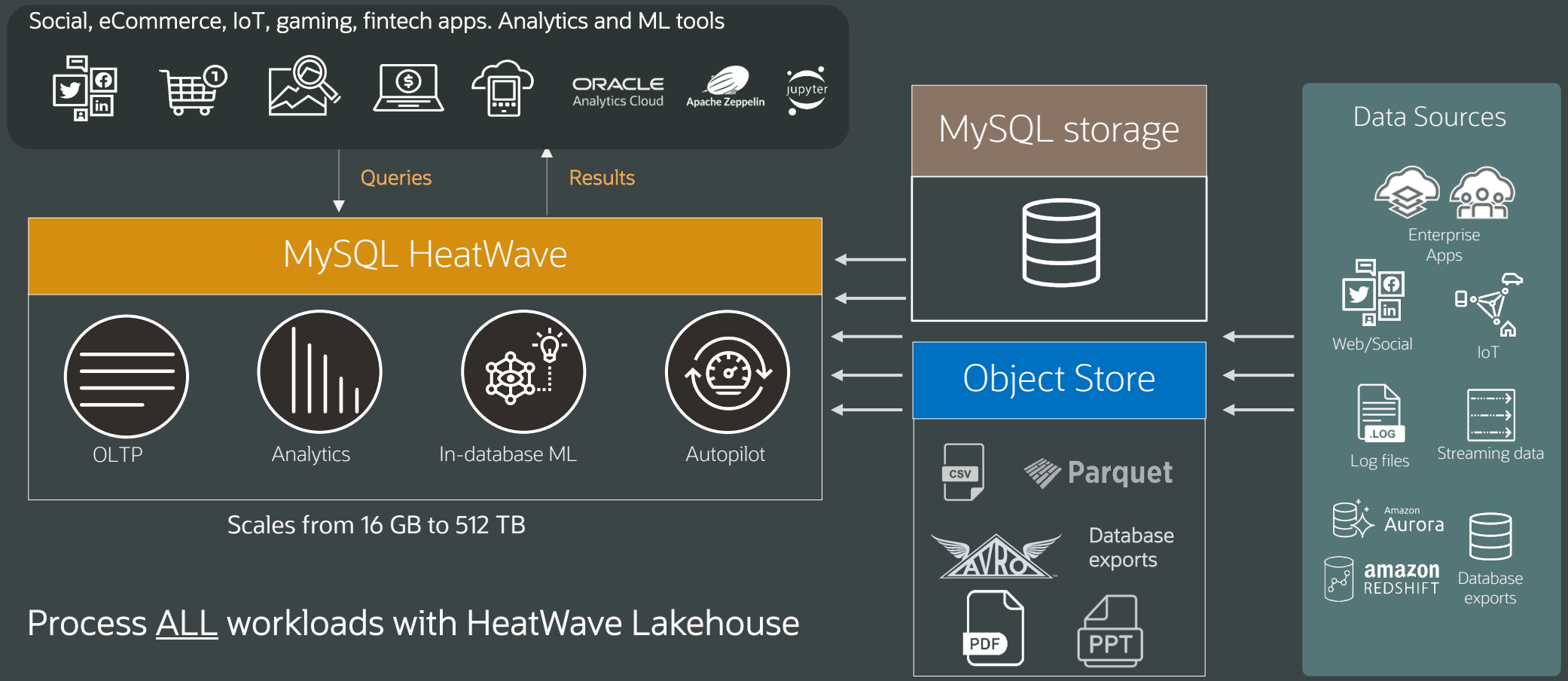


Data comes in different flavors and volumes



MySQL HeatWave

TRANSACTIONS, REAL-TIME ANALYTICS ACROSS DATA WAREHOUSE AND DATA LAKE, AND MACHINE LEARNING IN ONE DATABASE SERVICE



Process ALL workloads with HeatWave Lakehouse

Lowest cost in industry for data warehouse

Price performance comparison 10TB TPC-H

23X	27X	27X	60X
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.

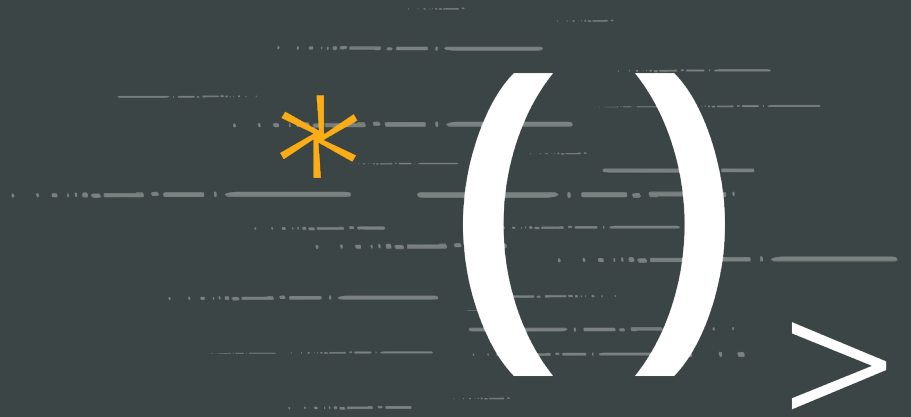
Analytic functions – CUBE, HLL

FACILITATES MIGRATION OF NON-MYSQL WORKLOADS

Operator	Snowflake	AWS Redshift	Google BigQuery	Databricks	PostgreSQL	MySQL HeatWave
CUBE	✓	✓	X	✓	✓	✓
HLL_COUNT	✓	✓	✓	✓	✓	✓
Grouping Sets	✓	✓	X	✓	✓	✓
Qualify	✓	✓	✓	✓	X	✓
Table Sample	✓	X	✓	✓	✓	✓

99.5%

99.5% of collected data remains unused



HeatWave Lakehouse table interface

Easy interface for data in object store as external table

- Provides Lakehouse-specific functionality with existing syntax and is extensible

External source file locations specified in extensible JSON interface

- Files can be distributed across multiple object store buckets

100% compliant with standard MySQL syntax

```
> CREATE TABLE tbl_name <create_definition> ENGINE=LAKEHOUSE
  ENGINE_ATTRIBUTE='<engine_options>'
  SECONDARY_ENGINE=RAPID;
```

MySQL Autopilot - Auto Parallel Load in Action

Automatically generated from files

```
# Load Script
CREATE DATABASE `tpch_500T`

CREATE TABLE `tpch_500T`.`lineitem`
(
    `col_1` int unsigned NOT NULL,          `col_2` mediumint unsigned NOT NULL,
    `col_3` mediumint unsigned NOT NULL,    `col_4` tinyint unsigned NOT NULL,
    `col_5` tinyint unsigned NOT NULL,      `col_6` decimal(8,2) NOT NULL,
    `col_7` decimal(3,2) NOT NULL,          `col_8` decimal(3,2) NOT NULL,
    `col_9` varchar(1) NOT,                 `col_10` varchar(1) NOT NULL,
    `col_11` date NOT NULL,                 `col_12` date NOT NULL,
    `col_13` date NOT NULL,                 `col_14` varchar(17),
    `col_15` varchar(7),                    `col_16` varchar(43),
    `col_17` varchar(0)
)

ENGINE=lakehouse
SECONDARY_ENGINE=RAPID
ENGINE_ATTRIBUTE='{"file":
    [{"name": "lineitem.tbl", "bucket": "tpch_500T", "region": "us-ashburn-1", "namespace": "mysql"}],
     "dialect": {"format": "csv", "field_delimiter": "|", "record_delimiter": "\\n"}}';

ALTER TABLE `tpch_500T`.`lineitem` SECONDARY_LOAD;
```

DDL to create non-existing DBs

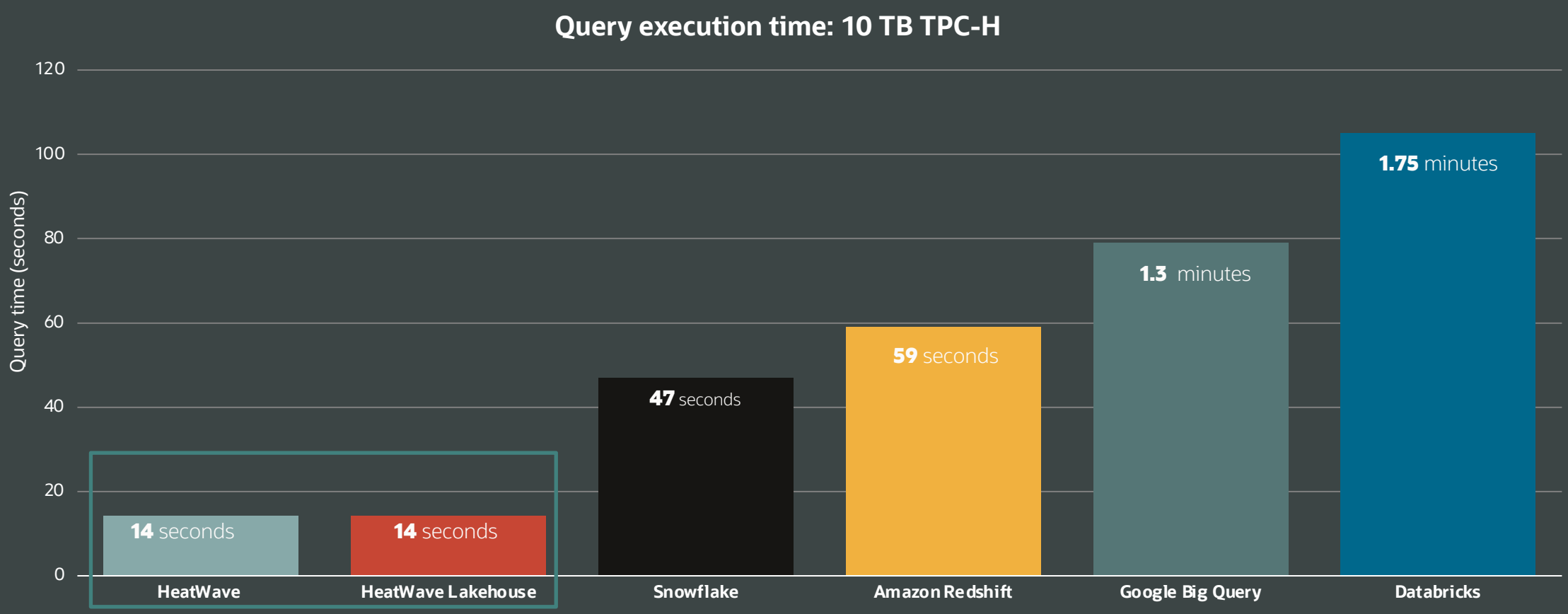
DDL to create non-existing tables

- Using inferred column types
  - Length
  - Precision
- Setting engines
- Setting engine attribute
- Can extract column names

Load command

Same performance for data in DB or in object store

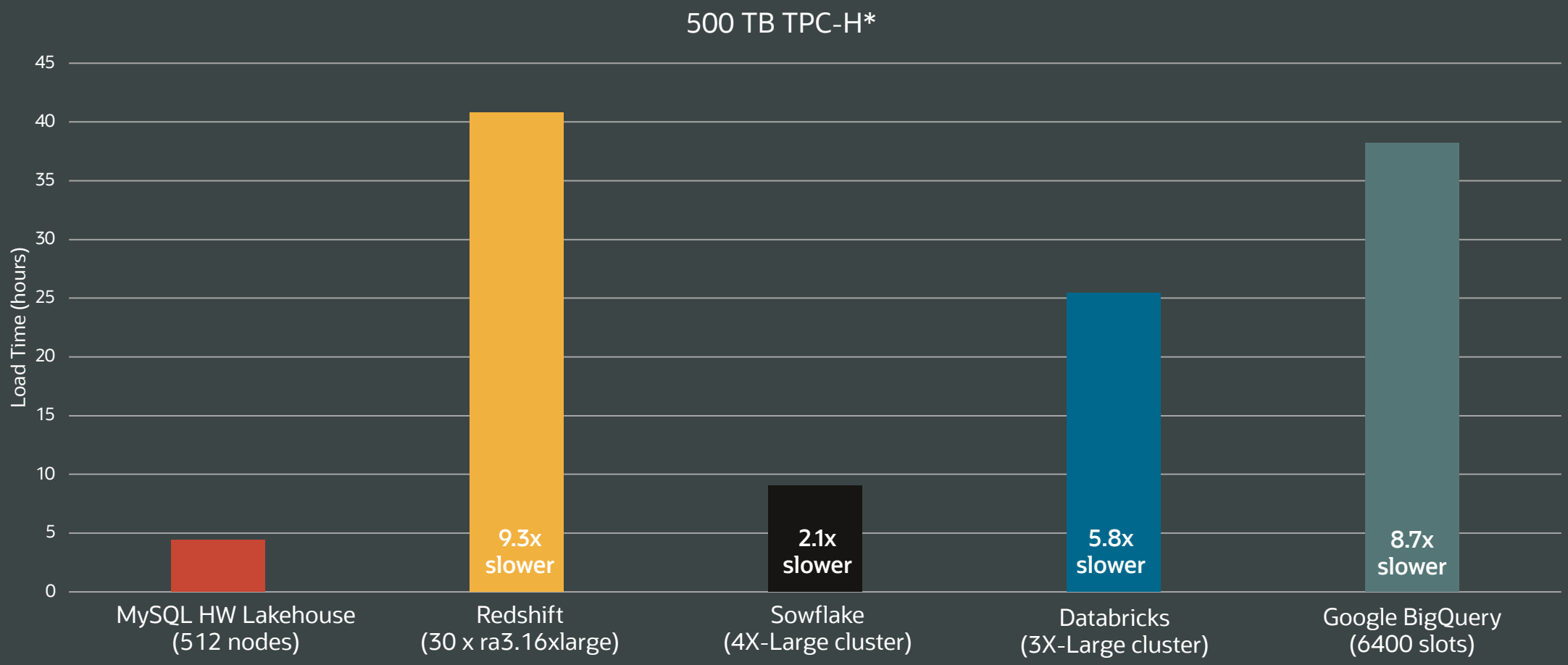
Develop applications with data on object store without any performance impact



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.

HeatWave Lakehouse scales all the way to 500 TB



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

HeatWave Lakehouse extends support to semi-structured data

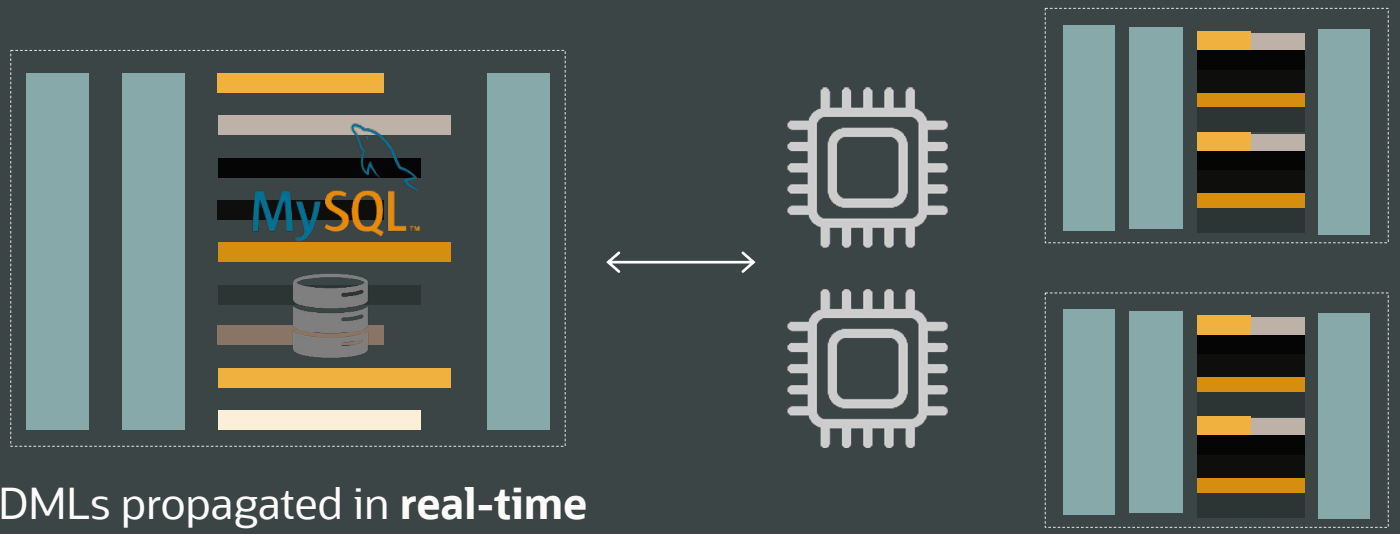
- **JSON** data in **CSV**, **Parquet**, and **Avro** file formats can now be processed by HeatWave
- Support extended to newline-delimited JSON files
  - Ease of parsing and streaming has made it the most popular JSON format
- **NDJSON** data ingestion and processing scales similarly to structured file formats


```
…
{ “name”: “Jane”, “academics”: { "undergraduate": "MIT", "graduate": "UT Austin” }, "age": 24 }
{ “name”: “Jill”, “academics”: { "undergraduate": ”Madison", "graduate": ”Stanford” }, "age": 27 }
…
```

Example NDJSON file

JSON acceleration with HeatWave

Query processing and real-time analytics on JSON documents



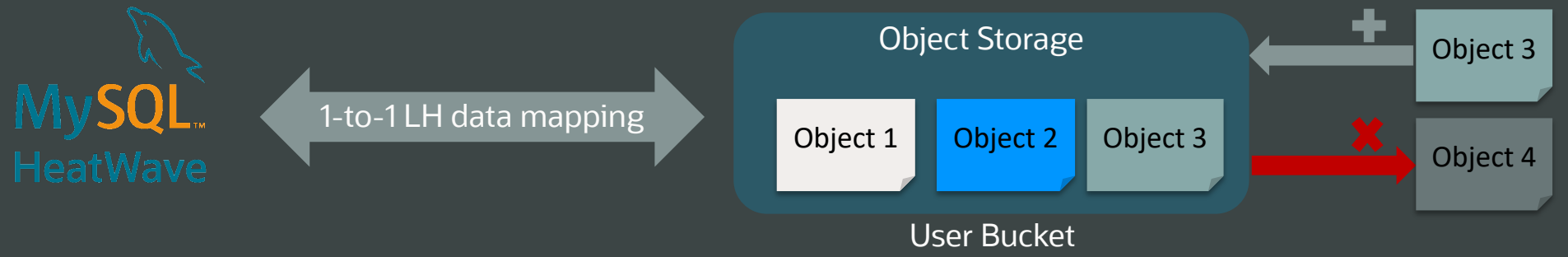
DMLs propagated in **real-time**

- Data compressed up to **3X**
- Scales across nodes

JSON Queries (512 GB)	MySQL (sec)	HeatWave (sec)	Speedup
Simple Filter Queries	5200	240	**20x**
Aggregation Queries	5500	250	**22x**
Large Join Queries	>10 hrs	300	**144x**

Incremental data load in Lakehouse tables

Features



- **Feature**: Lakehouse table data is updated to reflect modifications in user data
  - Provides 1-to-1 mapping between user data and Lakehouse table data at any point in time
  - Only delta in user data is applied incrementally over existing table data
  - Incremental load triggered manually through a SQL command

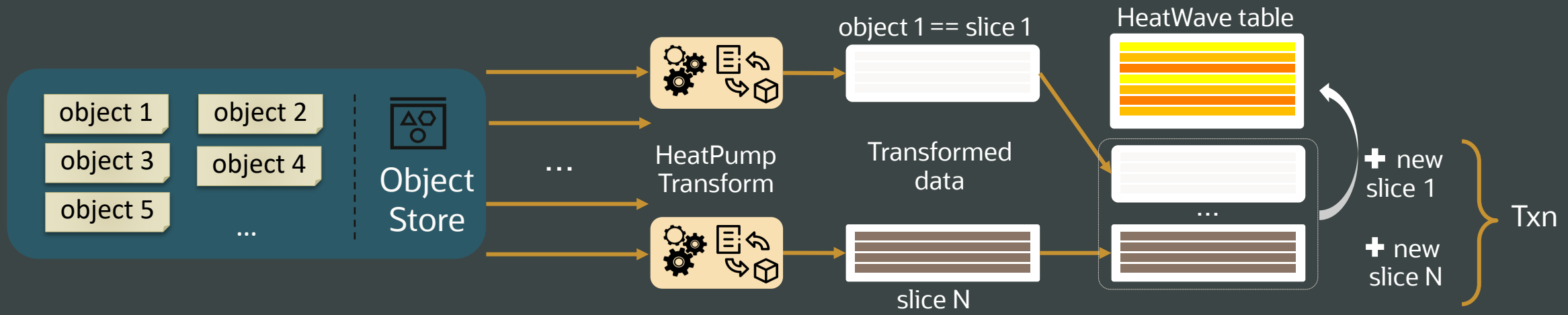
- **Read-committed** & **snapshot isolation**: Queries on Lakehouse tables are never blocked
  - Queries are run on the version of the data which is committed as of the query start time

- Integrated into existing AutoLoad interface

```
SET @options = JSON_OBJECT('mode', 'normal',
                           'refresh_external_tables', TRUE);
SET @heatwave_debug_output = TRUE;
CALL sys.heatwave_load(@db_list, @options);
```

Incremental data load in Lakehouse tables

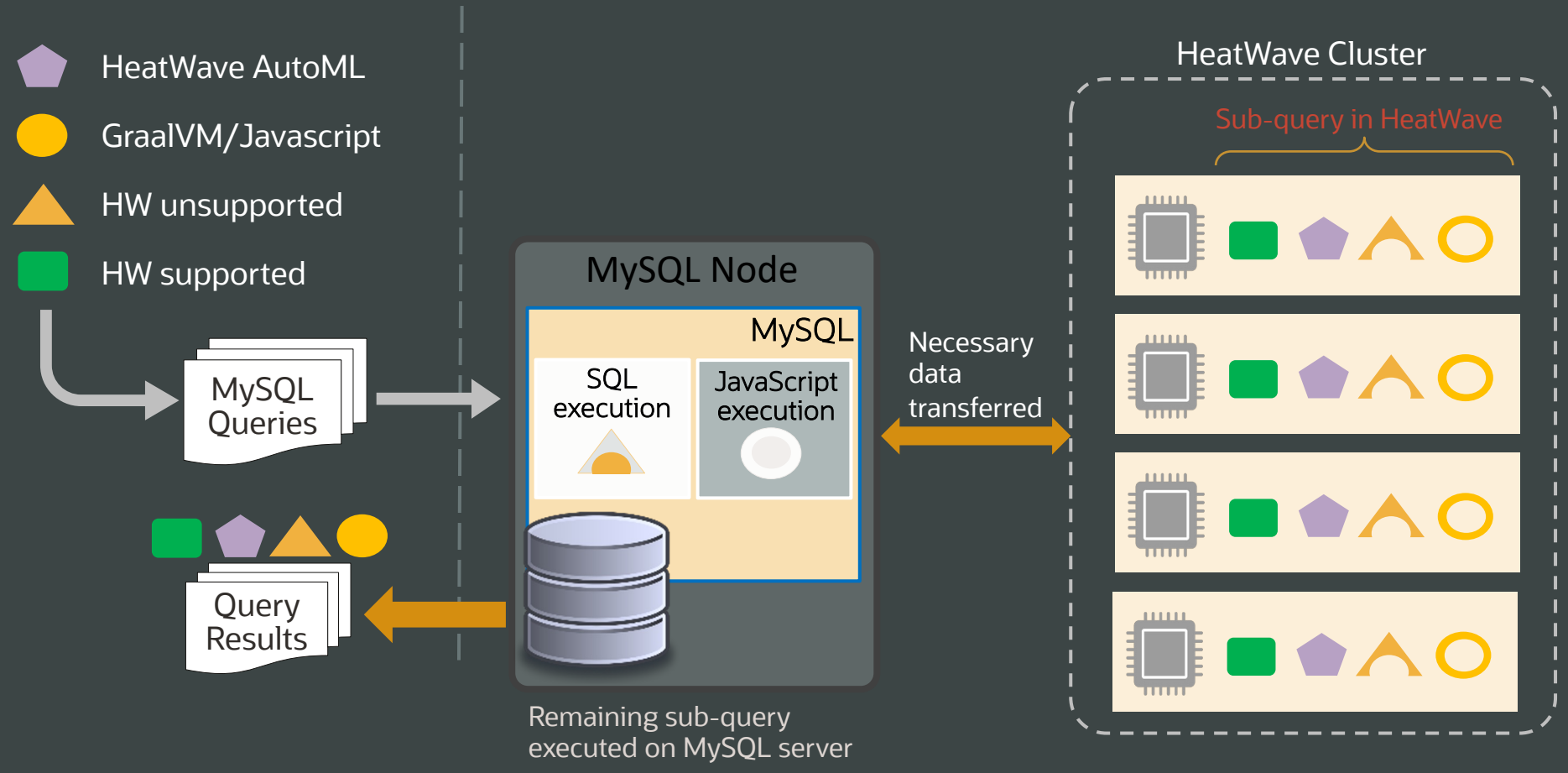
Scale-out delta ingestion



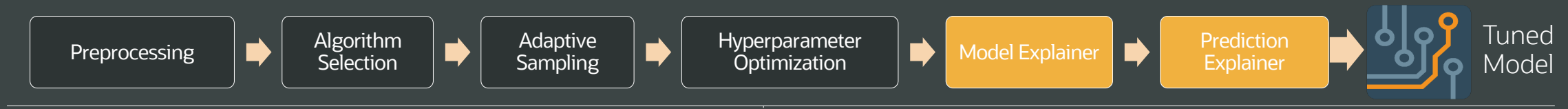
- **Granularity of data update** is an object corresponding to thousands of records
- **User data change detection**: On user-initiated SQL command, user data change is detected
  - Objects in user buckets can be **added**, **deleted**, or **updated**
  - Delta computed comparing current list of objects with the list from the last table load or incremental load
- **Delta apply design**: Treat each object as a new horizontal slice of the table
  - Objects added or updated are transformed and ingested in a scale-out manner across HeatWave cluster like table load
  - Bulk-inserts scale: HeatPump parallelism at inter-file & intra-file levels
  - Objects deleted – fast in-memory operation of dropping a table slice by updating table version

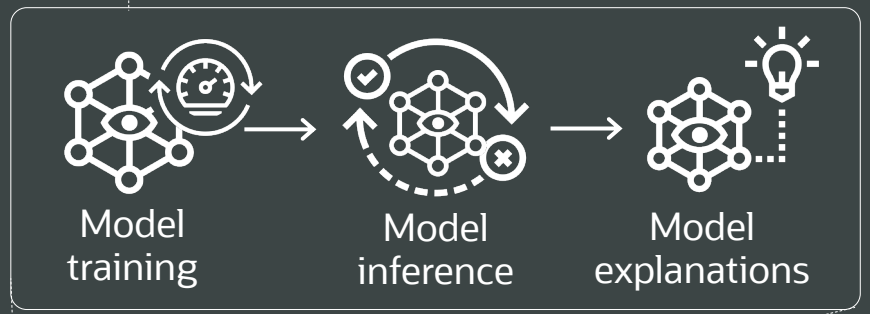
Partial query execution in HeatWave for data in object store

Execute part of the query in HeatWave, rest in MySQL



HeatWave AutoML: In-database machine learning





In-database ML

- Eliminates tedious and laborious steps
- Simple to use interface for beginners or advanced ML users
- Automatically selects algorithm and tunes it
- Explainable model behavior and predictions
- Fast training allows to quickly iterate and achieve desired outcome
- ML on data in InnoDB and Object Store (Lakehouse)

Native Vector Processing in MySQL HeatWave

Vector Datatype

- MySQL & HeatWave supports new Vector data type
- In-memory hybrid-columnar storage format for vector columns

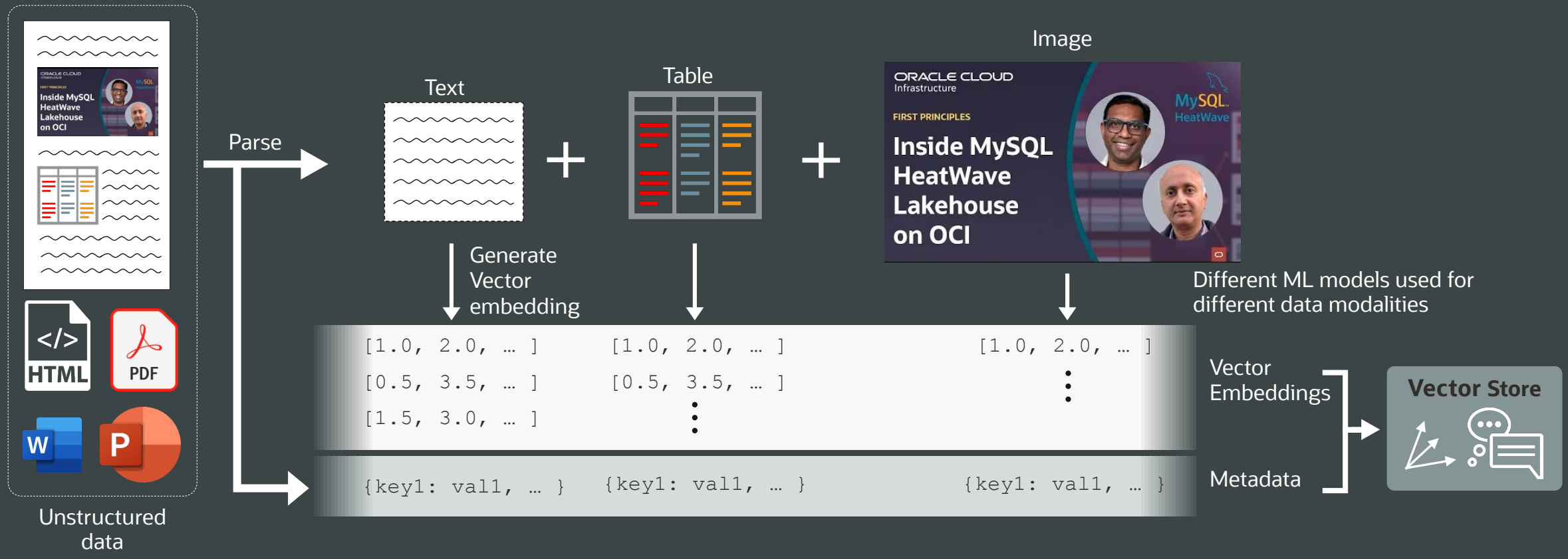
Vector Processing

- Leverage SIMD instructions for vector processing
- Processes at near memory bandwidth

Data Management

- End to end data management including embedding generation
- Integrated with features like in-bound replication

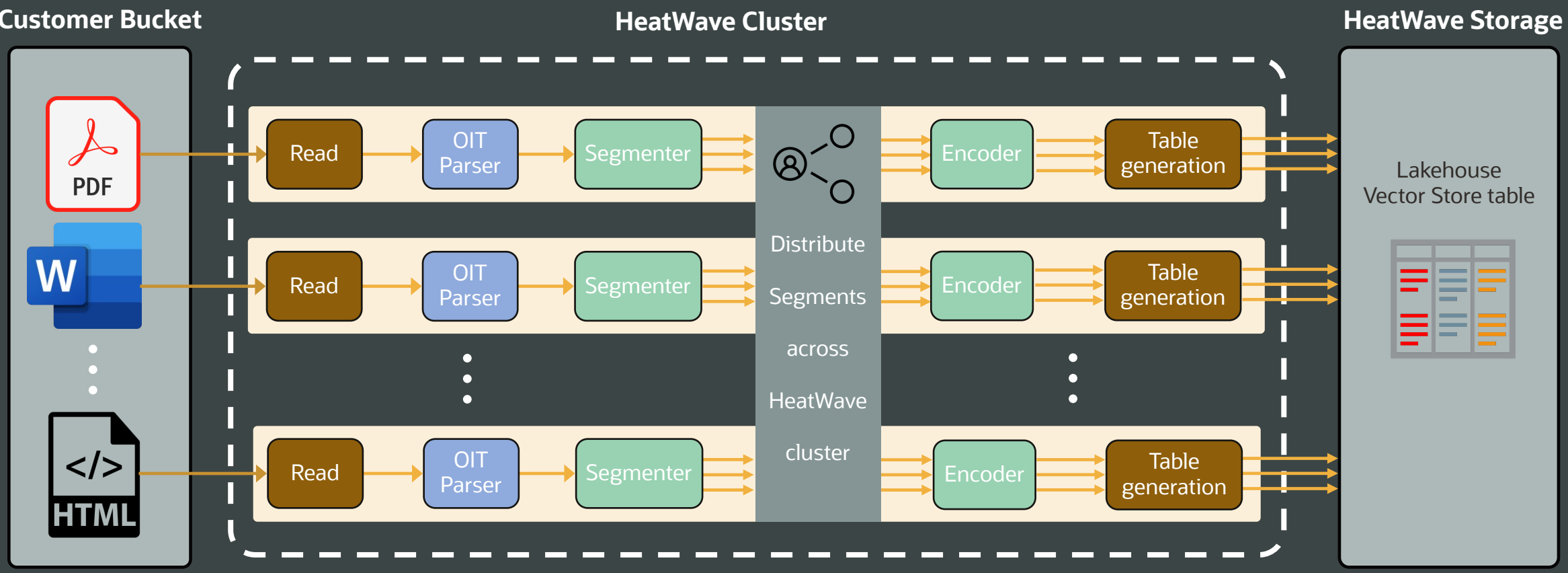
Unstructured data is transformed in HeatWave Vector Store



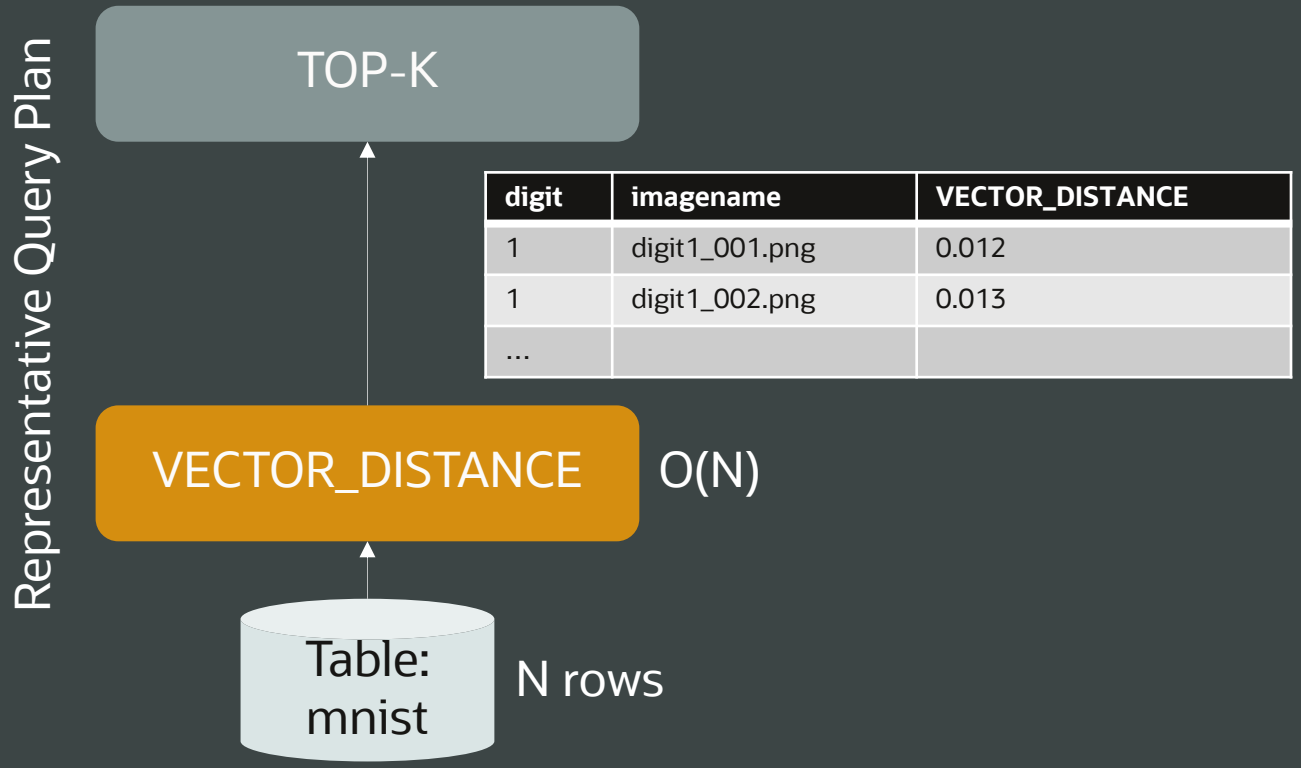
Automatically generate embedding for text from multiple file formats

Scale out Vector Store creation with HeatWave Lakehouse

Parse source files with OutsideIn (OIT) and concurrent embedding generation across nodes



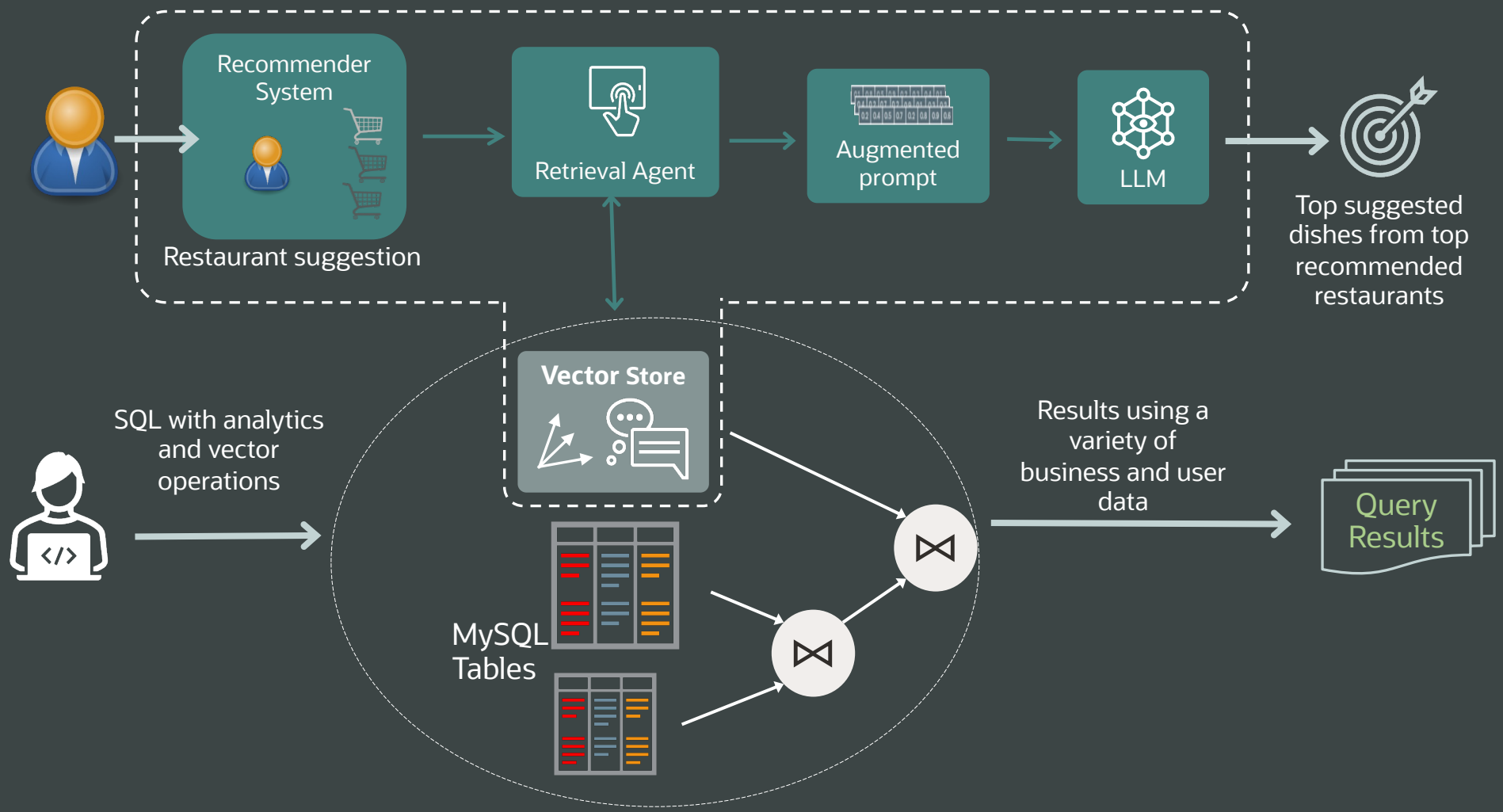
Exact Nearest Neighbor Search using SQL



digit	imagename	VECTOR_DISTANCE
1	digit1_001.png	0.012
1	digit1_002.png	0.013
...		

```
SELECT digit, imagename FROM mnist
ORDER BY VECTOR_DISTANCE(embedding,
@query_embedding)
LIMIT 3;
```

Vector Store can be used by SQL queries, or for RAG



Using HeatWave Vector Store

Create Vector Store

```
# Ingest documents from Object Store like any Lakehouse table
CALL sys.heatwave_load(“vector_store”, @load_params);
```

Query Vector Store
Native SQL syntax

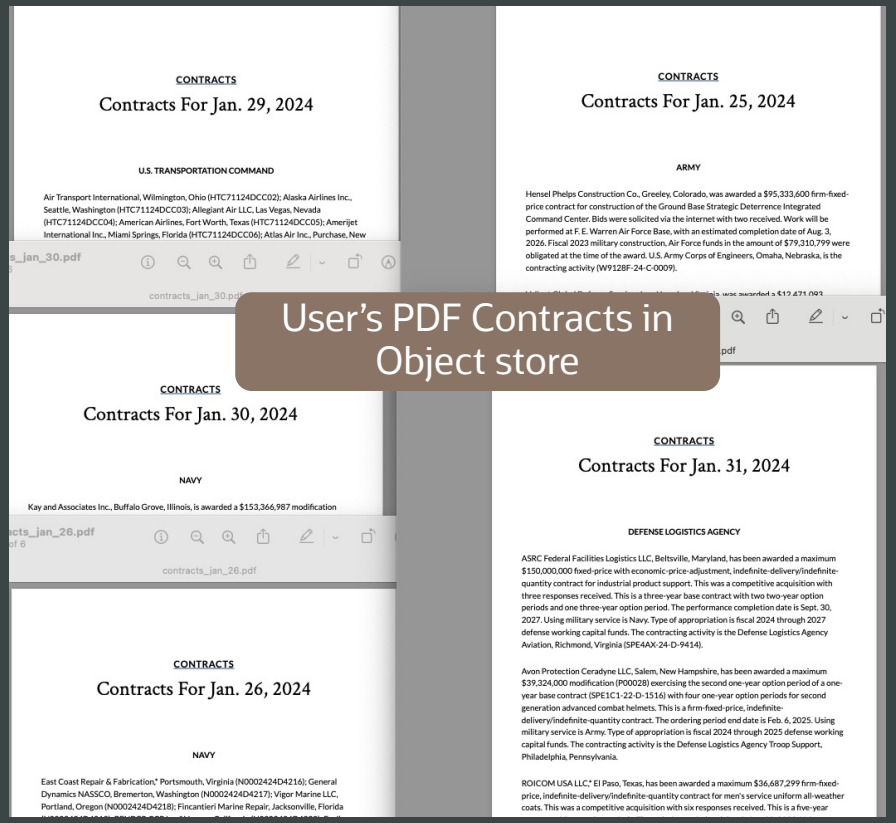
```
# Example: Find books semantically most similar to input and are in print
SELECT id, title
FROM books b, books_in_print ip
WHERE b.title = ip.title
ORDER BY DISTANCE(b.segment_embedding, @query_embedding, "DOT") as distance
DESC LIMIT 10;
```

Query Vector Store
ML_RAG

```
# Example: Answer questions using data in documents ingested into Vector Store
CALL sys.ML_RAG("Which state has maximum carbon?", @output);
```

Example: Employee Assistant

Natural language



User's PDF Contracts in Object store

Vector store ingest

Employee Assistant: Improve employee productivity

```
ask_question(query='Which firm was awarded a contract for Dell laptops?', color='blue')
```

✓ 15.7s Python

MCP Computer Products was awarded a contract for Dell laptops. Would you like to know more about this company?

```
ask_question(query='Were there any contracts for the Rolling Airframe Missile?', color='blue')
```

✓ 10.3s Python

Yes, Raytheon Co. was awarded a $13.4 million cost-plus-fixed-fee modification contract for design and engineering support services for the Rolling Airframe Missile. This was awarded on January 25, 2024, and work is expected to be completed by March 2028.
Would you like to know more about this contract?

```
ask_question(query='Were there any contracts mentioning Lassa Fever?', color='blue')
```

✓ 14.2s Python

Yes, Primmune Therapeutics Inc. based in San Diego, California, was awarded a contract worth $22,480,552 for the development of an oral broad-spectrum antiviral TLR7 agonist for the treatment of Lassa Fever. The contract is identified as HDTRA1-24-C-0014 and was awarded by the Defense Threat Reduction Agency (DTRA).
Would you like to know more about this contract?

Retrieval Augmented Generation

Addressing customer challenges with MySQL HeatWave

Expensive and slow to analyze growing data stored in files

Best performance and price-performance to query non-MySQL and MySQL workloads

Want to leverage machine learning and Gen AI on all their data

Fully automated ML for data in the database and in files. Gen AI with vector store

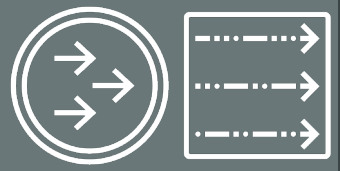
Complex and costly to use separate cloud services for OLTP, analytics, ML

One cloud service for OLTP, real-time analytics, and ML

Want flexibility to use multiple public clouds

Available in OCI, AWS, and Azure

IoT Events



- Sensors record temperature, torque, spin, humidity
- Each sensor sends data back via telemetry
- Data in CSV/JSON
- Written as separate files
- Read-only data

Metrics

- Anomalous hotspots
- How many parts have variances out of range
- Do I have that part in stock?
- Which vendor is likely to fulfil the order the fastest
- Which parts are likely to fail
- What is the impact of a parts failure

Challenges

- Ingest terabytes of data, thousands of files
- Different file formats
- Query should take seconds or minutes—not hours
- Configure a new ML service?
- Get ERP, SCM data

“HeatWave Lakehouse allows us to easily and quickly load data on object storage into HeatWave and combine it with MySQL data for analysis.”

Takashi Kinoshita
Chief Producer, e-Book Division
NTT SOLMARE CORPORATION

Industry analysts about MySQL HeatWave Lakehouse

“Organizations looking for the best value in the cloud data lakehouse landscape must seriously consider MySQL HeatWave Lakehouse.”
—Carl Olofson, Research Vice President, Data Management Software

“MySQL HeatWave demonstrates **that Lakehouse performance can be identical to transaction query performance—unheard of and even unthinkable.”**
—Holger Mueller, VP and Principal Analyst

“The ability of HeatWave to **load and query data on such a massive number of nodes in parallel is the first in the industry.”**
—Marc Staimer, Senior Analyst

“MySQL HeatWave Lakehouse can **simplify the life of data management professionals** and should **improve the customer experience**.”
—Matt Kimball, Vice President and Principal Analyst

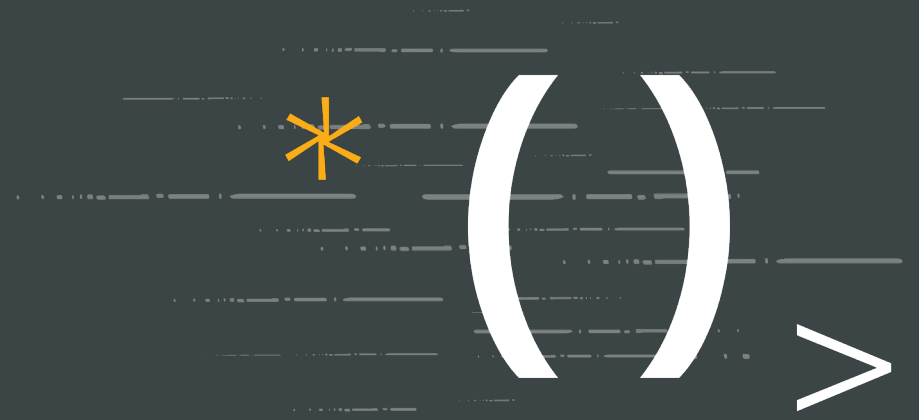
“Simply put: MySQL HeatWave Lakehouse **enables you to stay ahead of the competition by taking swift action on meaningful business insights.”**
—Steve McDowell, Principal Analyst & Founding Partner

Resources

- Web
  - Oracle.com/heatwave
- YouTube
  - youtube.com/@mysql
- Blog
  - https://blogs.oracle.com/mysql/
- Documentation
  - https://dev.mysql.com/doc/heatwave/en/
- Technical white paper
  - Technical Solution Brief
- Hands-on lab
  - New MySQL HeatWave Lakehouse Hands-on Lab
- Certification
  - https://education.oracle.com/
- Free trial
  - https://cloud.oracle.com

Media coverage

- Forbes: Oracle Outperforms Databricks, Snowflake and BigQuery
- The 65 Podcast: MySQL HeatWave Lakehouse is "Tremendously Powerful and Incredible"
- Futurum: MySQL Delivers New Competitive Level Set
- TechTarget: “HeatWave should win the Oscar for Fastest Innovation”
- Venture Beat: MySQL HeatWave goes GA to Query Data
- Chat GPT Global: Oracle Unleashes the Power of MySQL Heatwave Lakehouse for Efficient Data Queries
- The Register: MySQL HeatWave dives into object storage data lakes



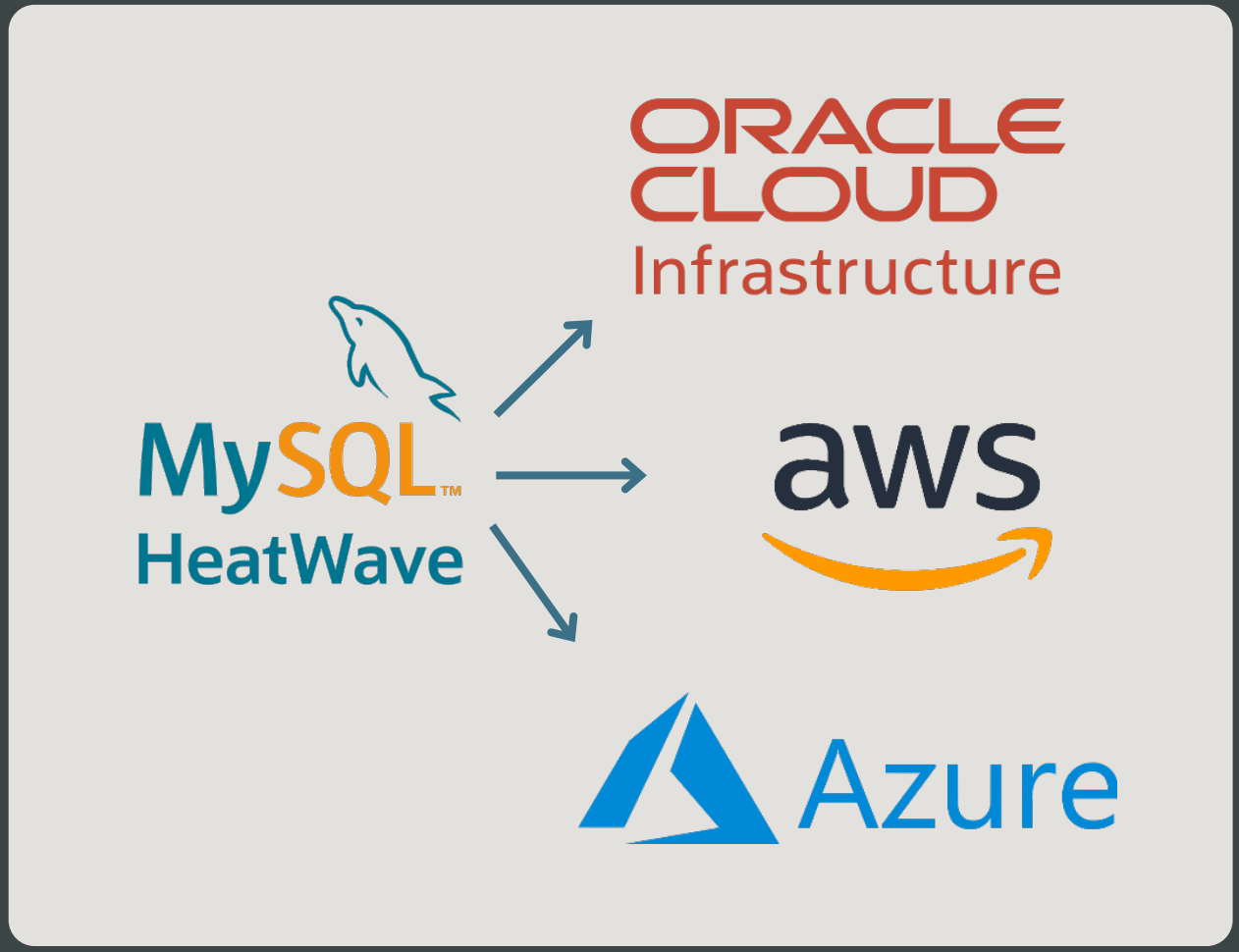
Thank you


ORACLE

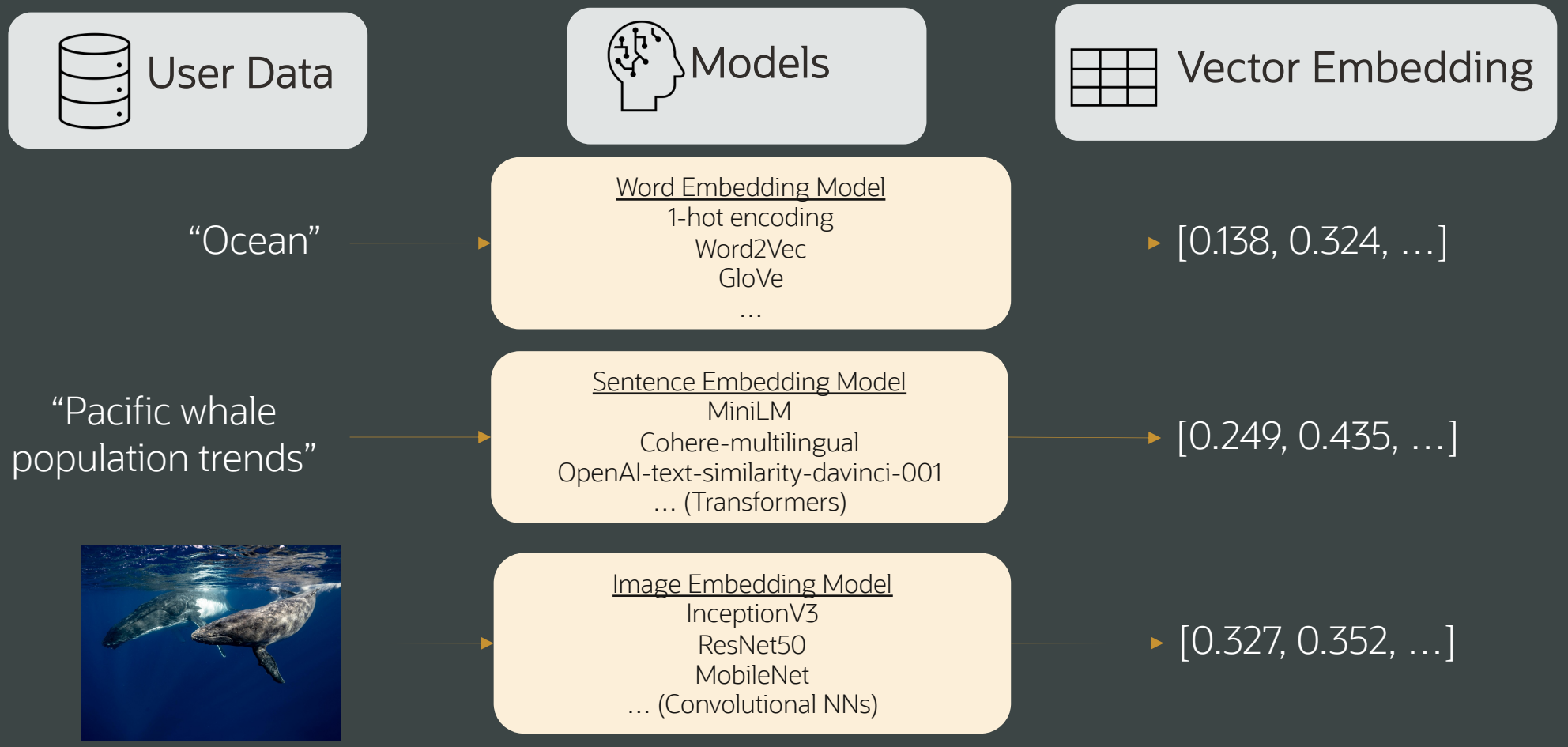
Our mission is to help people see data in new ways, discover insights, unlock endless possibilities.

MySQL HeatWave is optimized for multiple clouds

Maximum flexibility and choice



Vector is a compressed representation of data



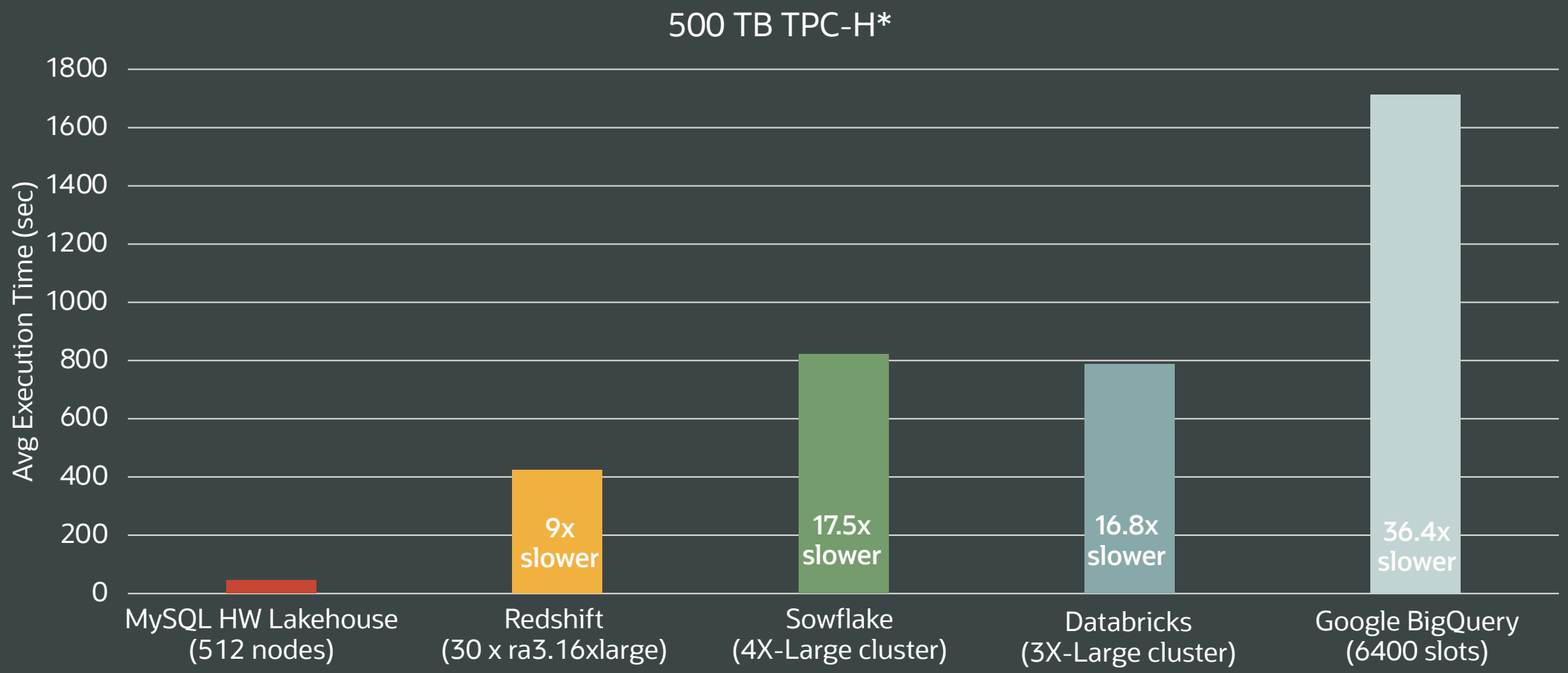
Entities that are similar/related will be closer in the latent space

Best performance in the industry for query and load at the lowest price
TPC-DS 100TB

TPC-DS 100TB	HeatWave	Snowflake 3XLarge	RedShift 10 ra3.16xlarge	BigQuery 3200 slots	Databricks 2XLarge
Hourly Cost ($)	56.43	128	86.06	74.56	103.39
Load time (hrs)	1.21	3.3	7.74	3.63	7.46
HeatWave Load advantage		**2.7x**	**6.4x**	**3x**	**6.1x**
Total Time (seconds)	3,719	5,379	5,108	11,694	13,704
Price-Perf ($)	58	191	122	242	394
HeatWave price-perf advantage		**3.3x**	**2.1x**	**4.1x**	**6.8x**

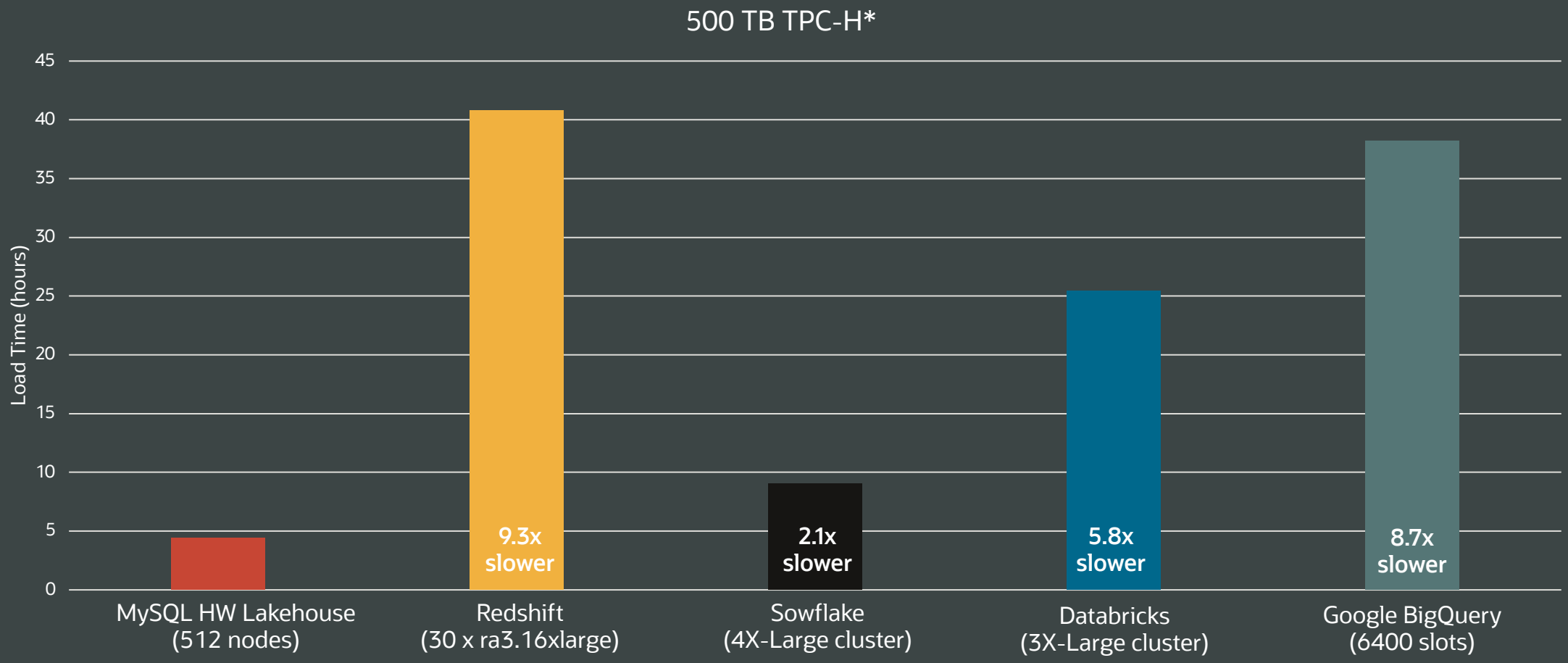
Benchmark queries are derived from the TPC-DS benchmarks, but results are not comparable to published TPC-DS benchmark results since these do not comply with the TPC-DS specifications.

Unmatched query performance



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

Data loading is much faster than the competition



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