

Data engineering and big data

UNDERSTANDING DATA ENGINEERING



About the course

- Conceptual course
- No coding involved
- **Objectives**
 - Being able to exchange with data engineers
 - Provide a solid foundation to learn more

Chapter 1

What is data engineering?

1. Data engineering and big data
2. Data engineers vs. data scientists
3. Data pipelines

Chapter 2

How data storage works

1. Structured vs unstructured data
2. SQL
3. Data warehouse and data lakes

Chapter 3

How to move and process data

1. Processing data
2. Scheduling data
3. Parallel computing
4. Cloud computing



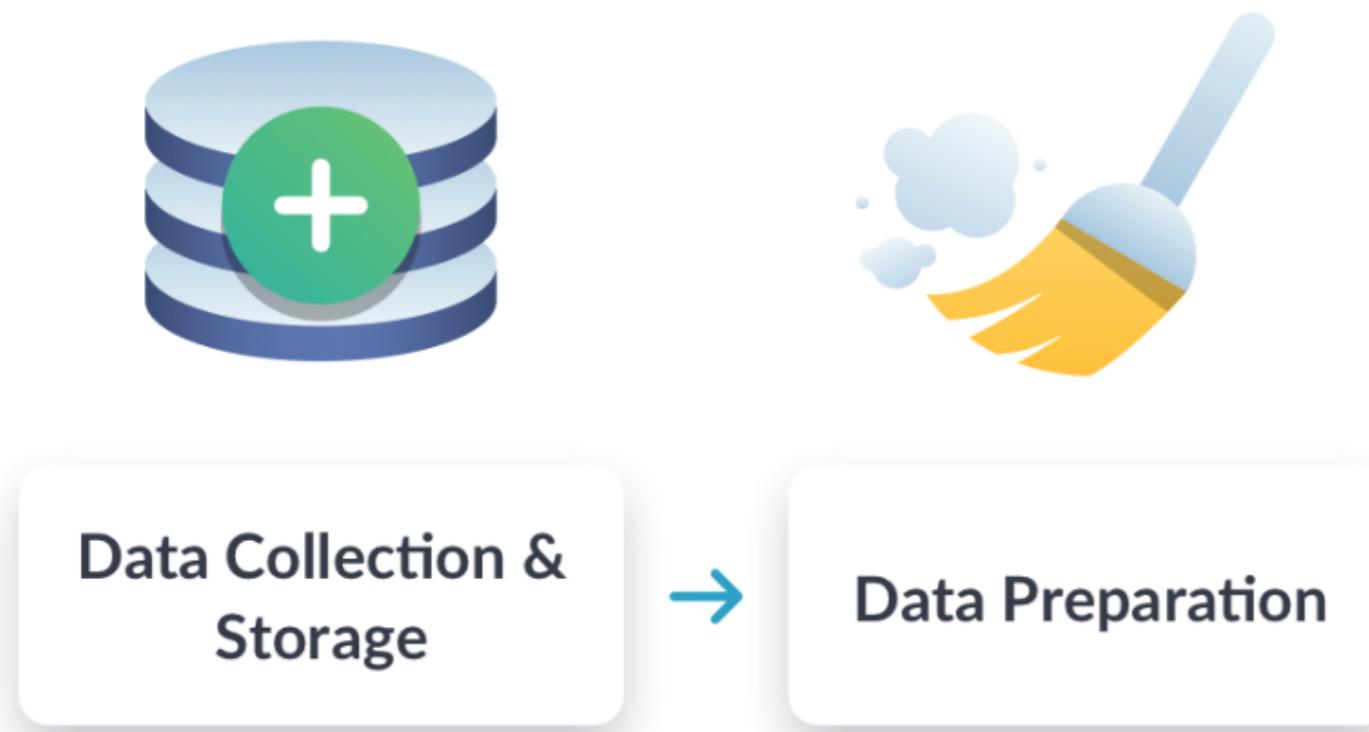
Spotfliix

Data workflow

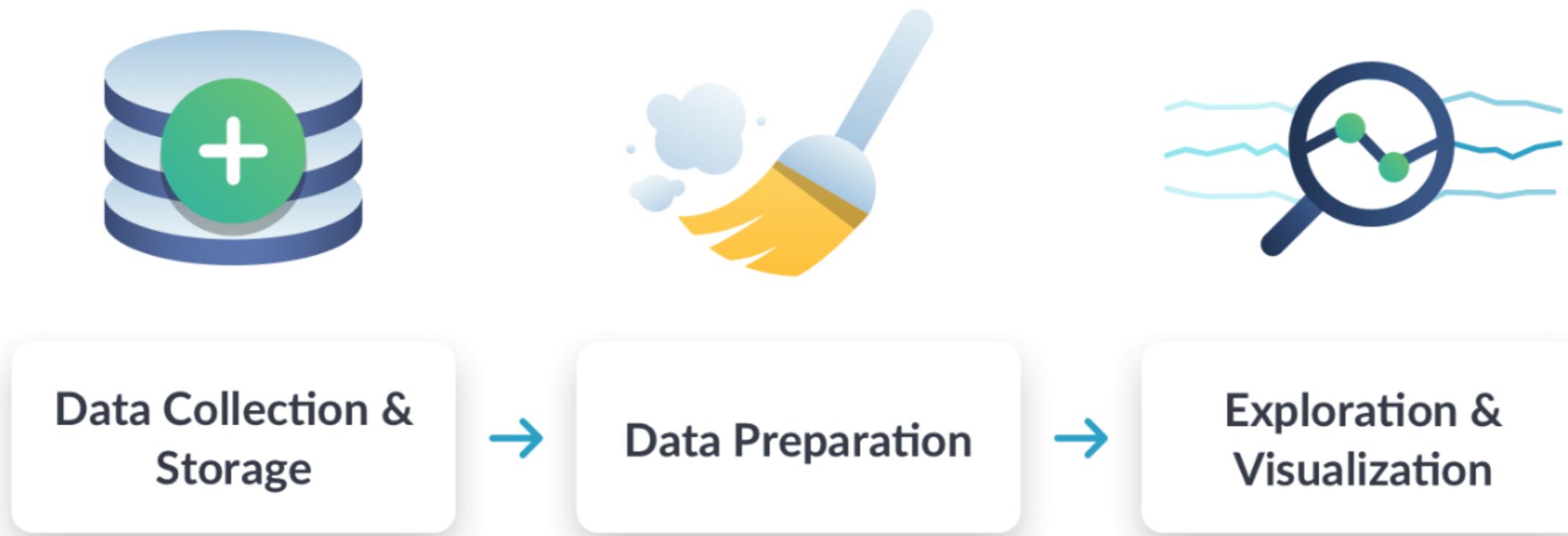


Data Collection &
Storage

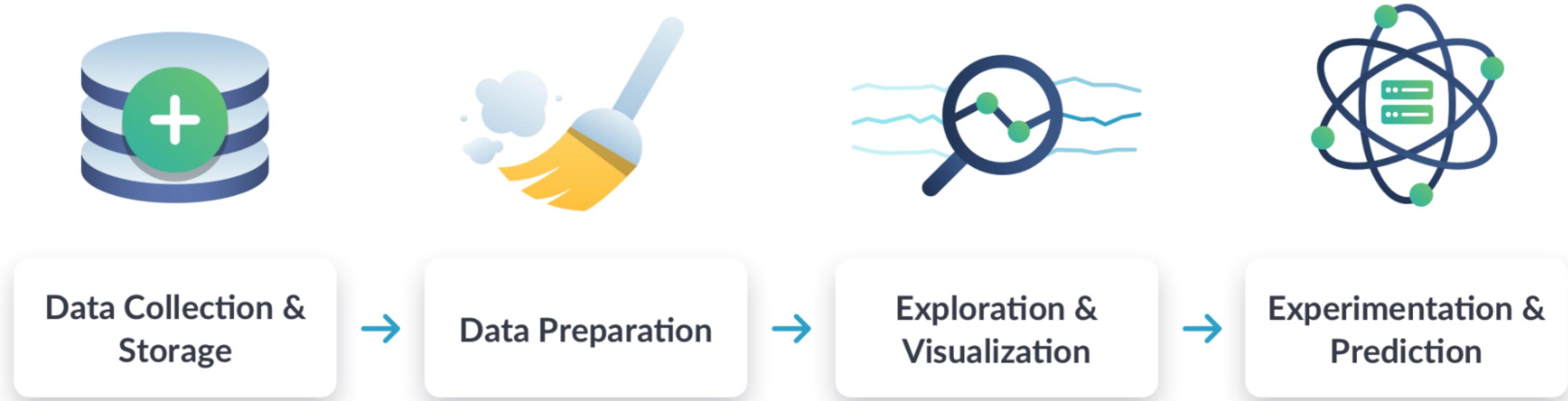
Data workflow



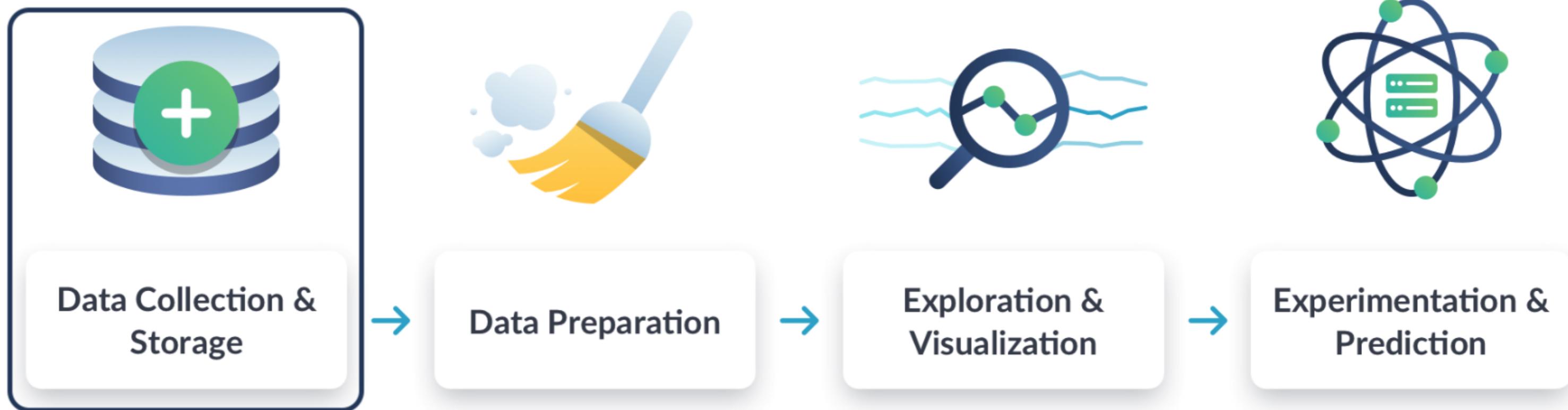
Data workflow



Data workflow



Data engineers



Data engineers

Data engineers deliver:

- the correct data
- in the right form
- to the right people
- as efficiently as possible

A data engineer's responsibilities

- Ingest data from different sources
- Optimize databases for analysis
- Remove corrupted data
- Develop, construct, test and maintain data architectures

Data engineers and big data

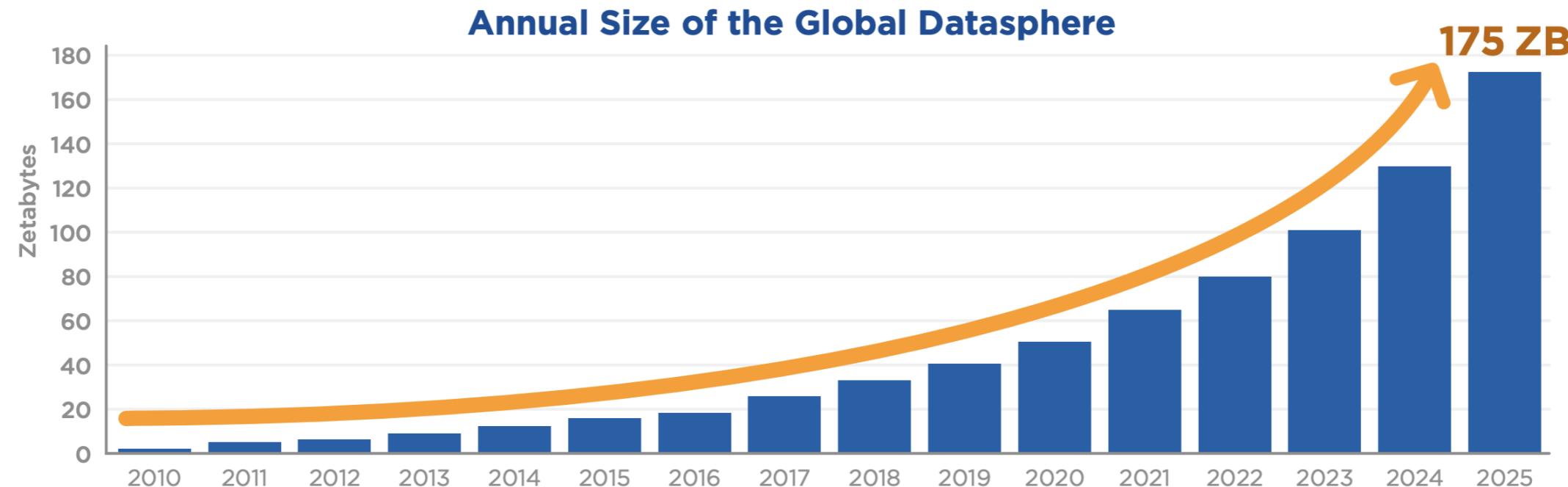
- Big data becomes the norm =>

Data engineers and big data

- Big data becomes the norm => data engineers are more and more needed
- Big data:
 - Have to think about how to deal with its size
 - So large traditional methods don't work anymore

Big data growth

- Sensors and devices
- Social media
- Enterprise data
- VoIP (voice communication, multimedia sessions)



¹ Data Age 2025, Seagate, November 2018

The five Vs

- Volume (how much?)
- Variety (what kind?)
- Velocity (how frequent?)
- Veracity (how accurate?)
- Value (how useful?)

Summary

- What's waiting for you
- How data flows through an organization
- When a data engineer intervenes
- What their responsibilities are
- How data engineering relates to big data

Let's practice!

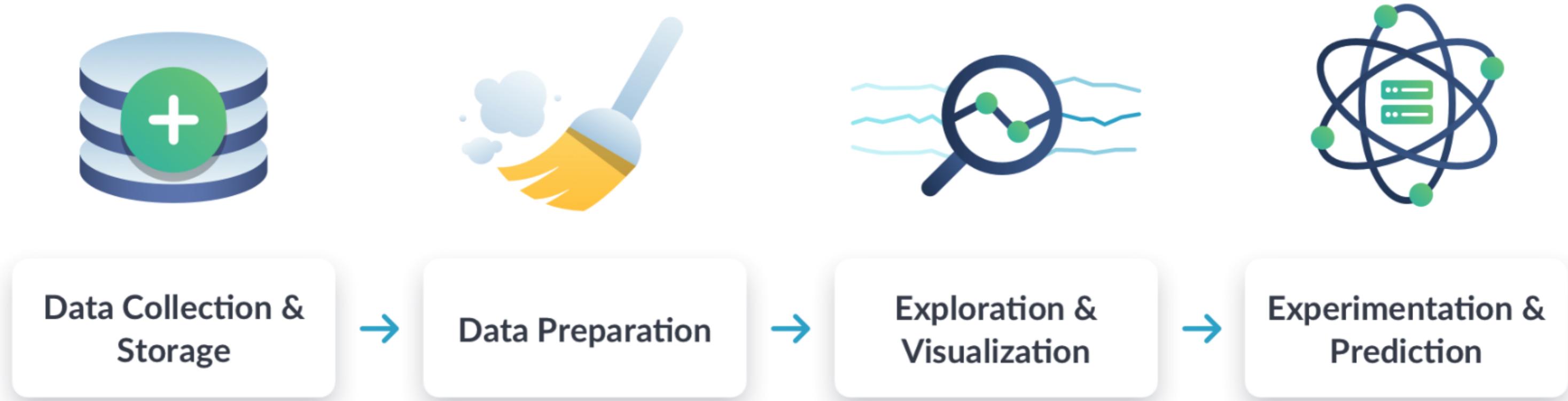
UNDERSTANDING DATA ENGINEERING

Data engineers vs. data scientists

UNDERSTANDING DATA ENGINEERING



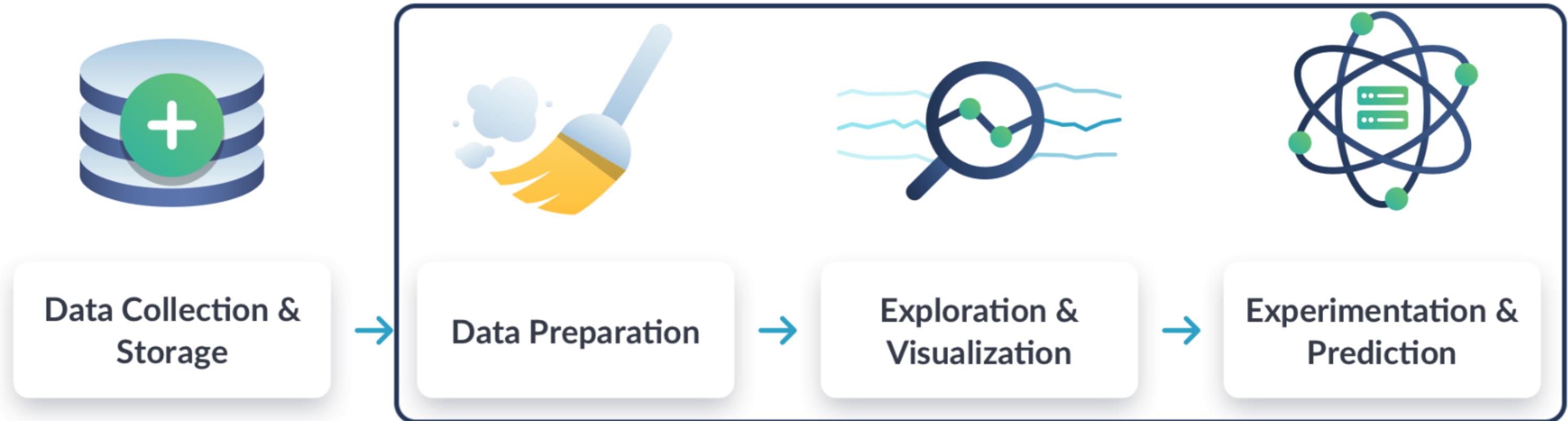
Data workflow



Data engineers



Data scientists



Data engineers enable data scientists

Data engineer

- Ingest and store data
- Set up databases
- Build data pipelines
- Strong software skills



Data scientist

- Exploit data
- Access databases
- Use pipeline outputs
- Strong analytical skills



Summary

- At which stages data engineers and data scientists intervene
- How data engineers enable data scientists

Let's practice!

UNDERSTANDING DATA ENGINEERING

The data pipeline

UNDERSTANDING DATA ENGINEERING



If data is the new oil...



¹ The Economist, 2017-05-06, by David Parkins



UNDERSTANDING DATA ENGINEERING



UNDERSTANDING DATA ENGINEERING



UNDERSTANDING DATA ENGINEERING



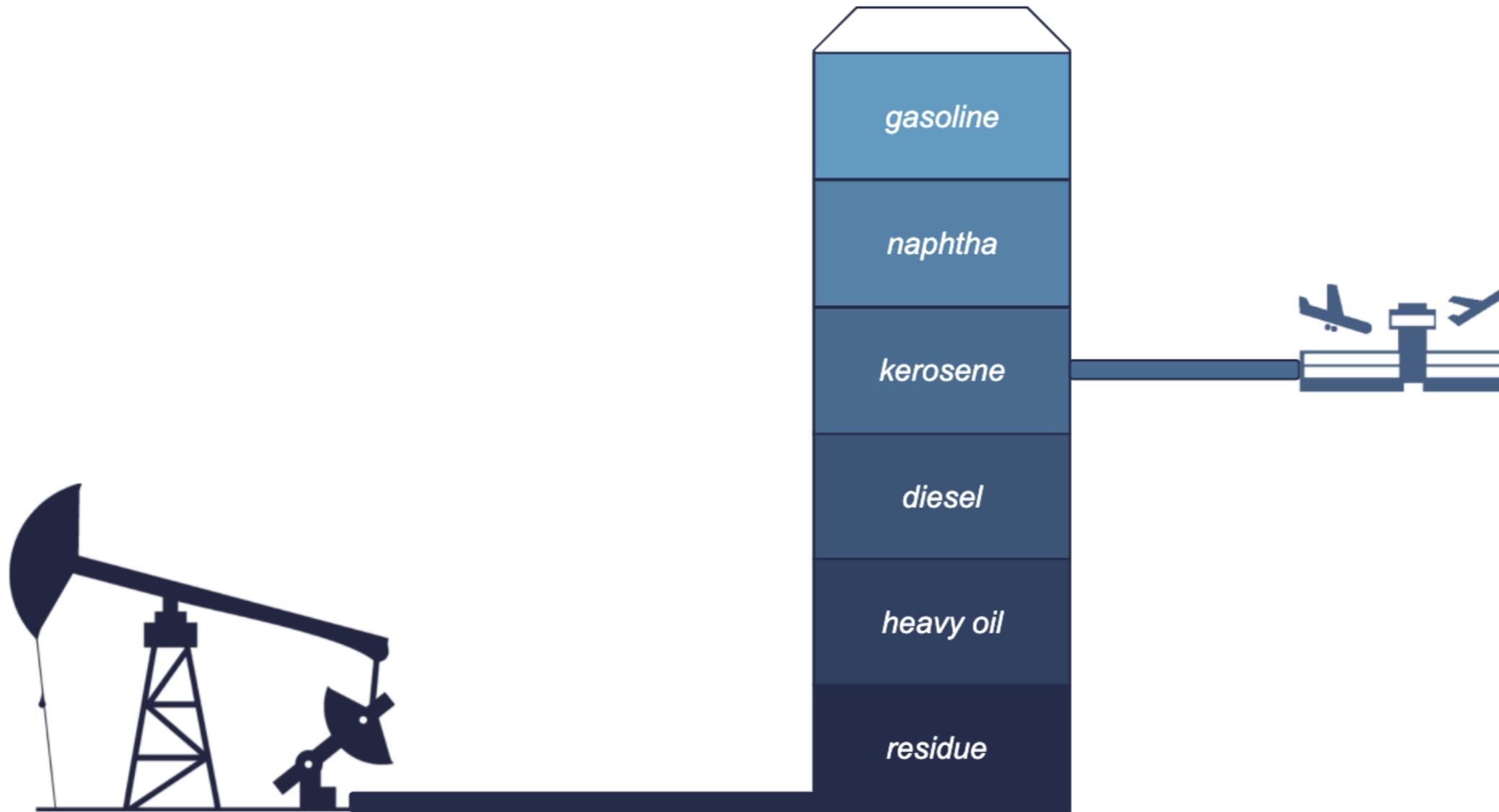


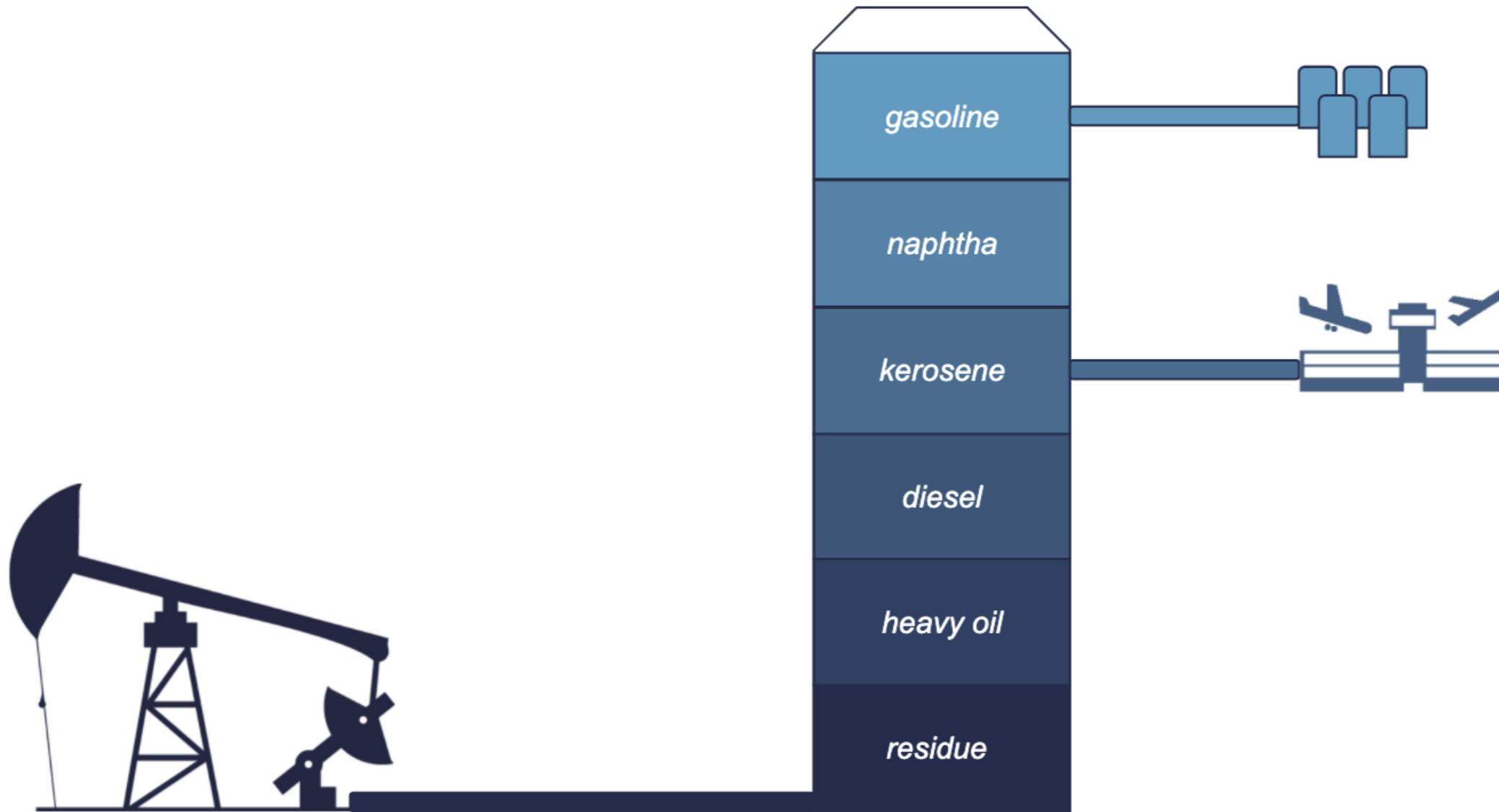


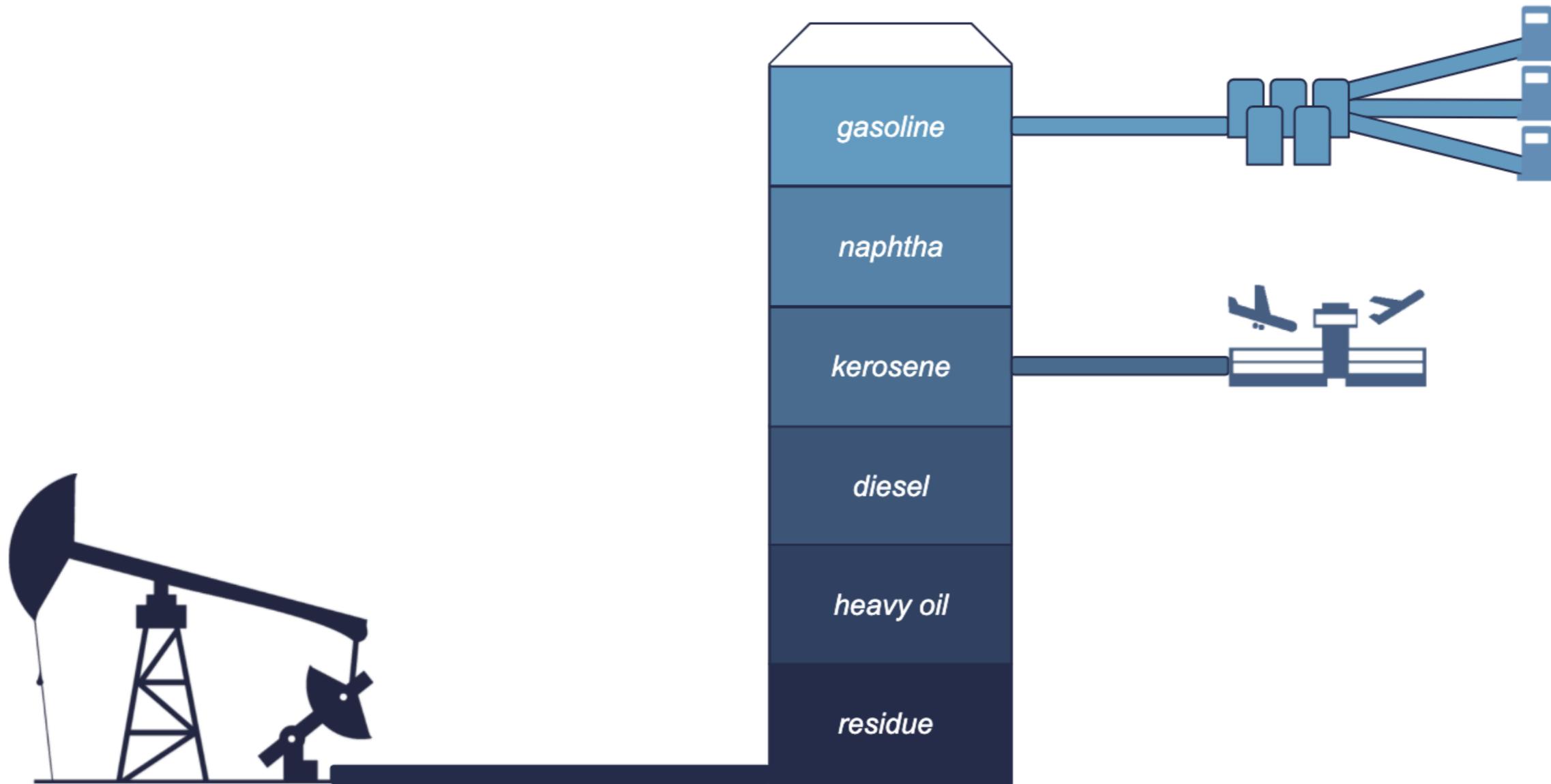


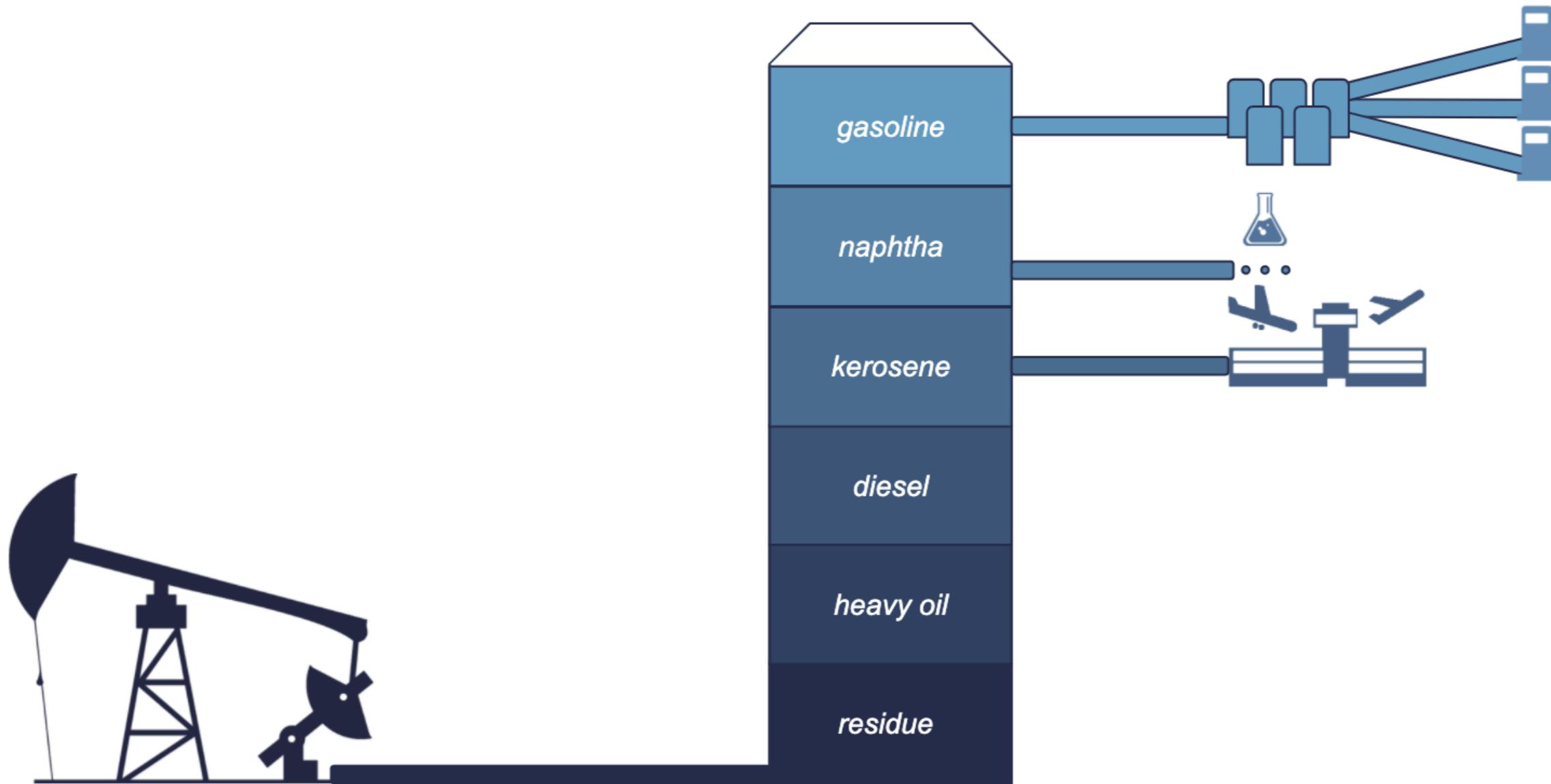


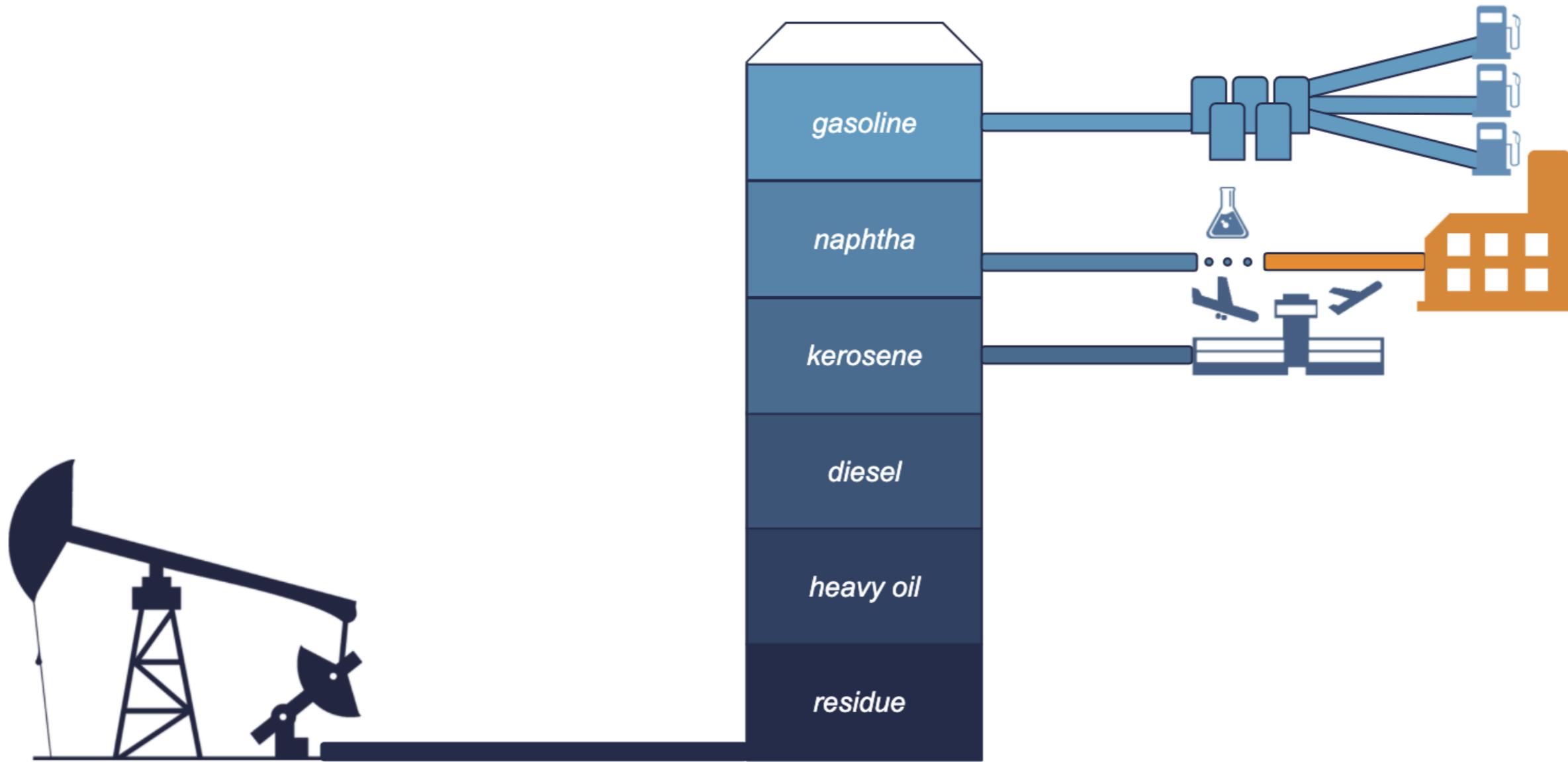












Back to data engineering

- Ingest
- Process
- Store
- Need pipelines
- Automate flow from one station to the next
- Provide up-to-date, accurate, relevant data



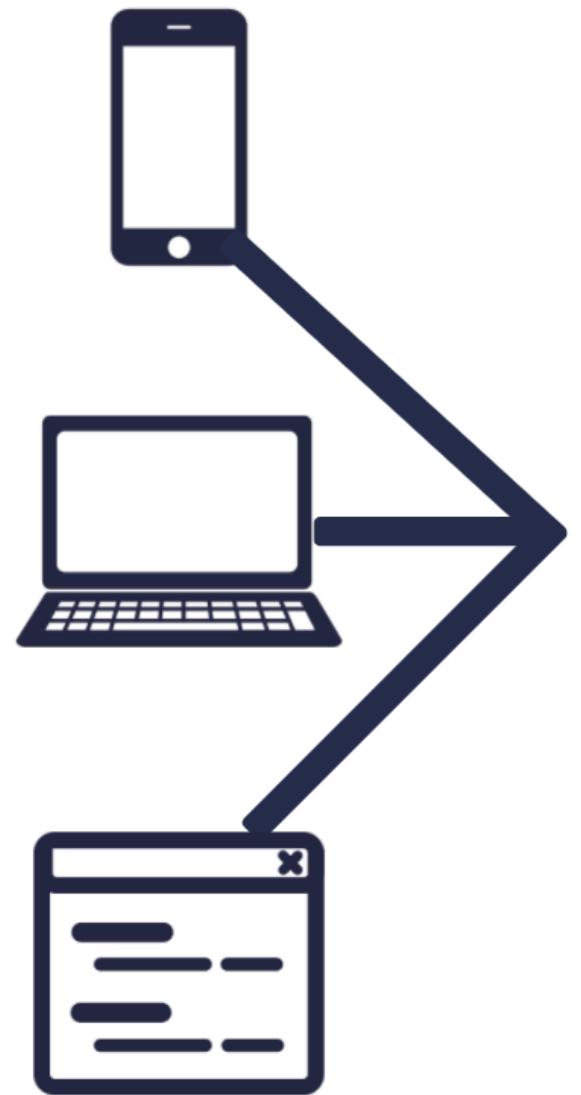


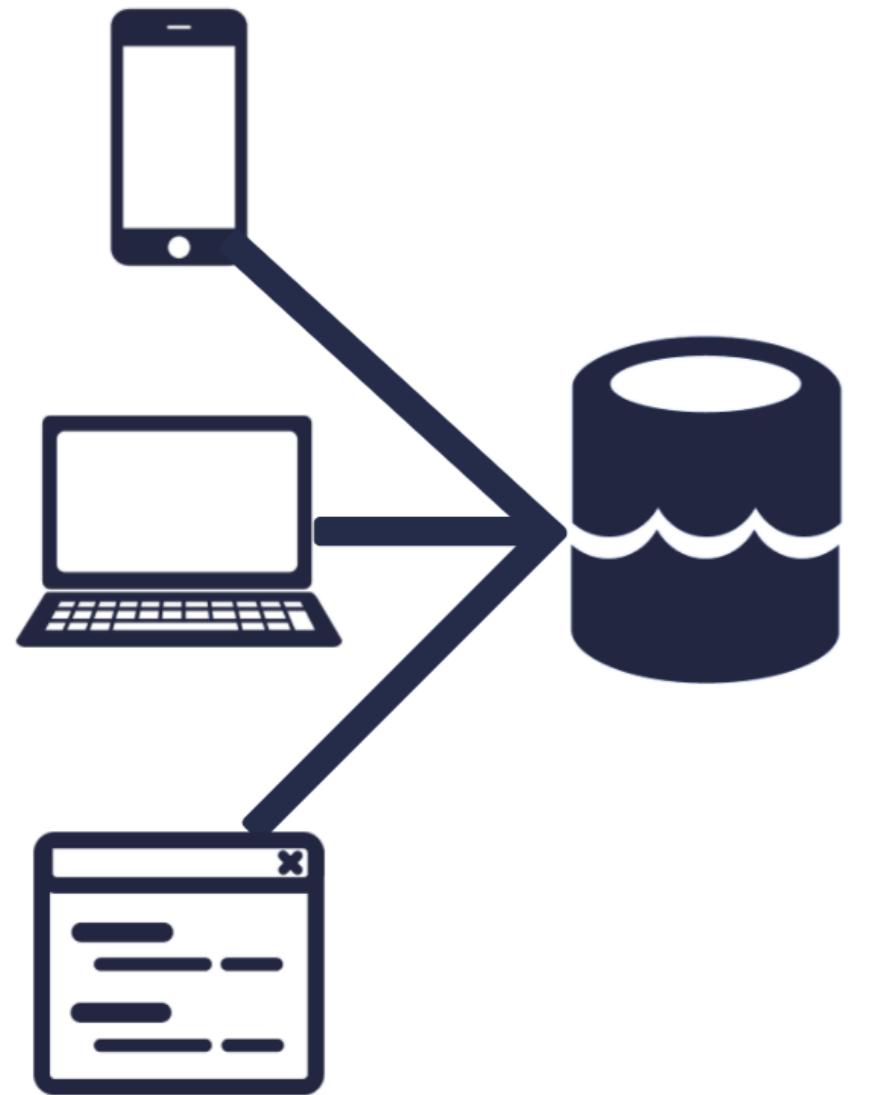


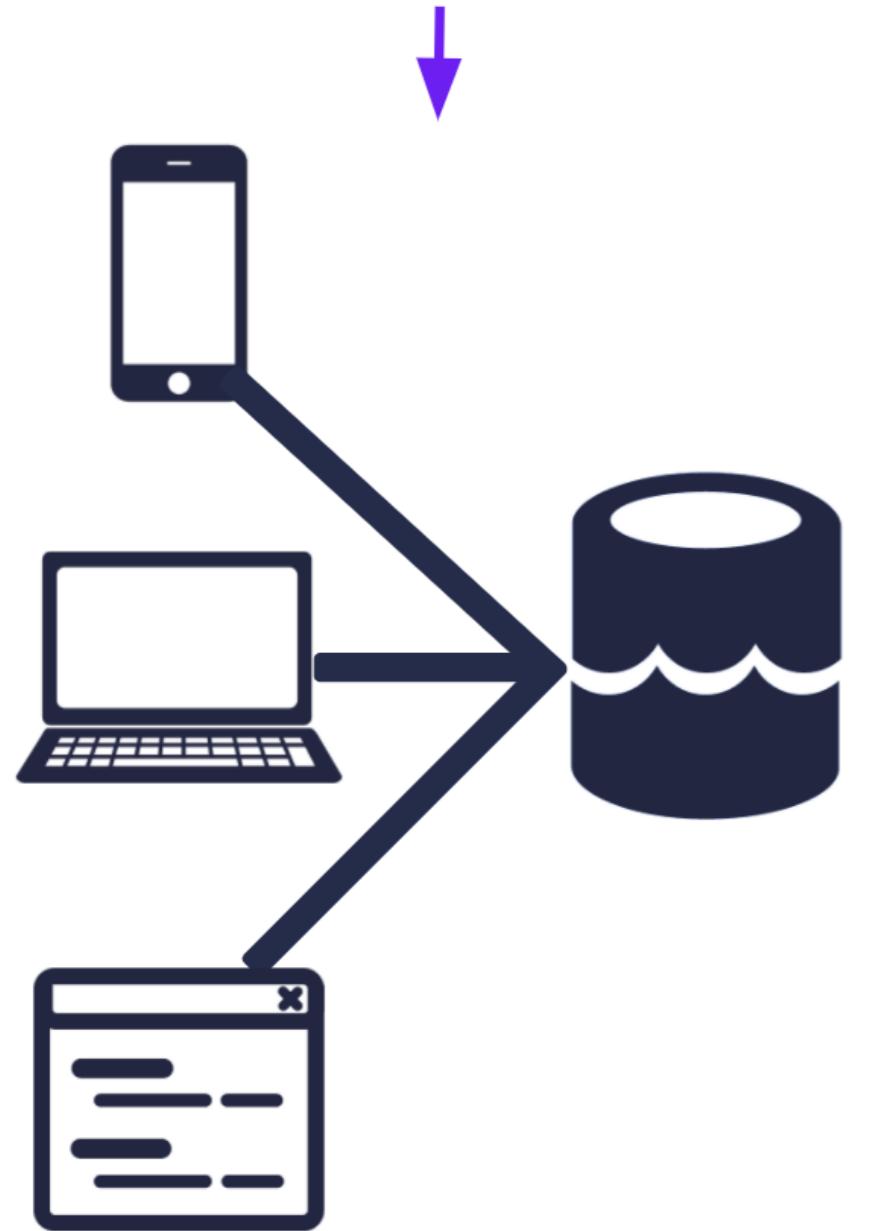


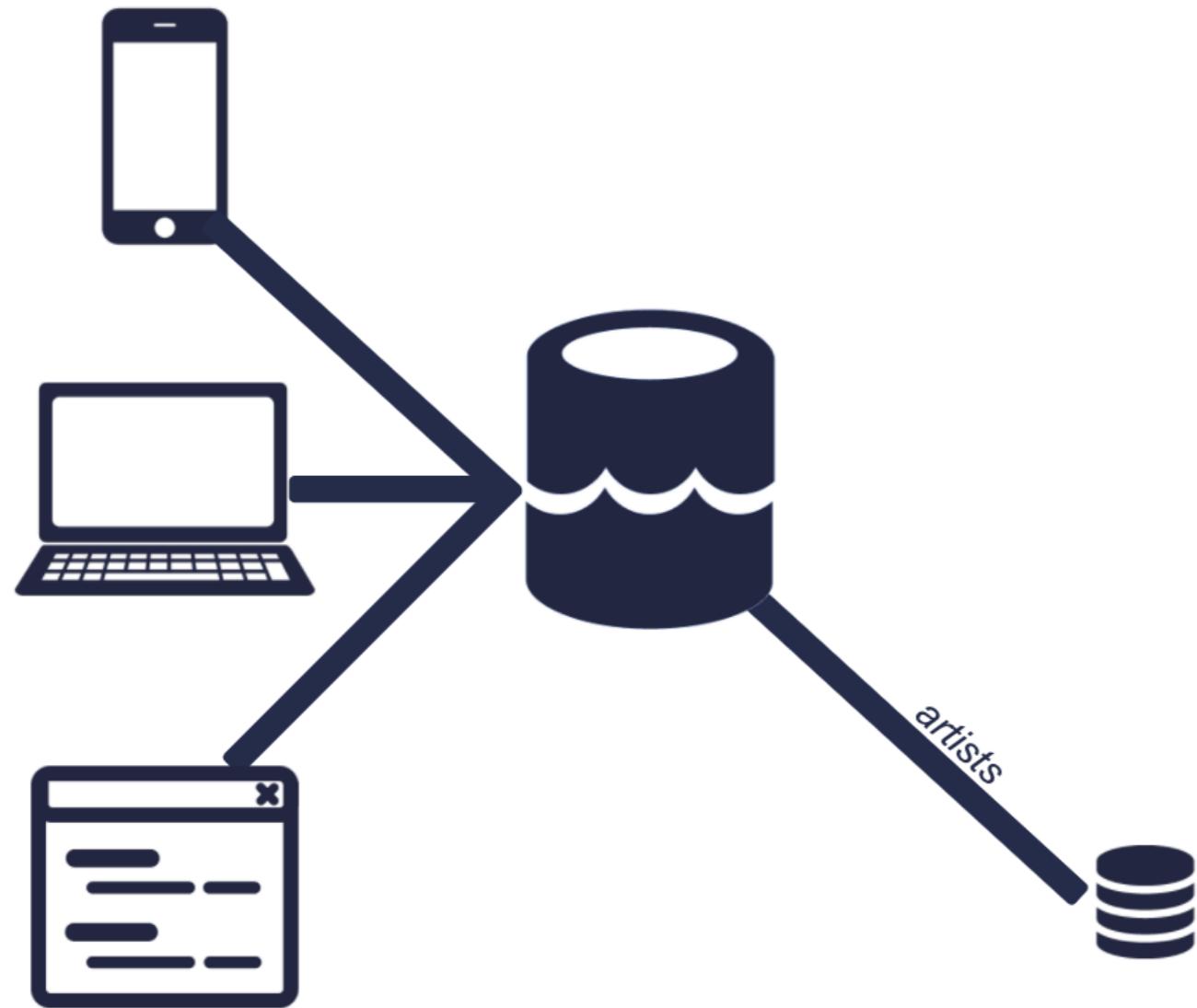


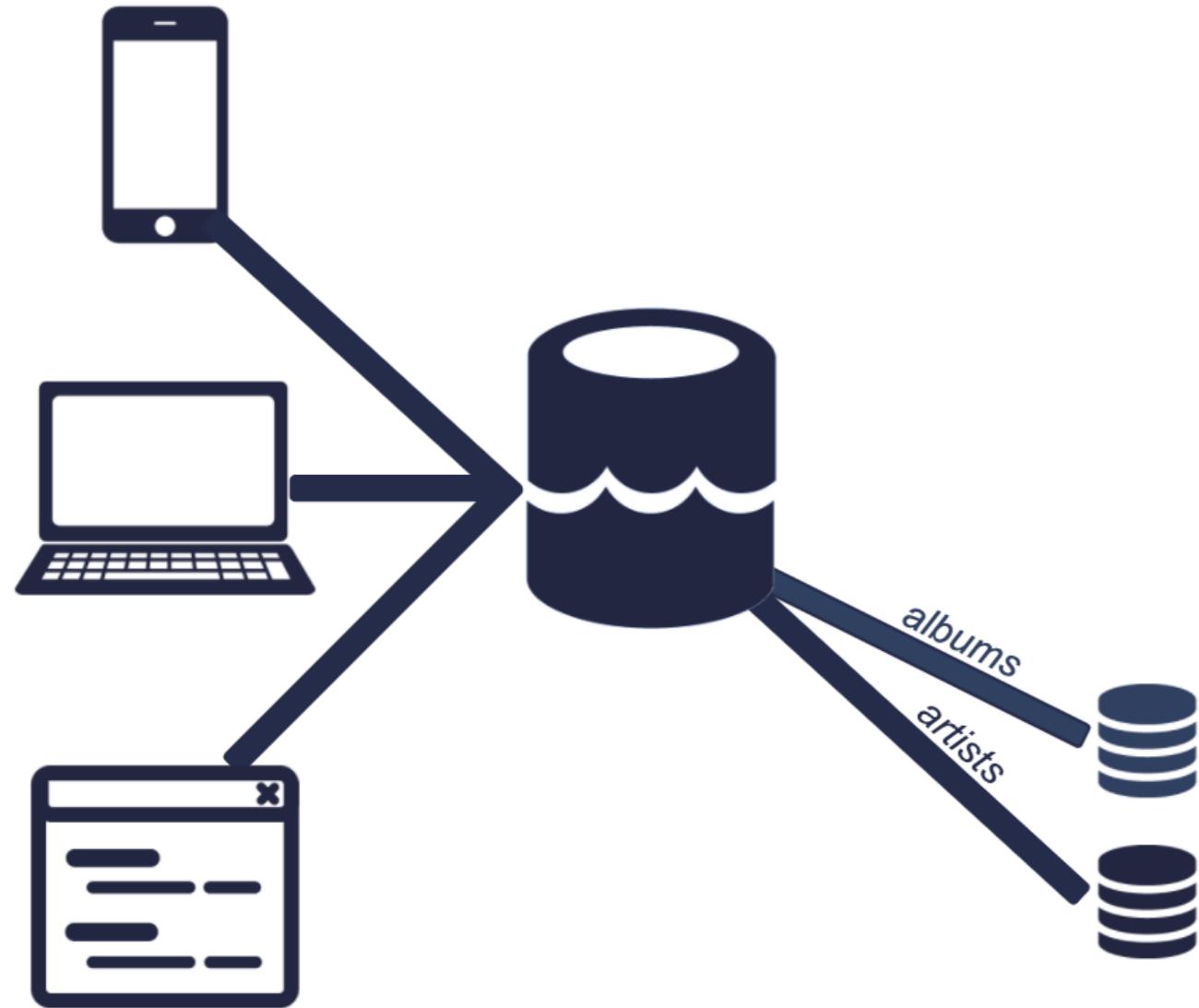


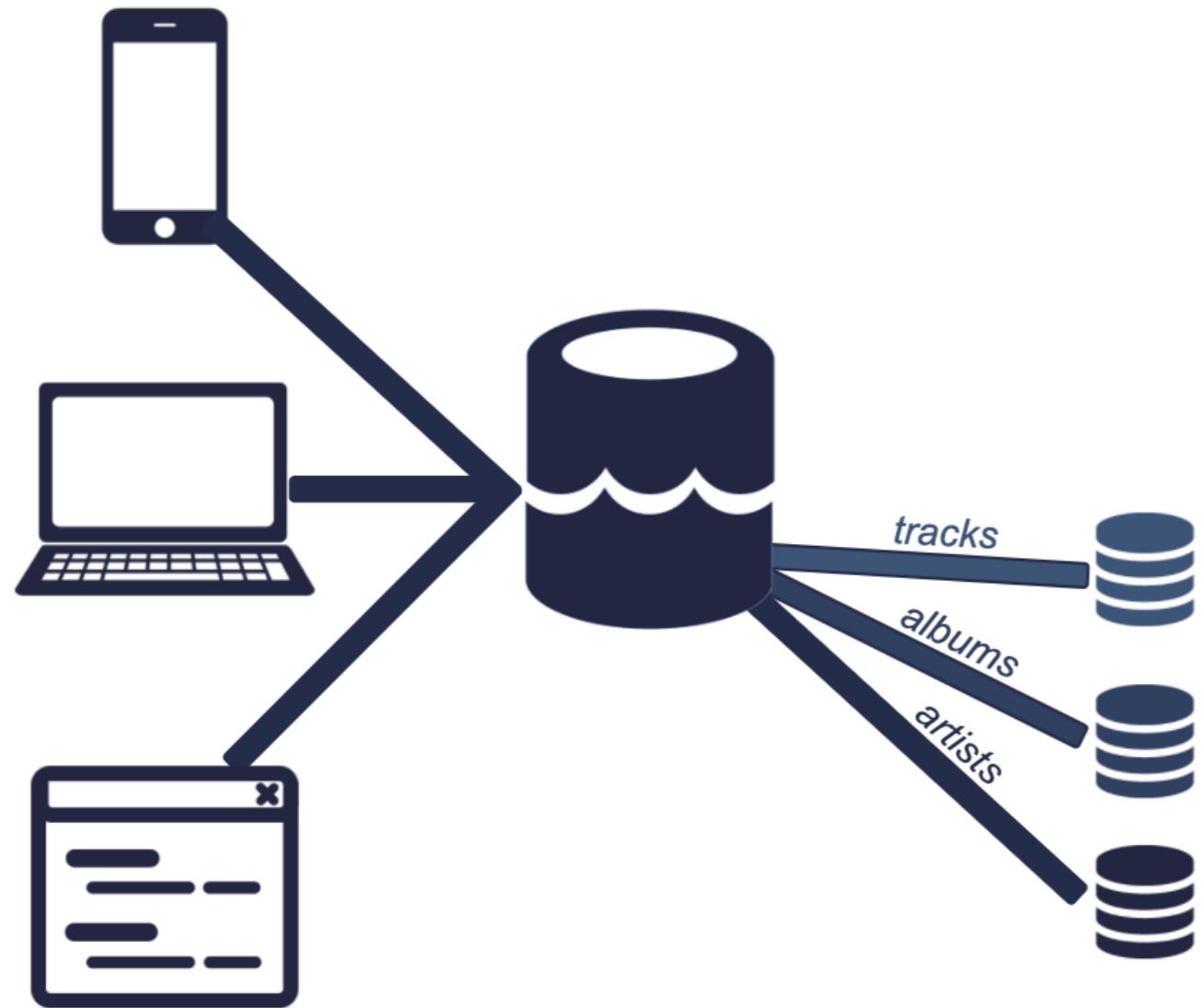


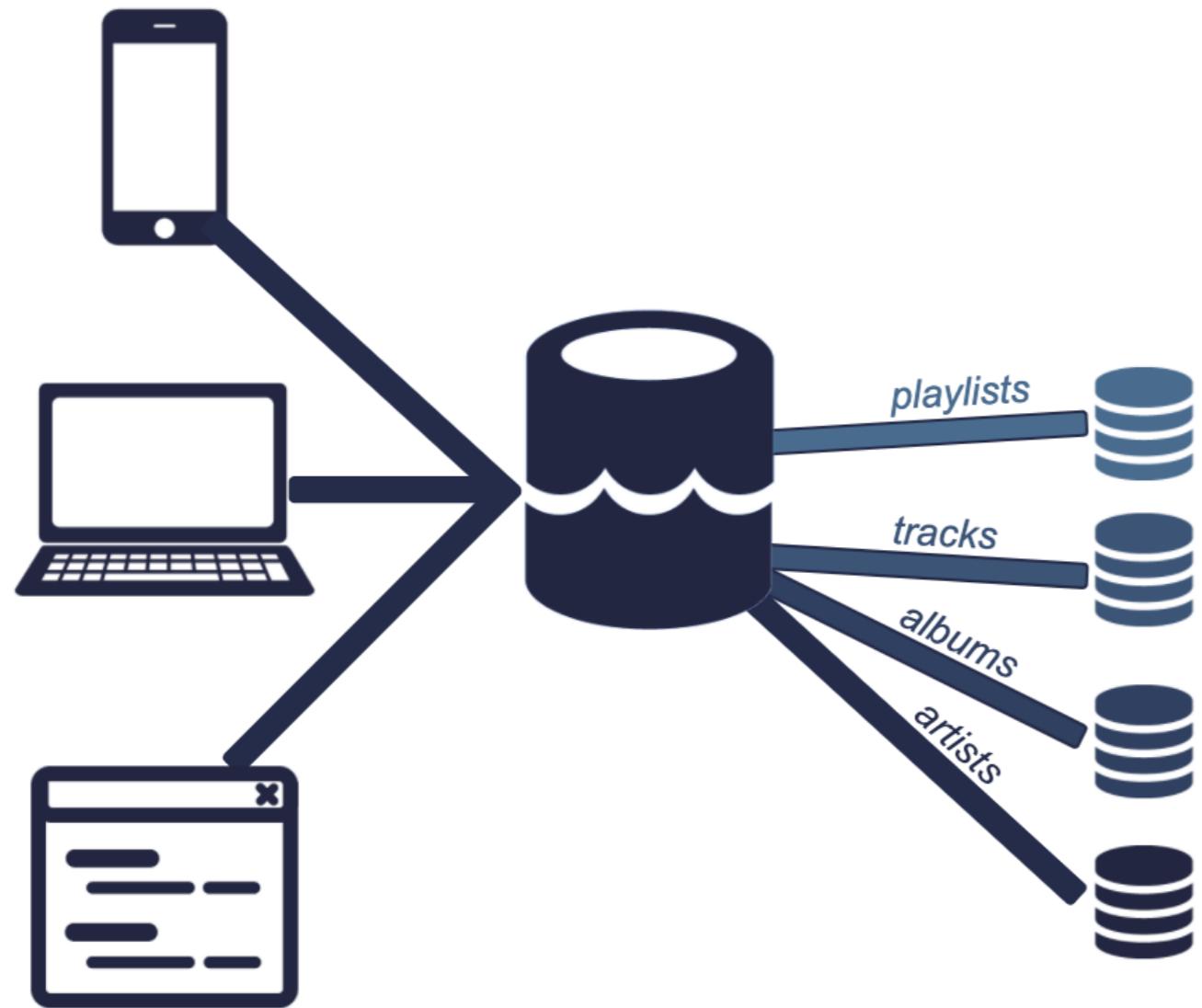


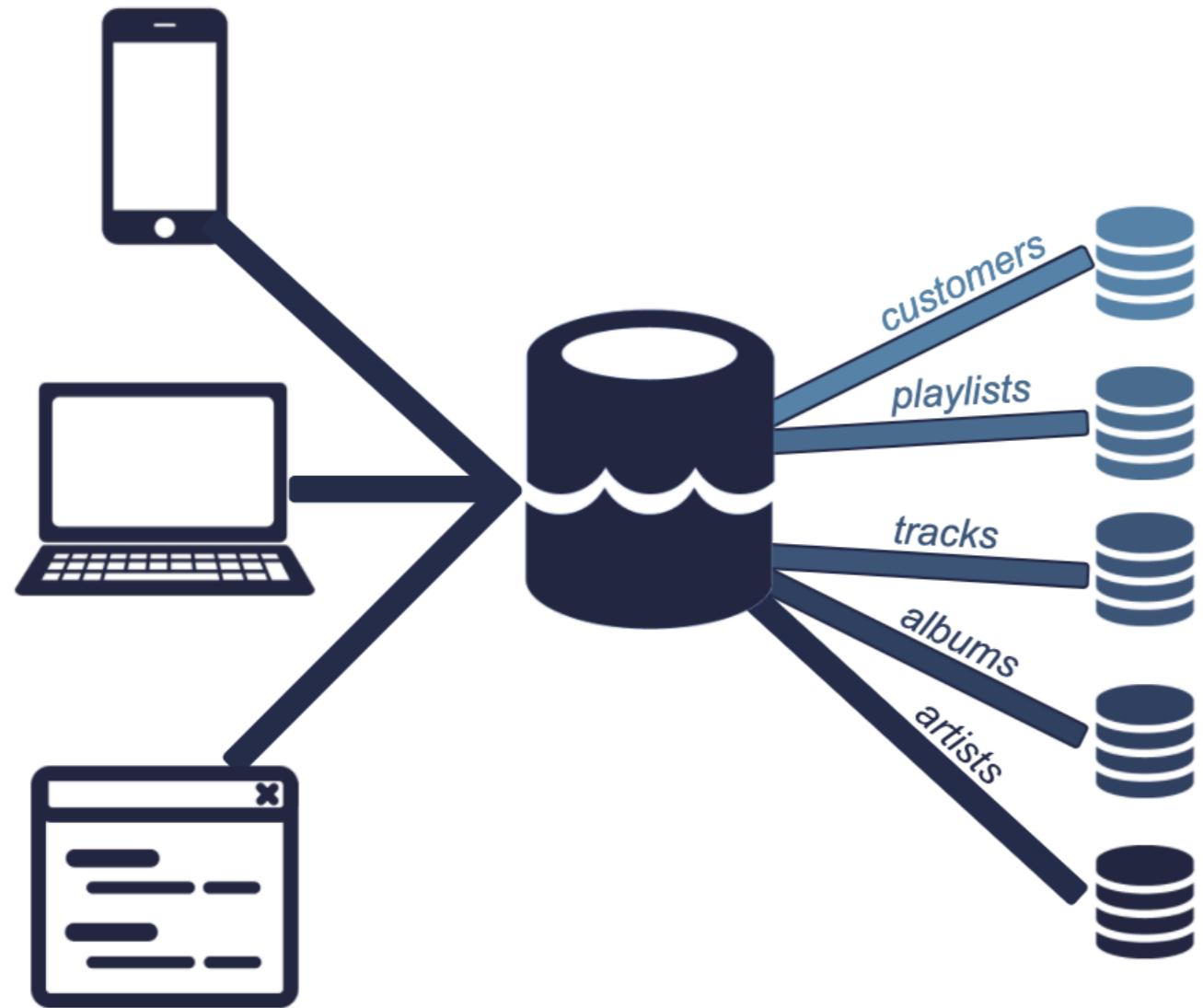


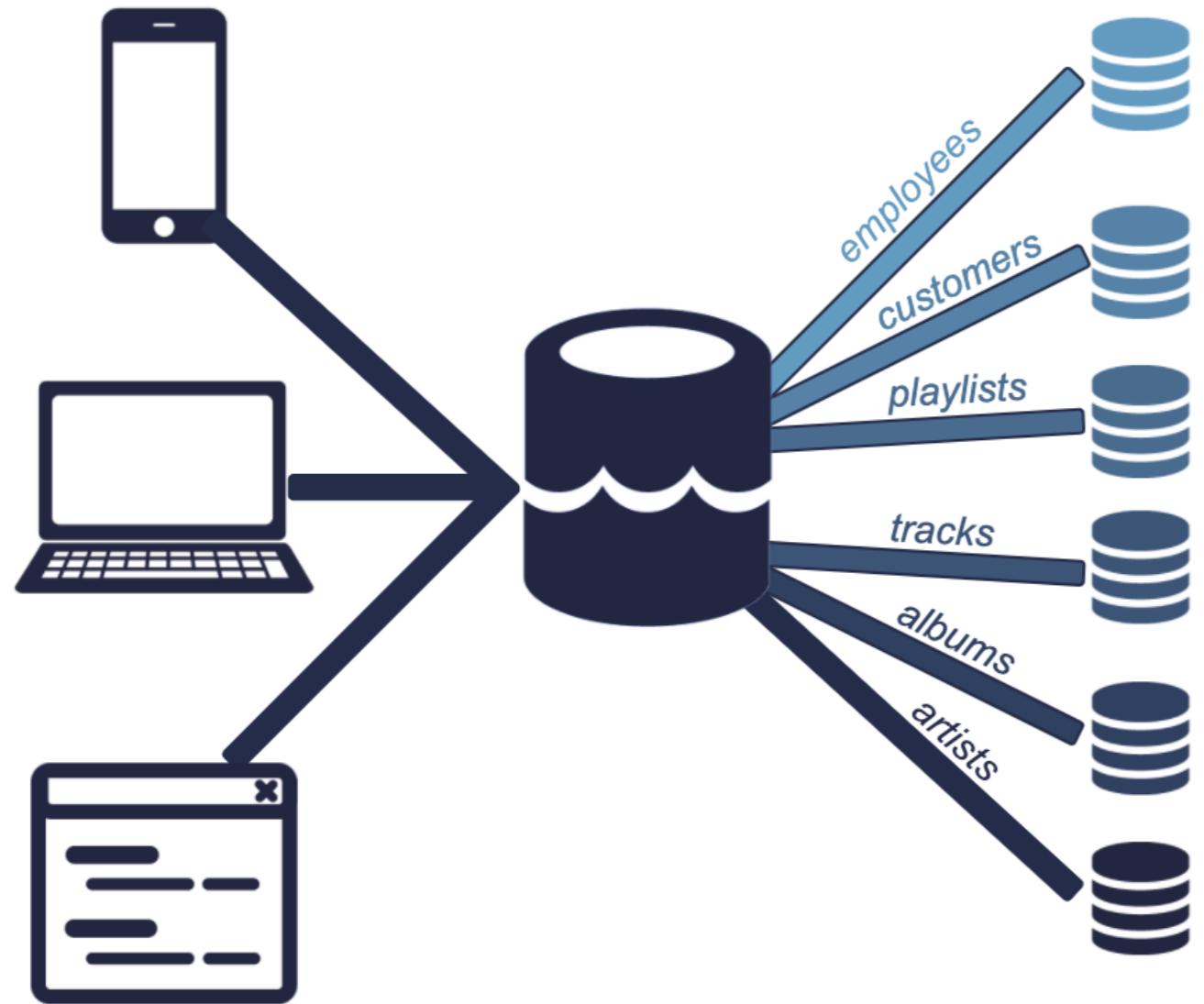


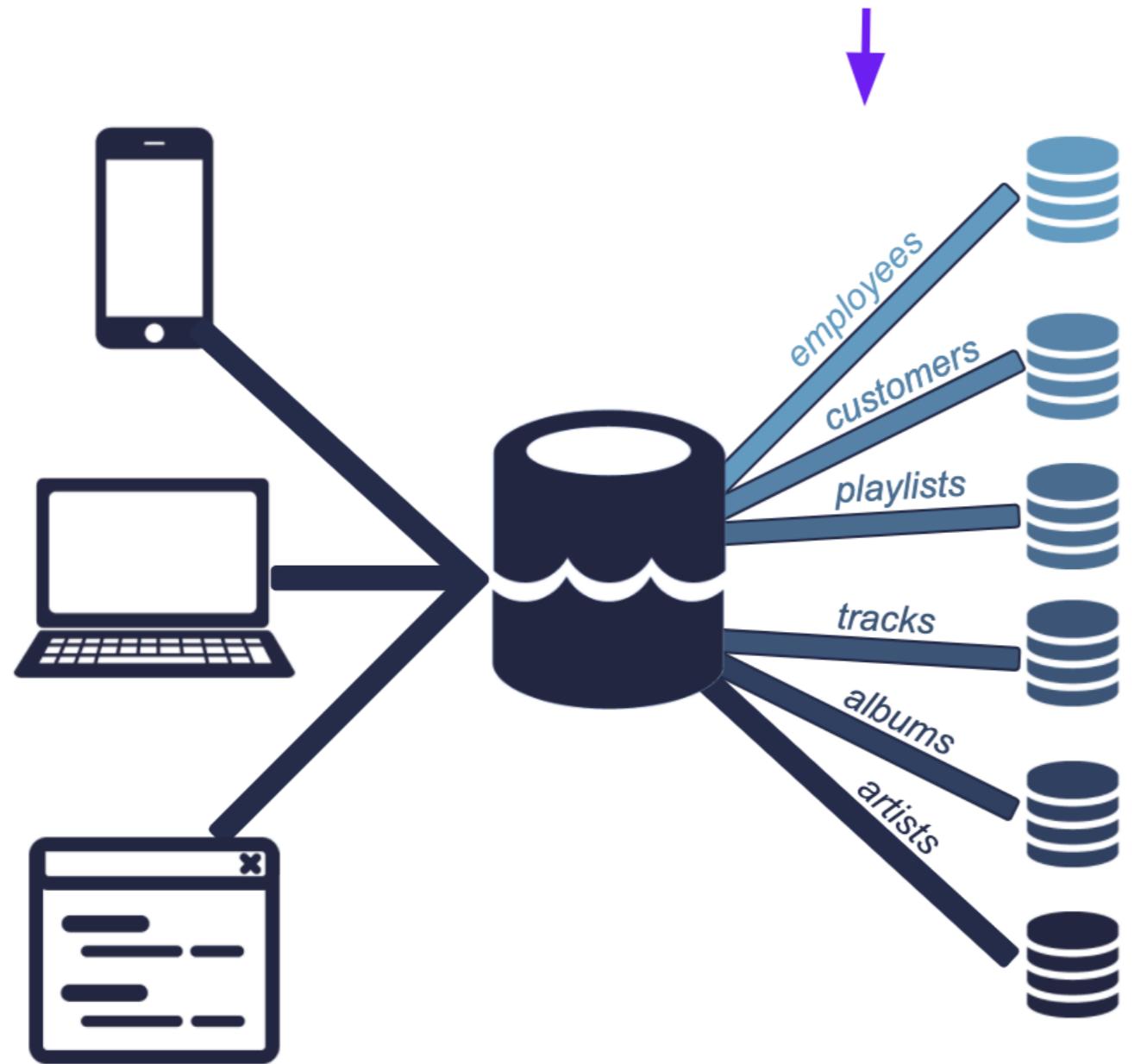


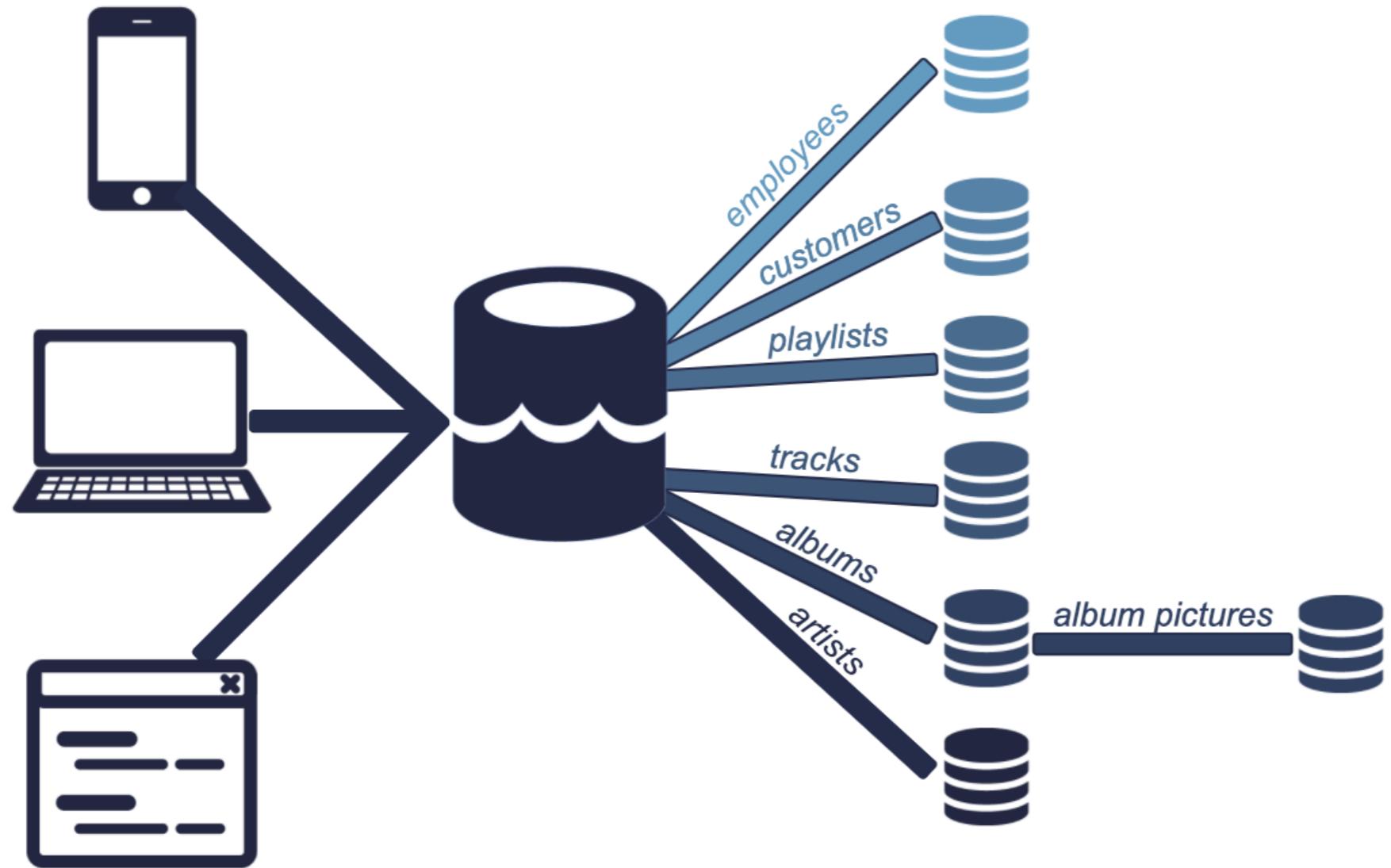


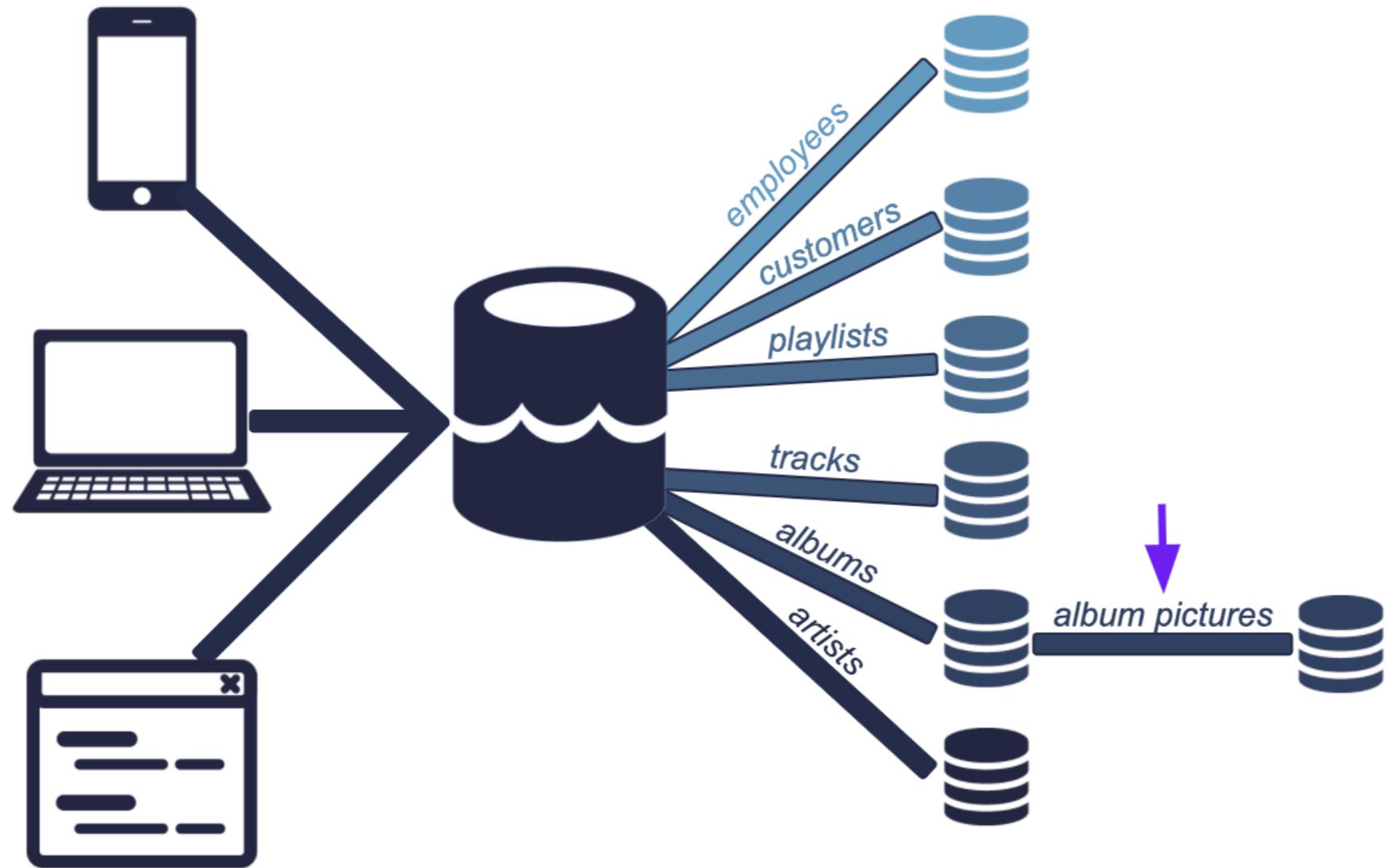


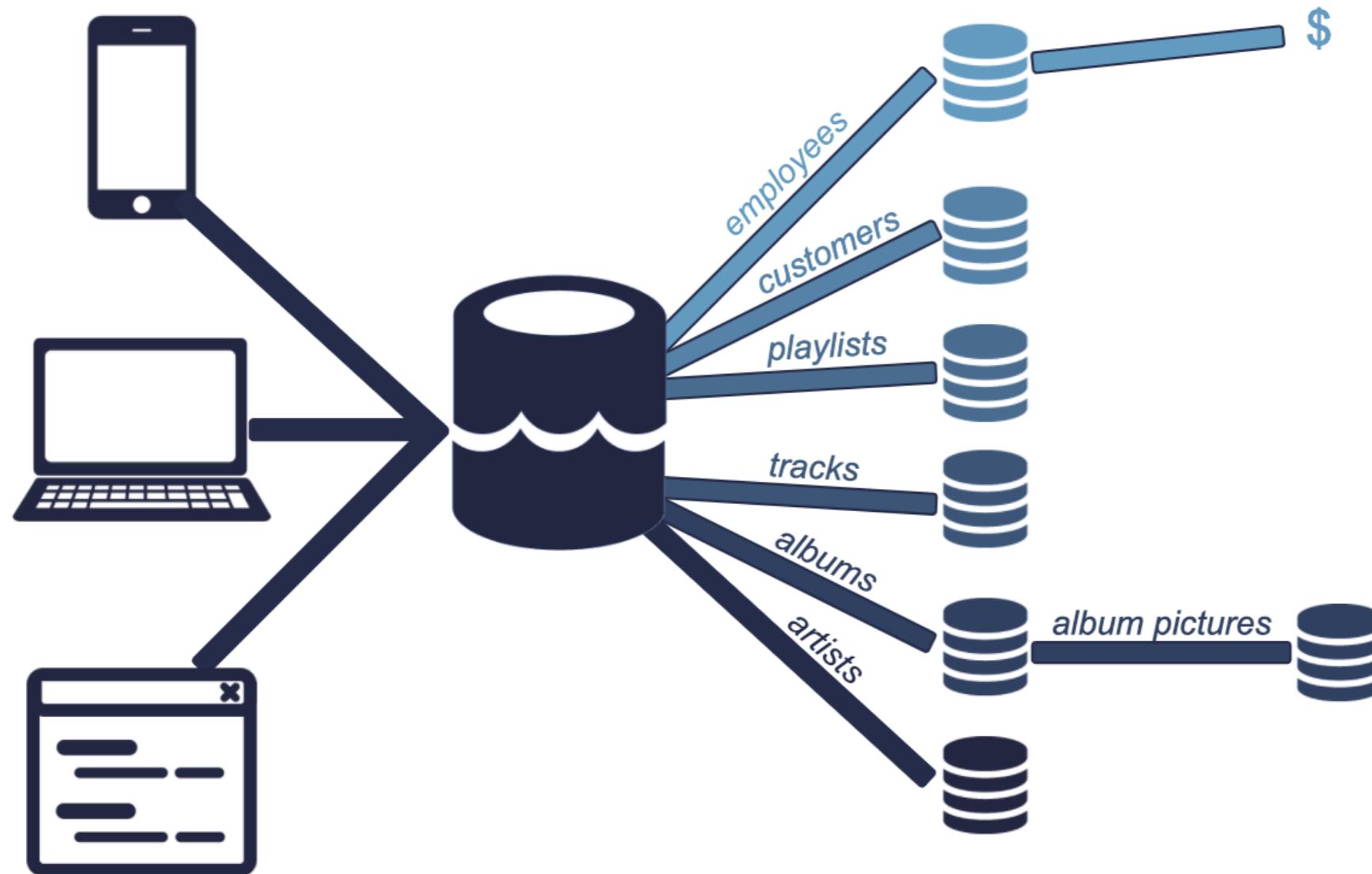


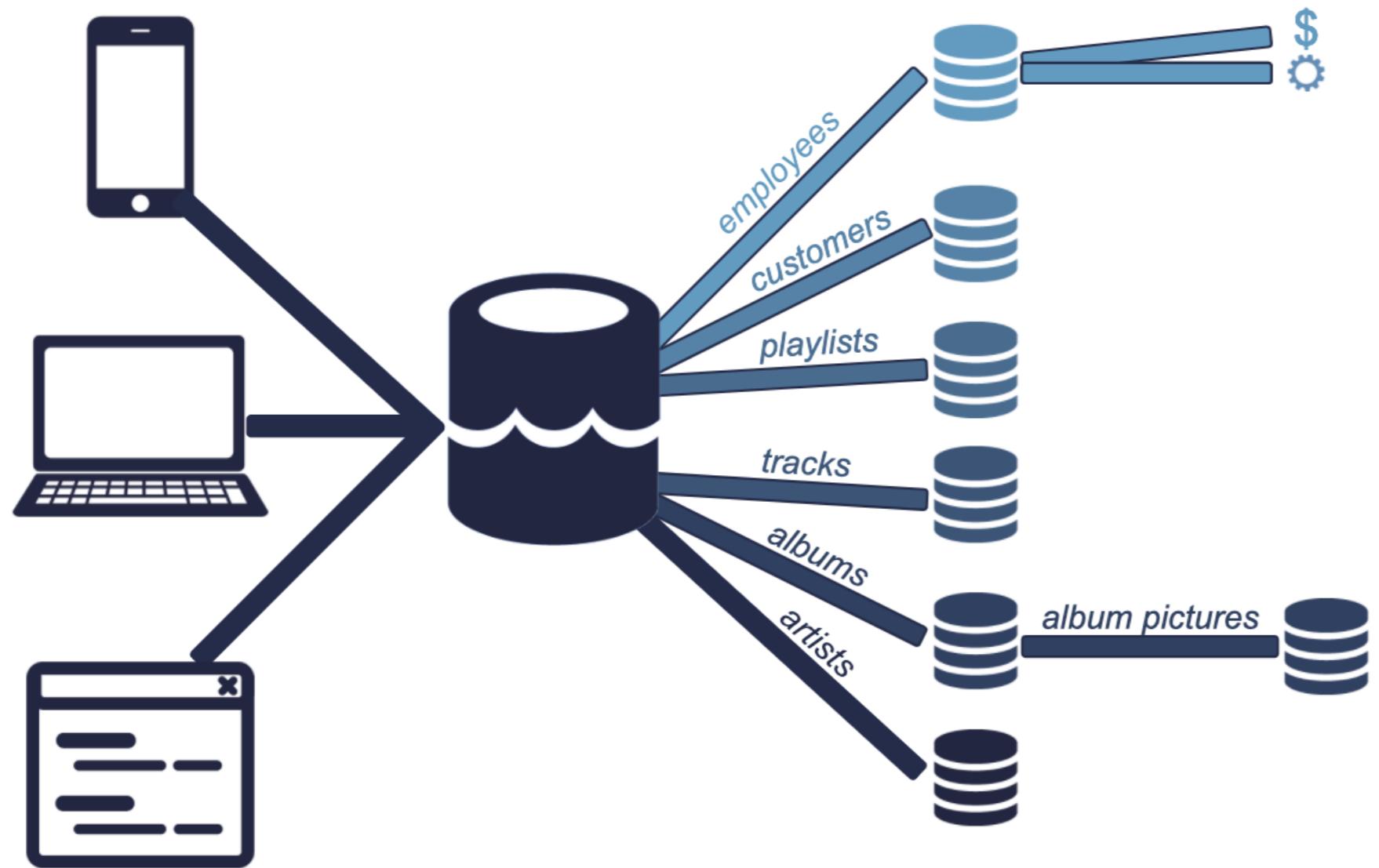


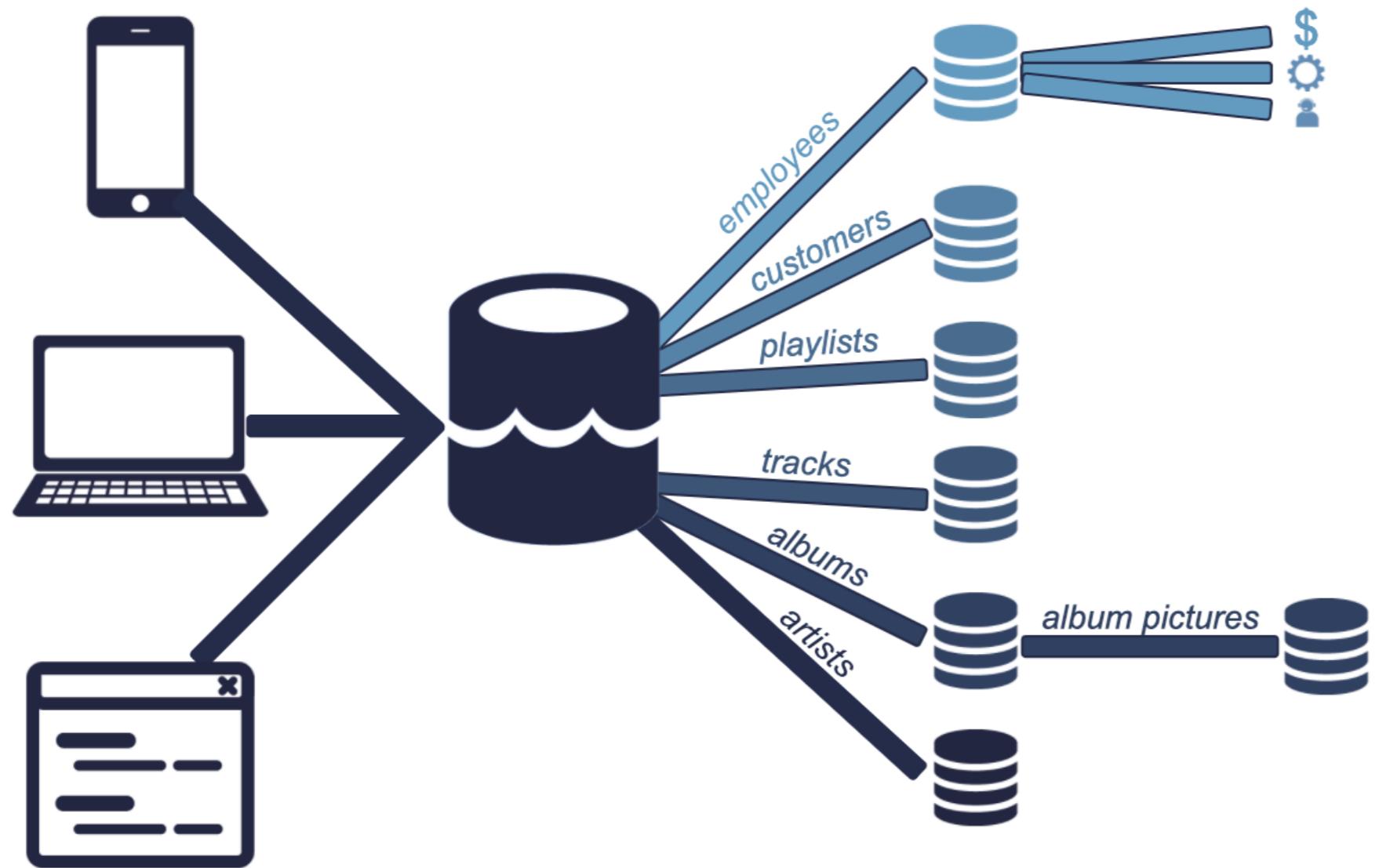


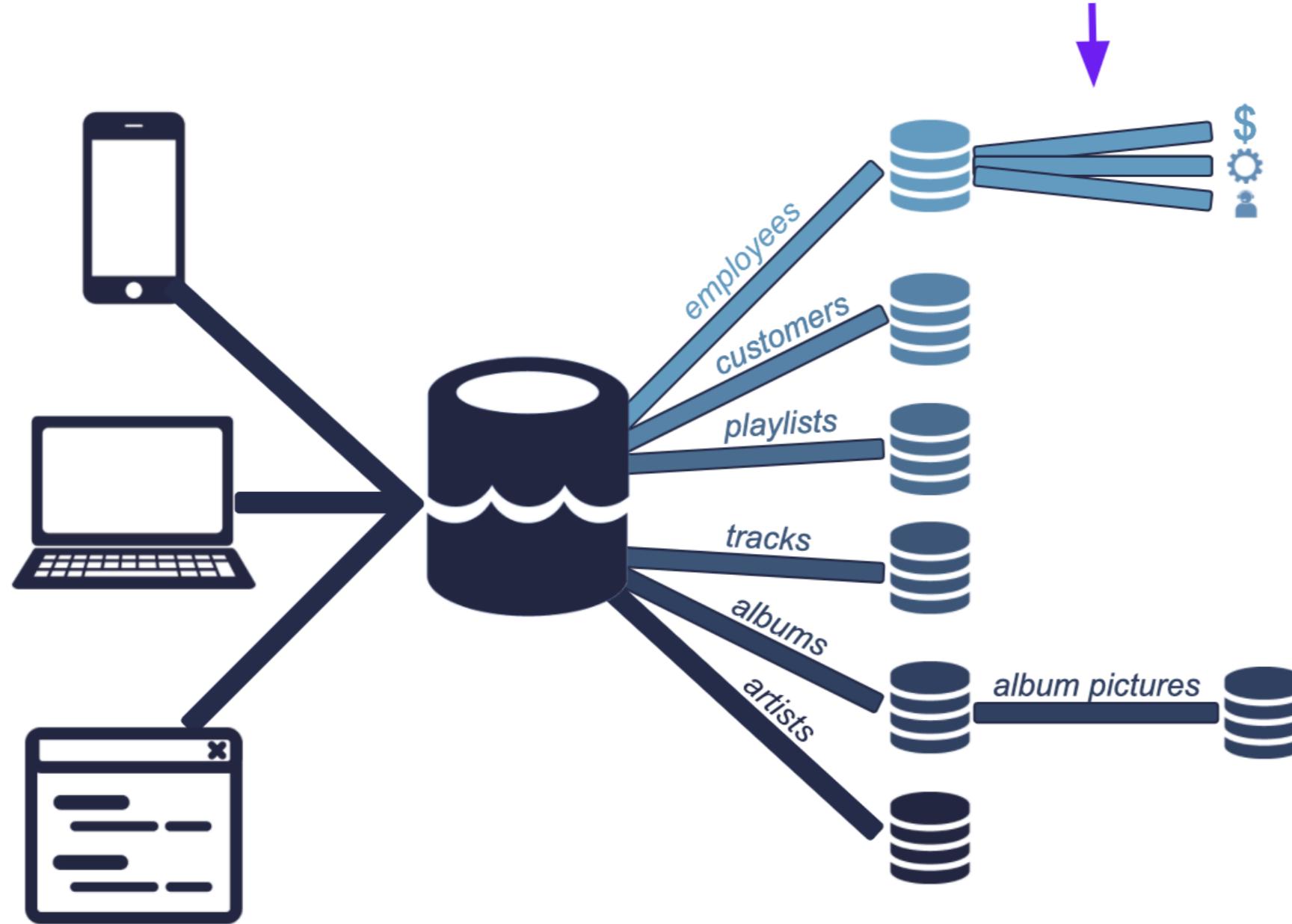


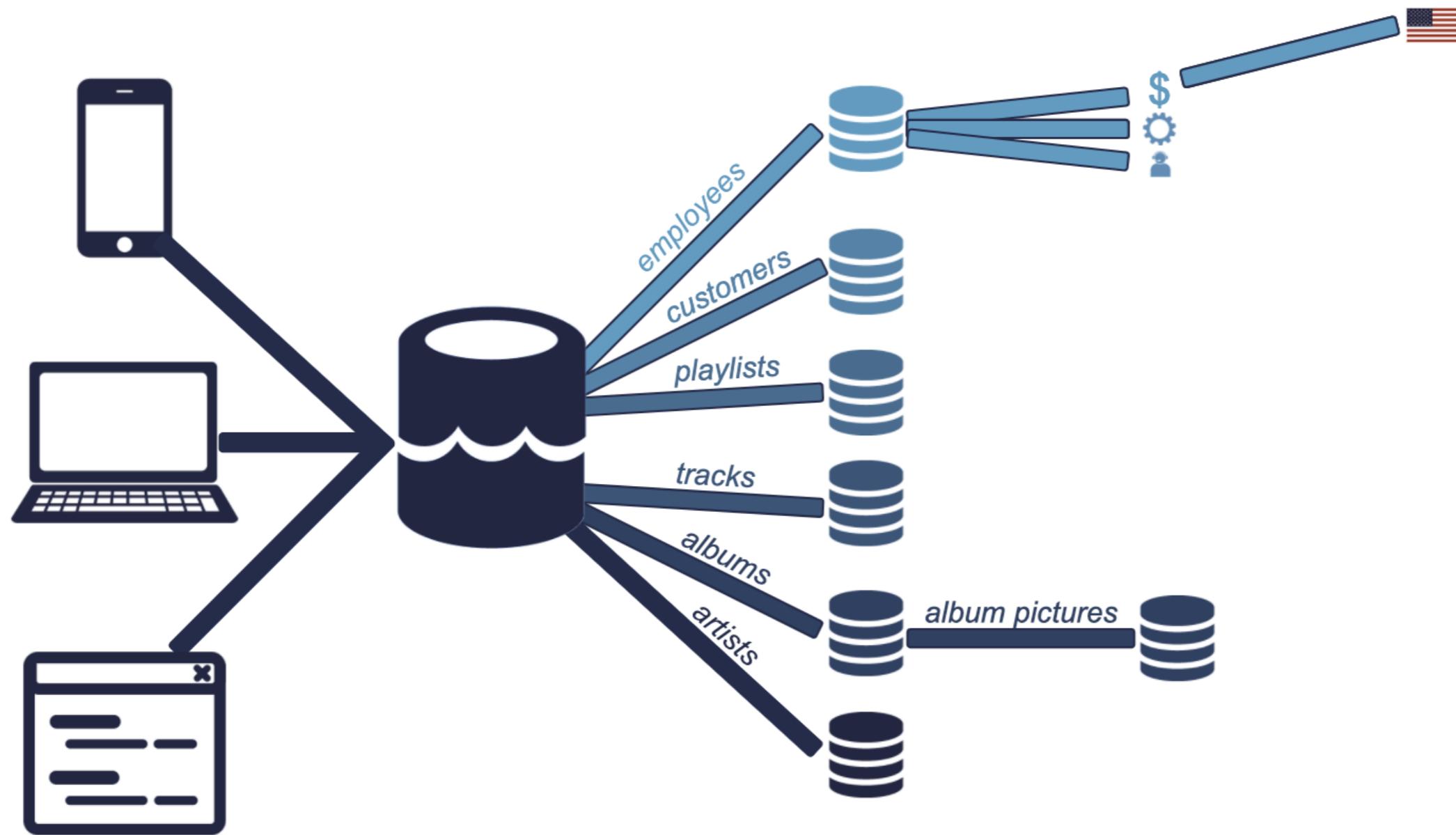


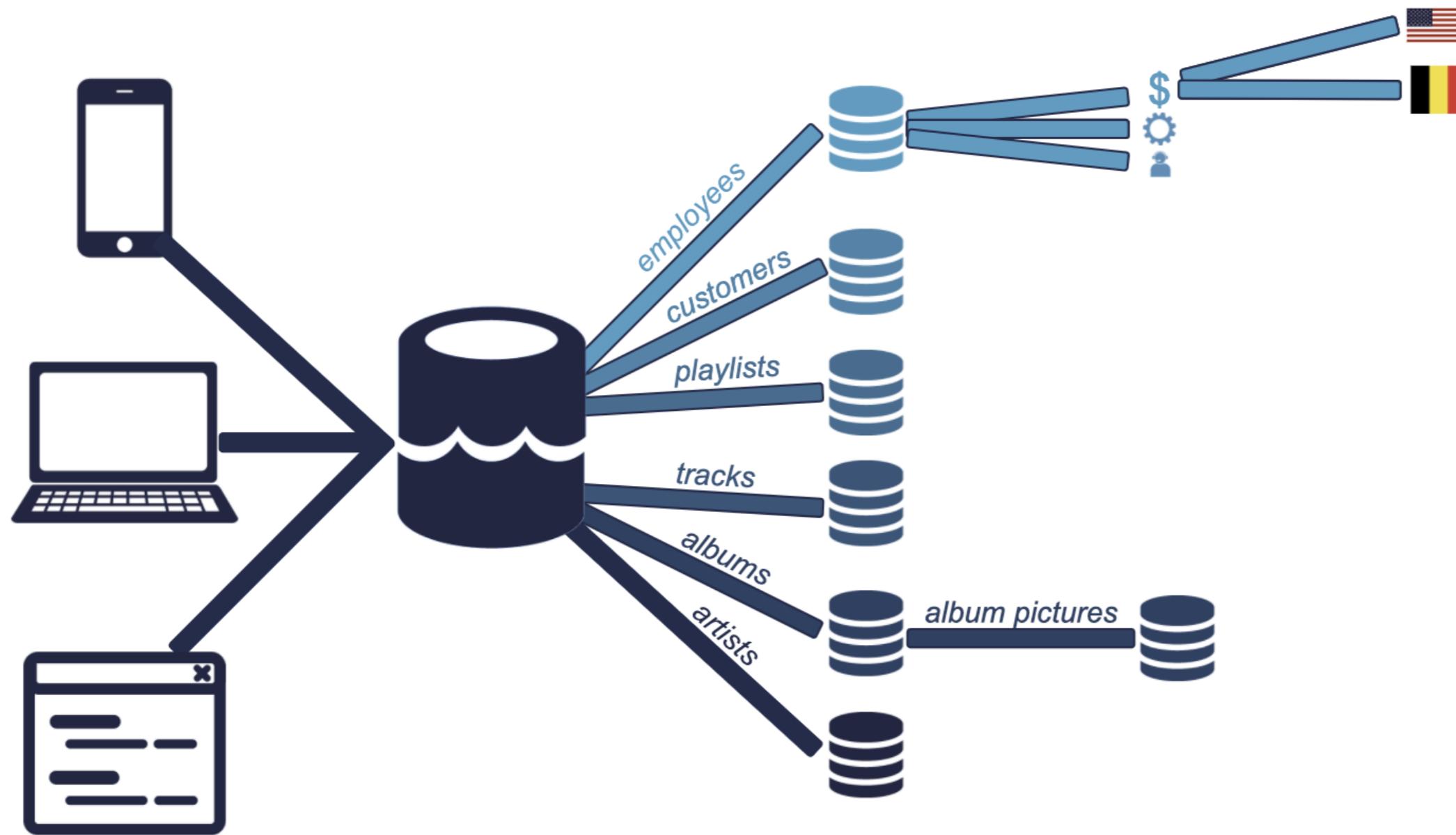


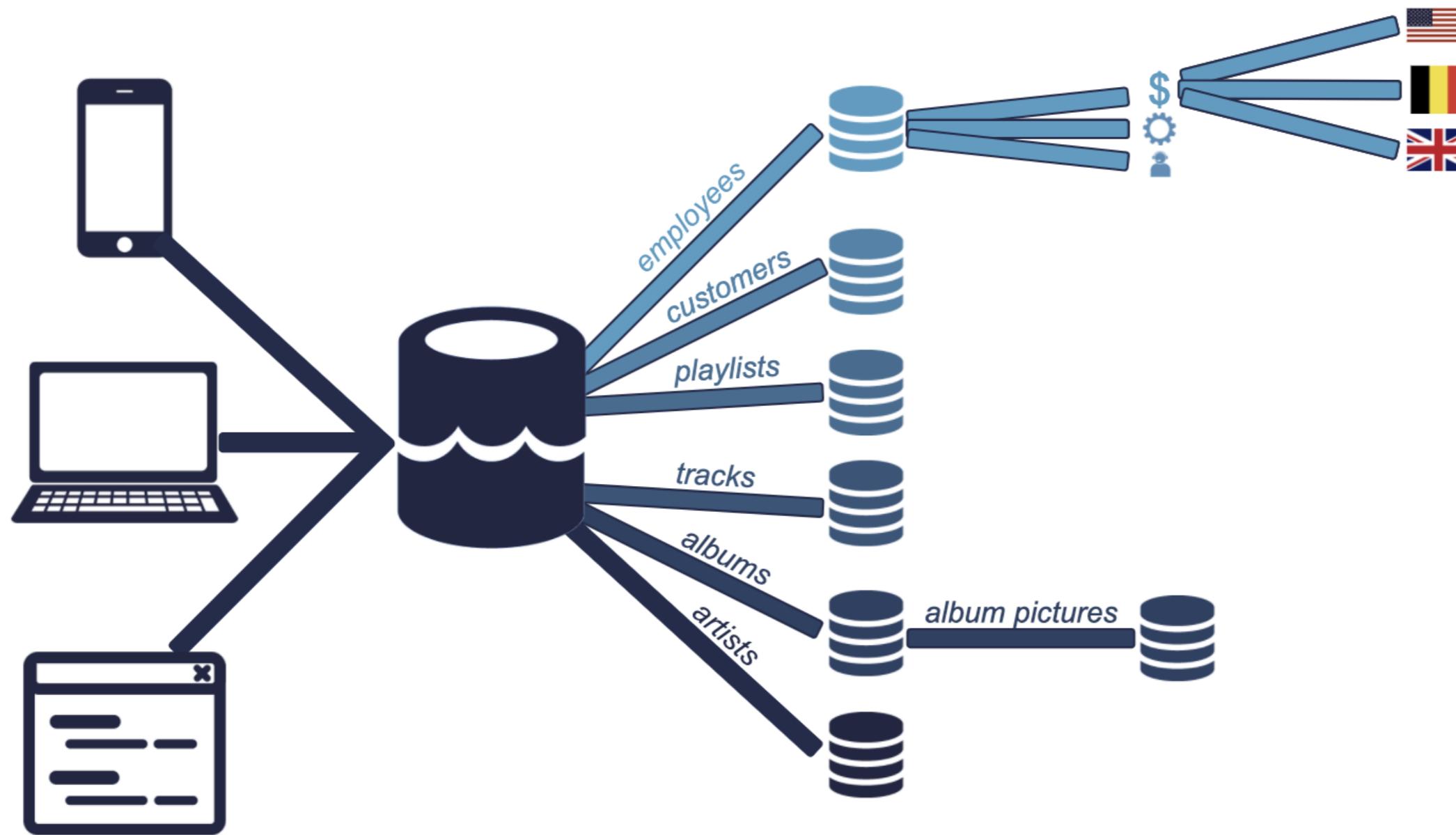


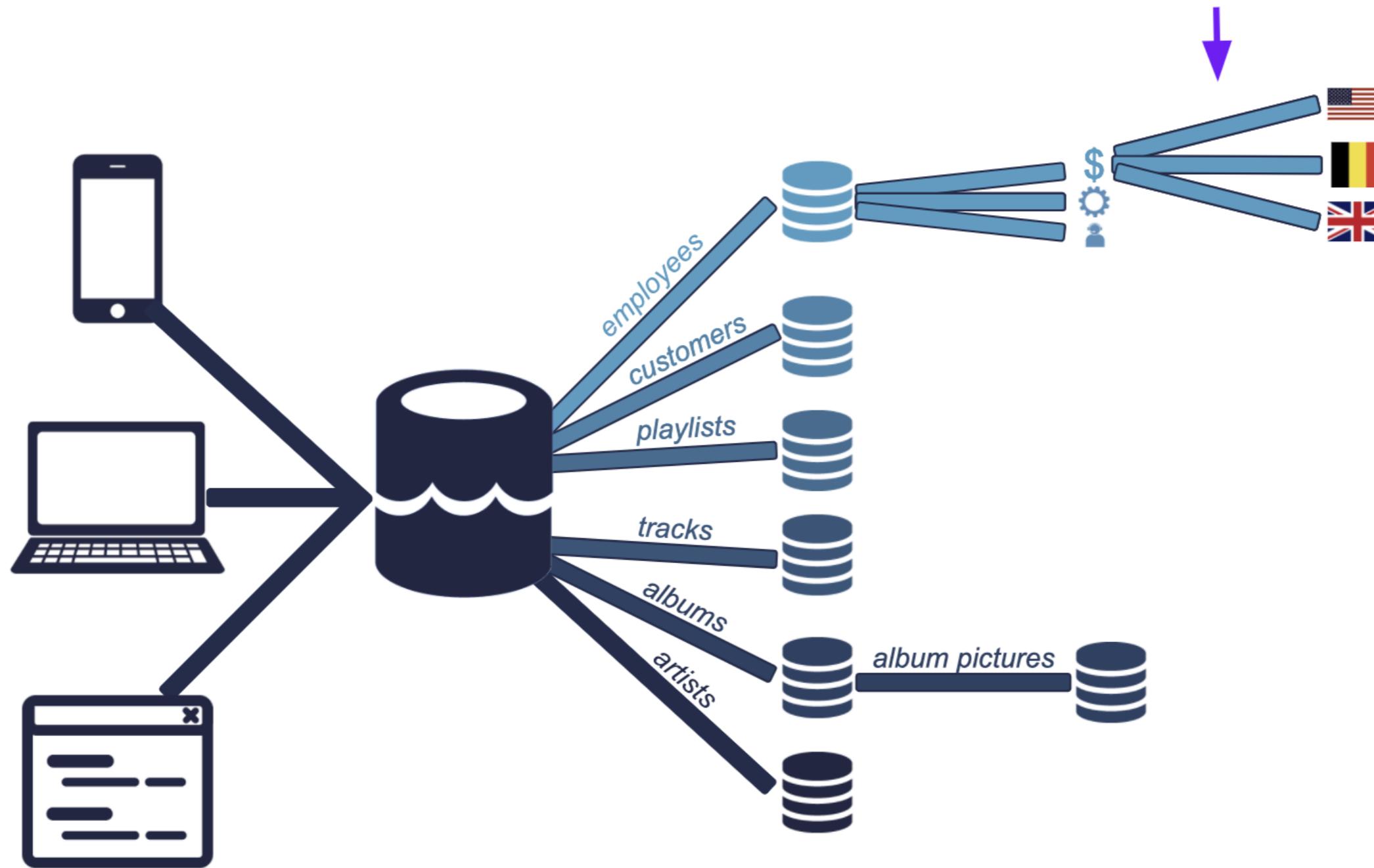


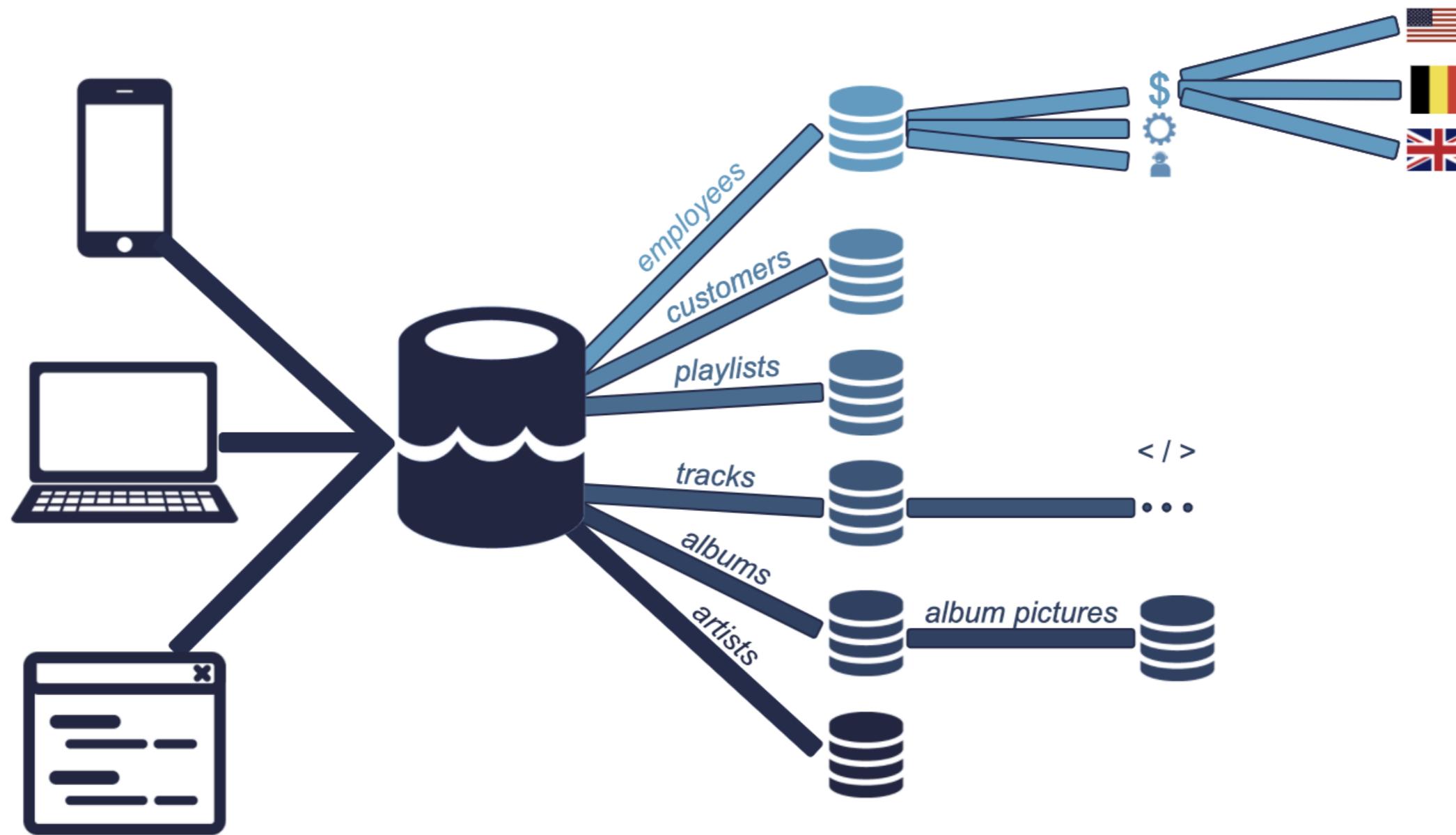


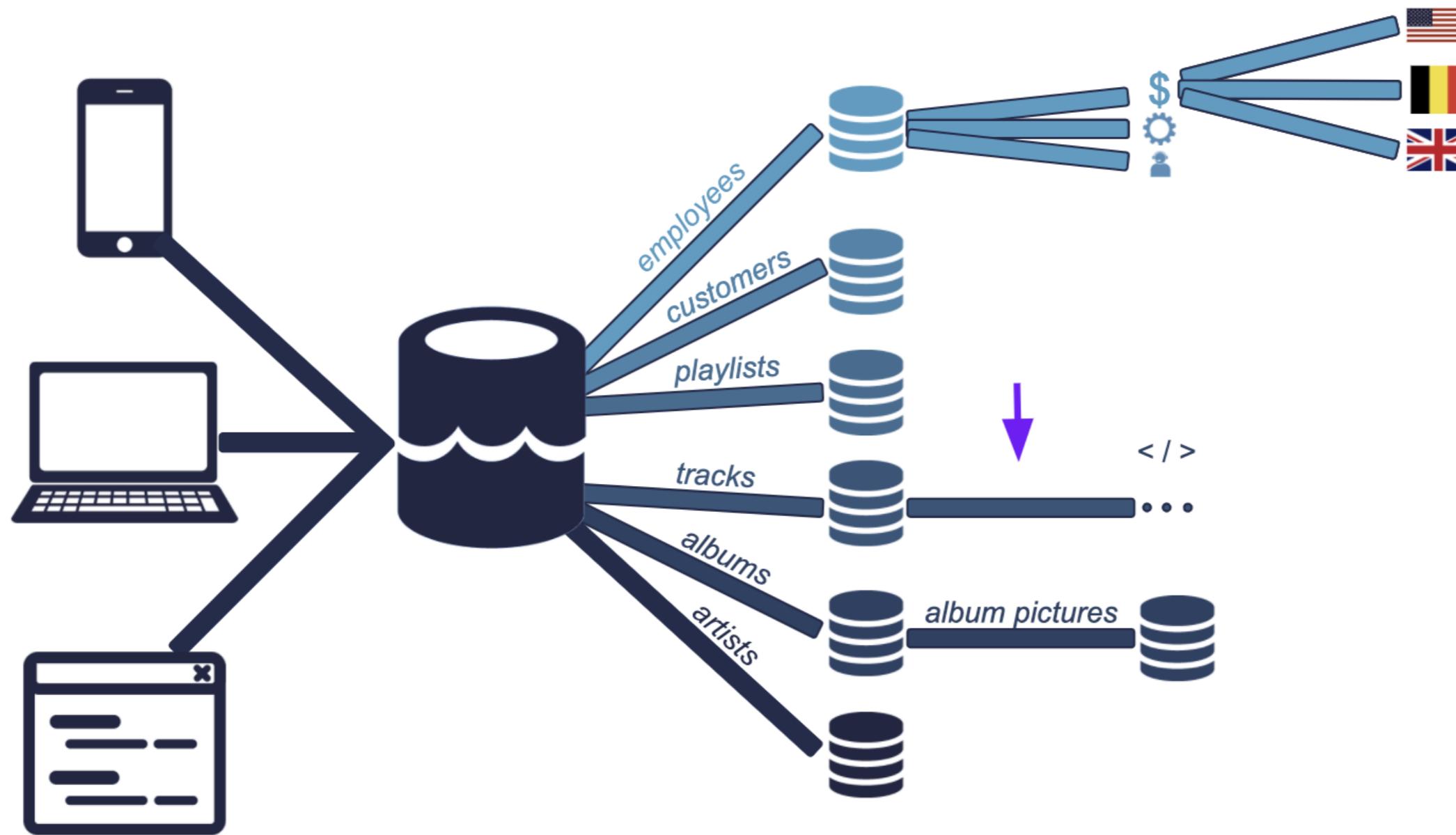


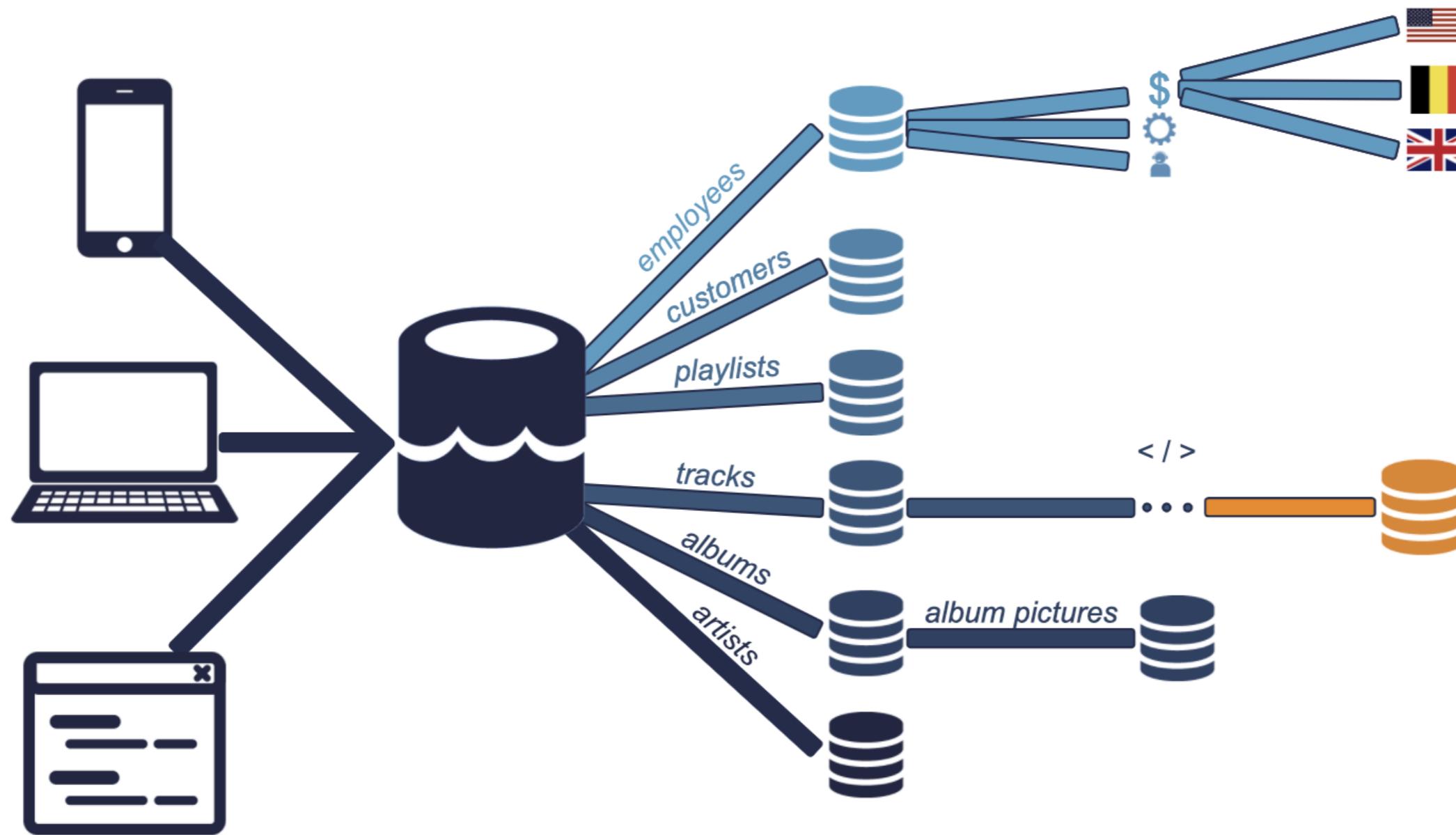


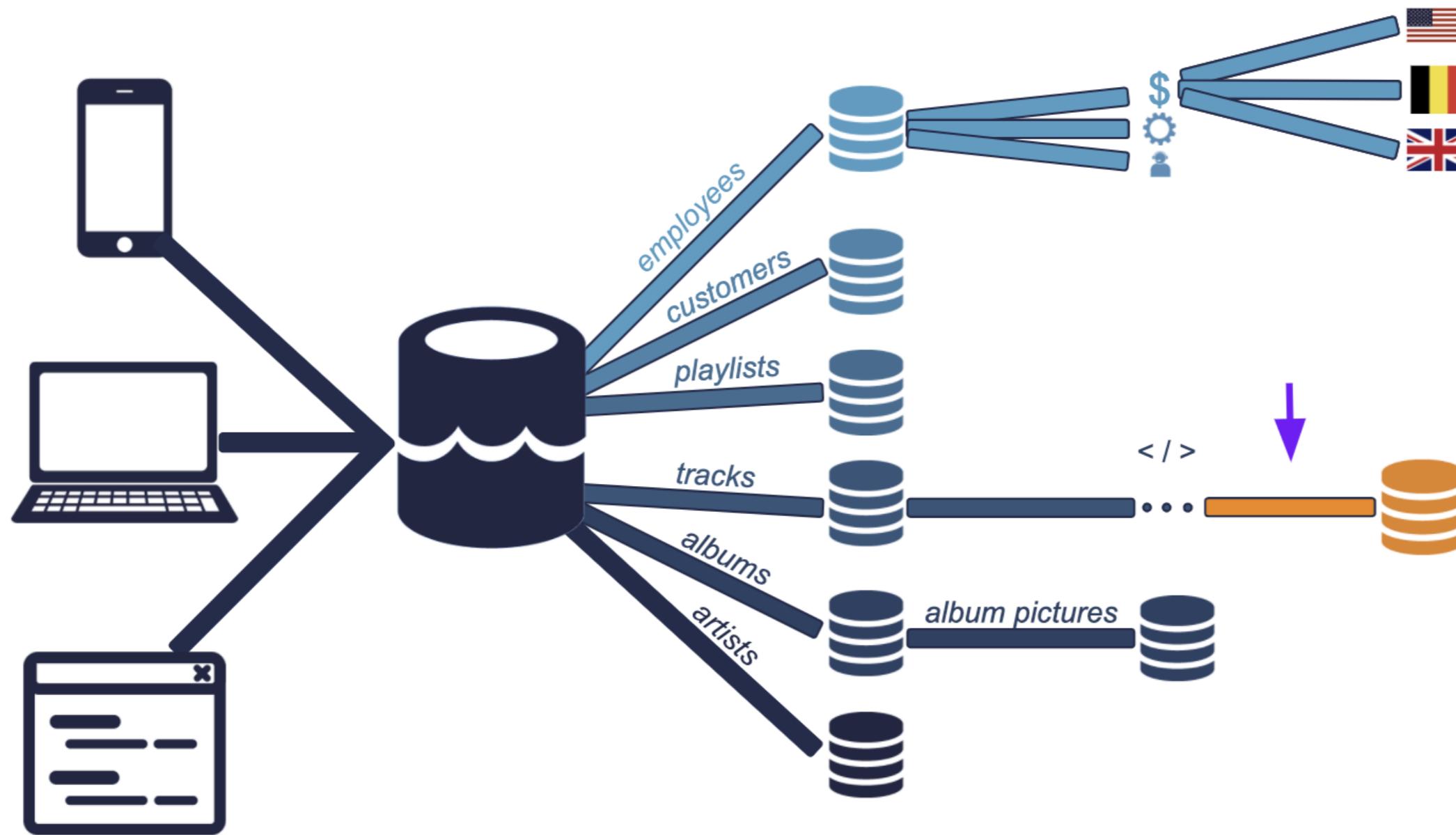














2004

You get a pipeline !



UNDERSTANDING DATA ENGINEERING

2004

Everybody gets a pipeline !!!



Data pipelines ensure an efficient flow of the data

Automate

- Extracting
- Transforming
- Combining
- Validating
- Loading

Reduce

- Human intervention
- Errors
- Time it takes data to flow

ETL and data pipelines

ETL

- Popular framework for designing data pipelines
- 1) **Extract** data
- 2) **Transform** extracted data
- 3) **Load** transformed data to another database

Data pipelines

- Move data from one system to another
- May follow ETL
- Data may not be transformed
- Data may be directly loaded in applications

Summary

- What a data pipeline is
- What it does
- Why it's important
- How data pipelines are implemented at Spotflix
- What ETL is and its nuances

Let's practice!

UNDERSTANDING DATA ENGINEERING

Data structures

UNDERSTANDING DATA ENGINEERING



Structured data

- Easy to search and organize
- Consistent model, rows and columns
- Defined types
- Can be grouped to form relations
- Stored in relational databases
- About 20% of the data is structured
- Created and queried using SQL

Employee table

index	last_name	first_name	role	team	full_time	office
0	Thien	Vivian	Data Engineer	Data Science	1	Belgium
1	Huong	Julian	Data Scientist	Data Science	1	Belgium
2	Duplantier	Norbert	Software Developer	Infrastructure	1	United Kingdom
3	McColgan	Jeff	Business Developer	Sales	1	United States
4	Sanchez	Rick	Support Agent	Customer Service	0	United States

Relational database

office	address	number	city	zipcode
Belgium	Martelarenlaan	38	Leuven	3010
UK	Old Street	207	London	EC1V 9NR
USA	5th Ave	350	New York	10118

Relational database

index	last_name	first_name	office	address	number	city	zipcode
0	Thien	Vivian	Belgium	Martelarenlaan	38	Leuven	3010
1	Huong	Julian	Belgium	Martelarenlaan	38	Leuven	3010
2	Duplantier	Norbert	UK	Old Street	207	London	EC1V 9NR
3	McColgan	Jeff	USA	5th Ave	350	New York	10118
4	Sanchez	Rick	USA	5th Ave	350	New York	10118

Semi-structured data

- Relatively easy to search and organize
- Consistent model, less-rigid implementation: different observations have different sizes
- Different types
- Can be grouped, but needs more work
- NoSQL databases: JSON, XML, YAML

Favorite artists JSON file

```
{  
  {"user_1645156":  
    "last_name": "Lacroix",  
    "first_name": "Hadrien",  
    "favorite_artists": ["Fools in Deed", "Gojira", "Pain", "Nanowar of Steel"]},  
  {"user_5913764":  
    "last_name": "Billen",  
    "first_name": "Sara",  
    "favorite_artists": ["Tamino", "Taylor Swift"]},  
  {"user_8436791":  
    "last_name": "Sulmont",  
    "first_name": "Lis",  
    "favorite_artists": ["Arctic Monkeys", "Rihanna", "Nina Simone"]},  
  ...  
}
```

Unstructured data

- Does not follow a model, can't be contained in rows and columns
- Difficult to search and organize
- Usually text, sound, pictures or videos
- Usually stored in data lakes, can appear in data warehouses or databases
- Most of the data is unstructured
- Can be extremely valuable

Una mattina mi son alzato
O bella ciao, bella ciao, bella ciao, ciao, ciao
Una mattina mi son alzato
E ho trovato l'invasor

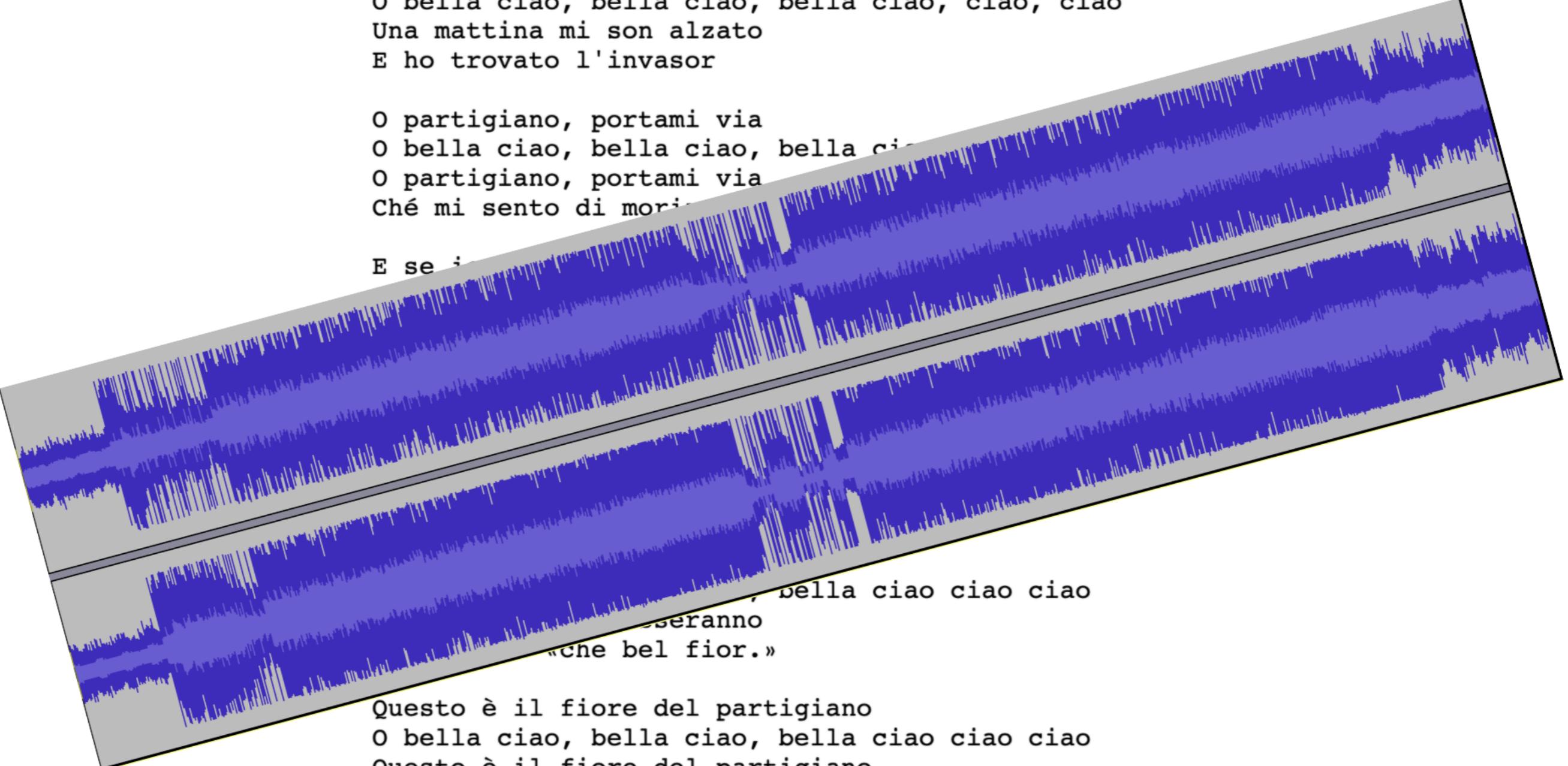
O partigiano, portami via
O bella ciao, bella ciao, bella ciao, ciao, ciao
O partigiano, portami via
Ché mi sento di morir

E se io muoio da partigiano
O bella ciao, bella ciao, bella ciao, ciao, ciao
E se io muoio da partigiano
Tu mi devi seppellir

E seppellire lassù in montagna
O bella ciao, bella ciao, bella ciao, ciao, ciao
E seppellire lassù in montagna
Sotto l'ombra di un bel fior

E le genti che passeranno
O bella ciao, bella ciao, bella ciao ciao ciao
E le genti che passeranno
Mi diranno «che bel fior.»

Questo è il fiore del partigiano
O bella ciao, bella ciao, bella ciao ciao ciao
Questo è il fiore del partigiano
Morto per la libertà



Una mattina mi son alzato
O bella ciao, bella ciao, bella ciao, ciao, ciao
Una mattina mi son alzato
E ho trovato l'invasor

O partigiano, portami via
O bella ciao, bella ciao, bella ciao
O partigiano, portami via
Ché mi sento di morir

E se :

, bella ciao ciao ciao
seranno
"che bel fior."

Questo è il fiore del partigiano
O bella ciao, bella ciao, bella ciao ciao ciao
Questo è il fiore del partigiano
Morto per la libertà

Una mattina mi son alzato
O bella ciao, bella ciao, bella ciao, ciao, ciao

Una mattina mi son alzato
E ho tro

O parti
O bella
O part
Ché mi

E se

Questo è il fiore dei partigiani
O bella ciao, bella ciao, bella ciao
Questo è il fiore del partigiano
Morto per la libertà





Adding some structure

- Use AI to search and organize unstructured data
- Add information to make it semi-structured

Summary

- Structured data
- Semi-structured data
- Unstructured data
- Differences between the three
- Give examples

Let's practice!

UNDERSTANDING DATA ENGINEERING

SQL databases

UNDERSTANDING DATA ENGINEERING



SQL

- Structured Query Language
- Industry standard for Relational Database Management System (RDBMS)
- Allows you to access many records at once, and group, filter or aggregate them
- Close to written English, easy to write and understand
- Data engineers use SQL to create and maintain databases
- Data scientists use SQL to query (request information from) databases

Remember the employees table

index	last_name	first_name	role	team	full_time	office
0	Thien	Vivian	Data Engineer	Data Science	1	Belgium
1	Huong	Julian	Data Scientist	Data Science	1	Belgium
2	Duplantier	Norbert	Software Developer	Infrastructure	1	United Kingdom
3	McColgan	Jeff	Business Developer	Sales	1	United States
4	Sanchez	Rick	Support Agent	Customer Service	0	United States

SQL for data engineers

- Data engineers use SQL to create, maintain and update tables.

```
CREATE TABLE employees (
    employee_id INT,
    first_name VARCHAR(255),
    last_name VARCHAR(255),
    role VARCHAR(255),
    team VARCHAR(255),
    full_time BOOLEAN,
    office VARCHAR(255)
);
```

SQL for data scientists

- Data scientist use SQL to query, filter, group and aggregate data in tables.

```
SELECT first_name, last_name  
FROM employees  
WHERE role LIKE '%Data%'
```

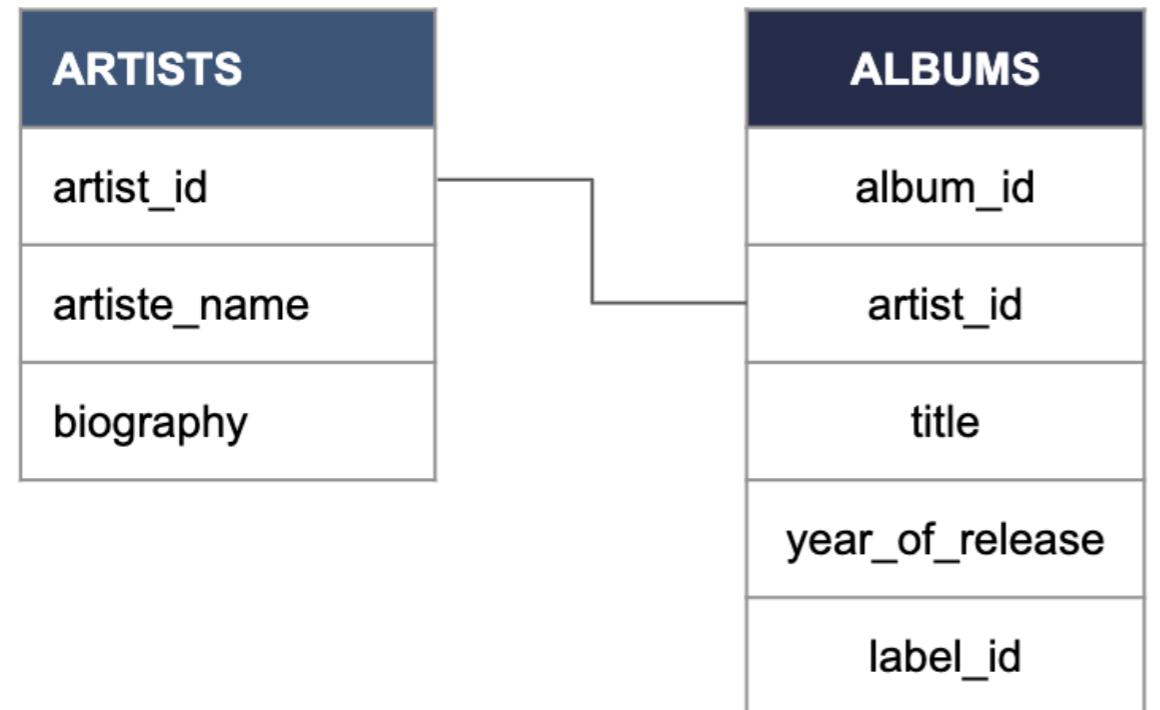
Database schema

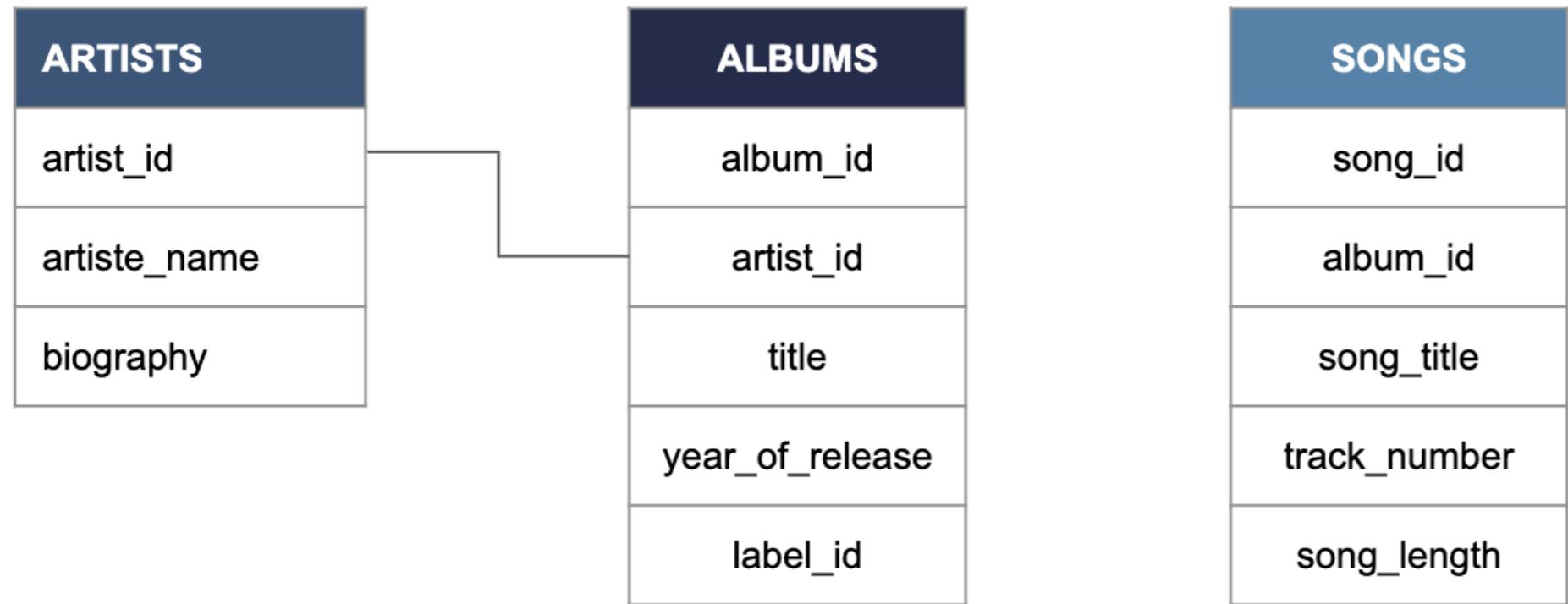
- Databases are made of tables
- The database schema governs how tables are related

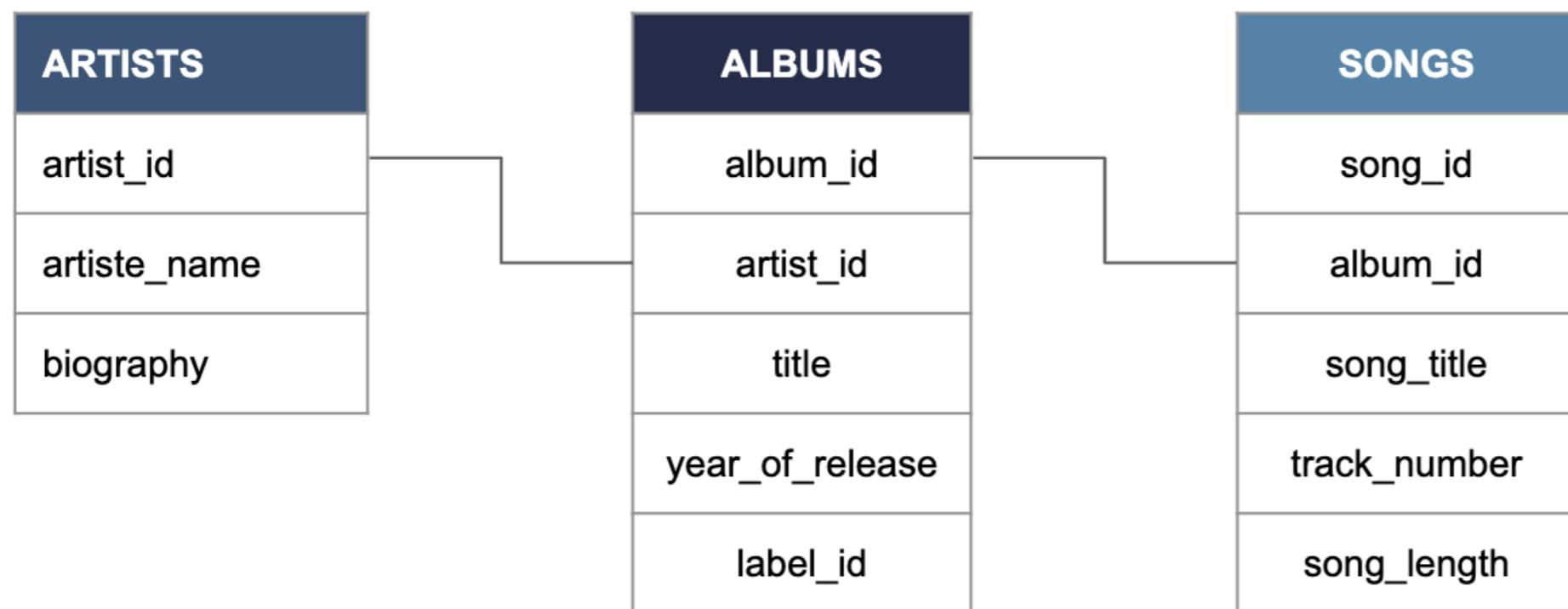
ALBUMS
album_id
artist_id
title
year_of_release
label_id

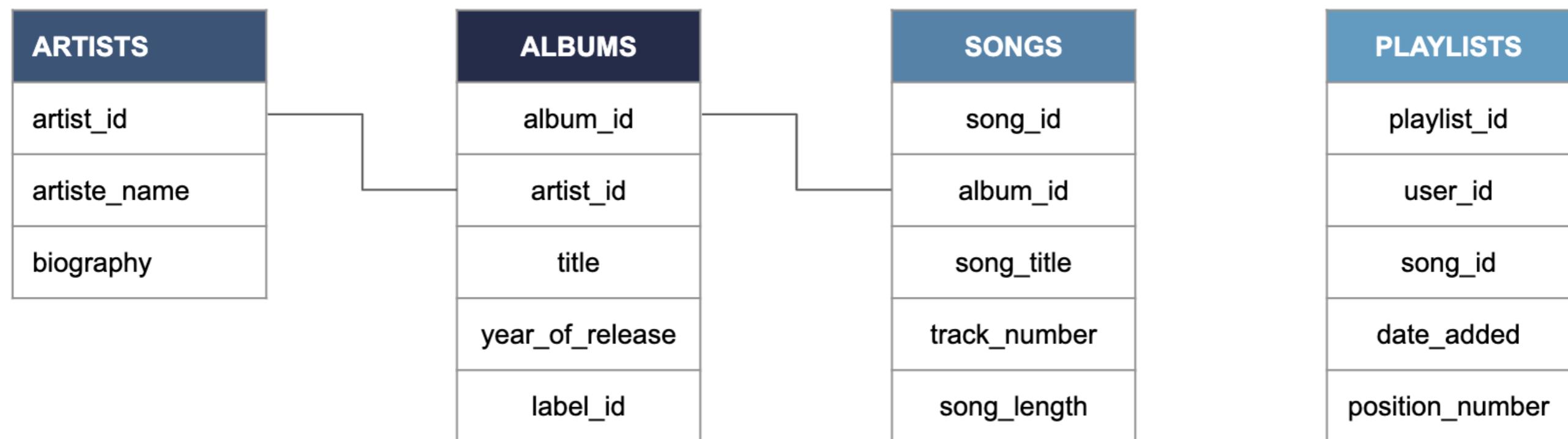
ARTISTS
artist_id
artiste_name
biography

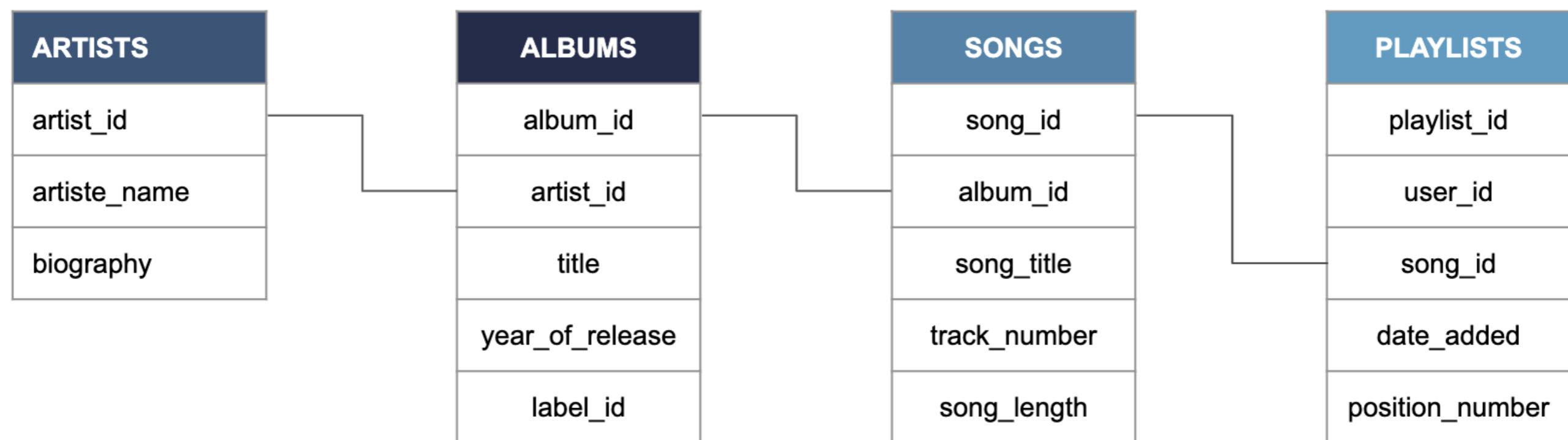
ALBUMS
album_id
artist_id
title
year_of_release
label_id











Several implementations

- SQLite
- MySQL
- PostgreSQL
- Oracle SQL
- SQL Server

Summary

- SQL = industry standard
- Explain how Data engineers and Data scientists use it differently
- Database schema
- SQL implementations

Let's practice!

UNDERSTANDING DATA ENGINEERING

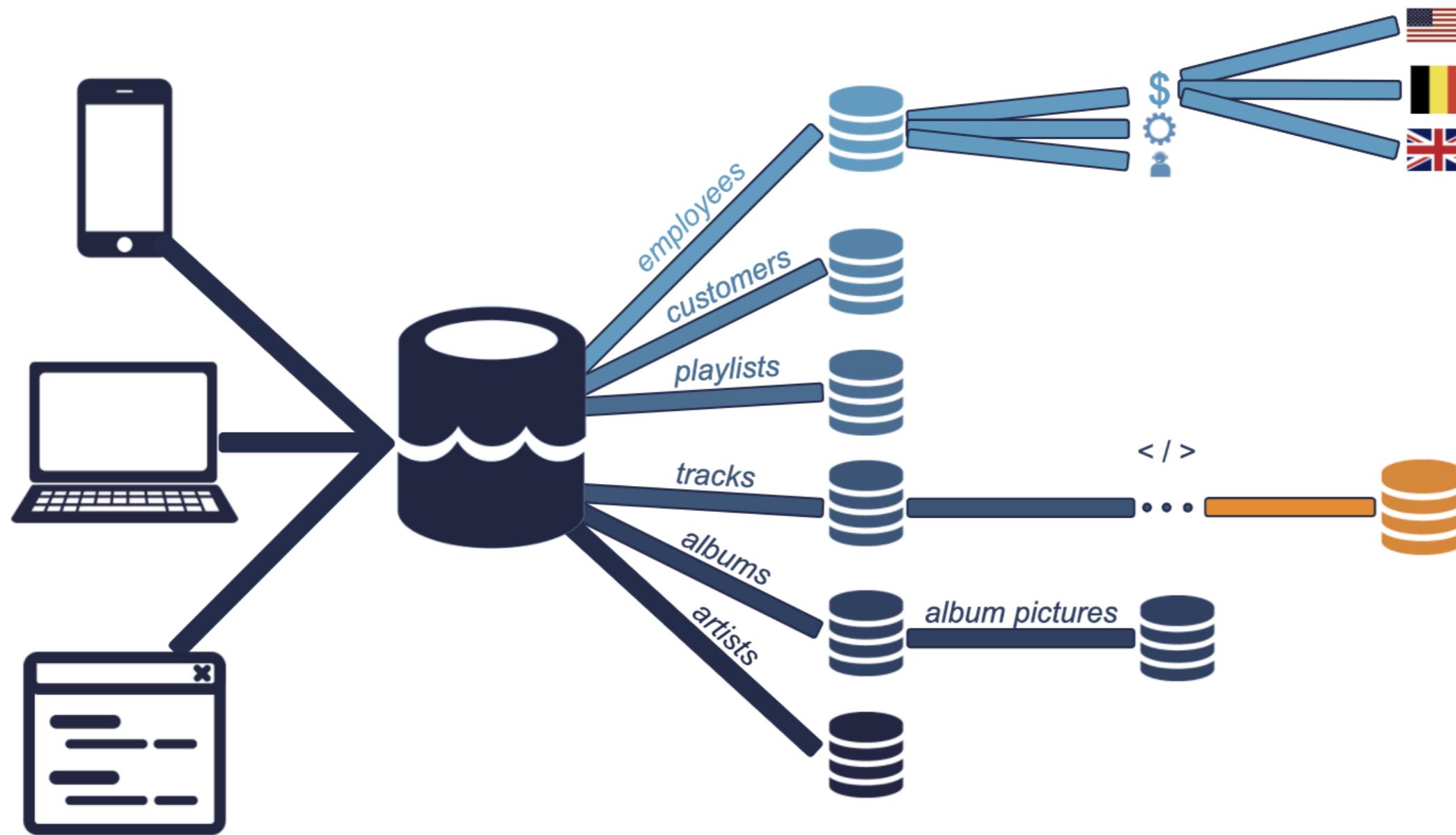
Data warehouses and data lakes

UNDERSTANDING DATA ENGINEERING



Warehouses with stunning view on the lake





Data lakes and data warehouses

Data lake

- Stores all the raw data
- Can be petabytes (1 million GBs)
- Stores all data structures
- Cost-effective
- Difficult to analyze
- Requires an up-to-date data catalog
- Used by data scientists
- Big data, real-time analytics

Data warehouse

- Specific data for specific use
- Relatively small
- Stores mainly structured data
- More costly to update
- Optimized for data analysis
- Also used by data analysts and business analysts
- Ad-hoc, read-only queries

Data catalog for data lakes

- What is the source of this data?
- Where is this data used?
- Who is the owner of the data?
- How often is this data updated?
- Good practice in terms of data governance
- Ensures reproducibility
- No catalog --> data swamp
- **Good practice for any data storage solution**
 - Reliability
 - Autonomy
 - Scalability
 - Speed

Database vs. data warehouse

- Database:
 - General term
 - Loosely defined as *organized data stored and accessed on a computer*
- Data warehouse is a type of database

Summary

- Data lakes
- Data warehouses
- Databases
- Data catalog

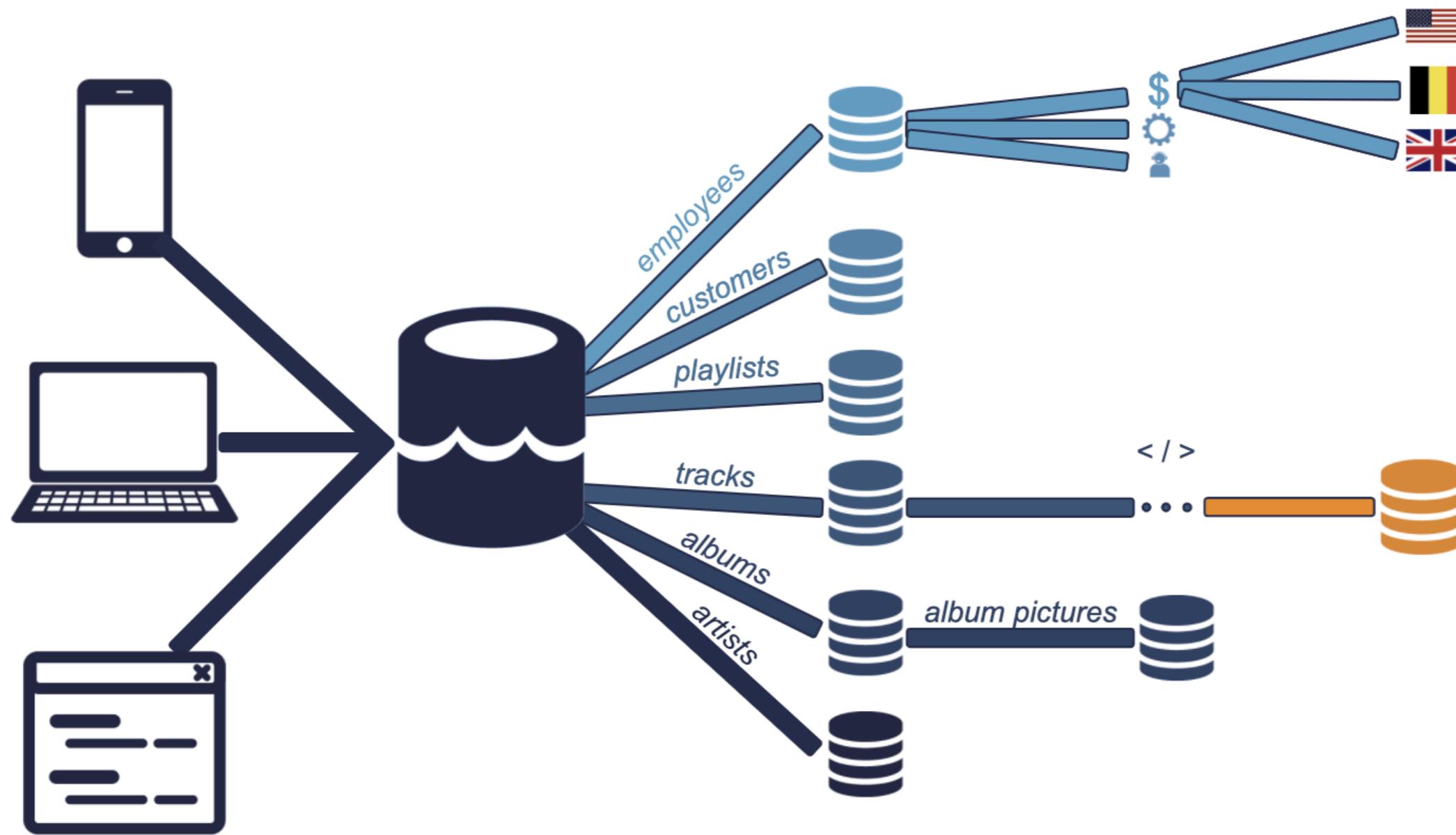
Let's practice!

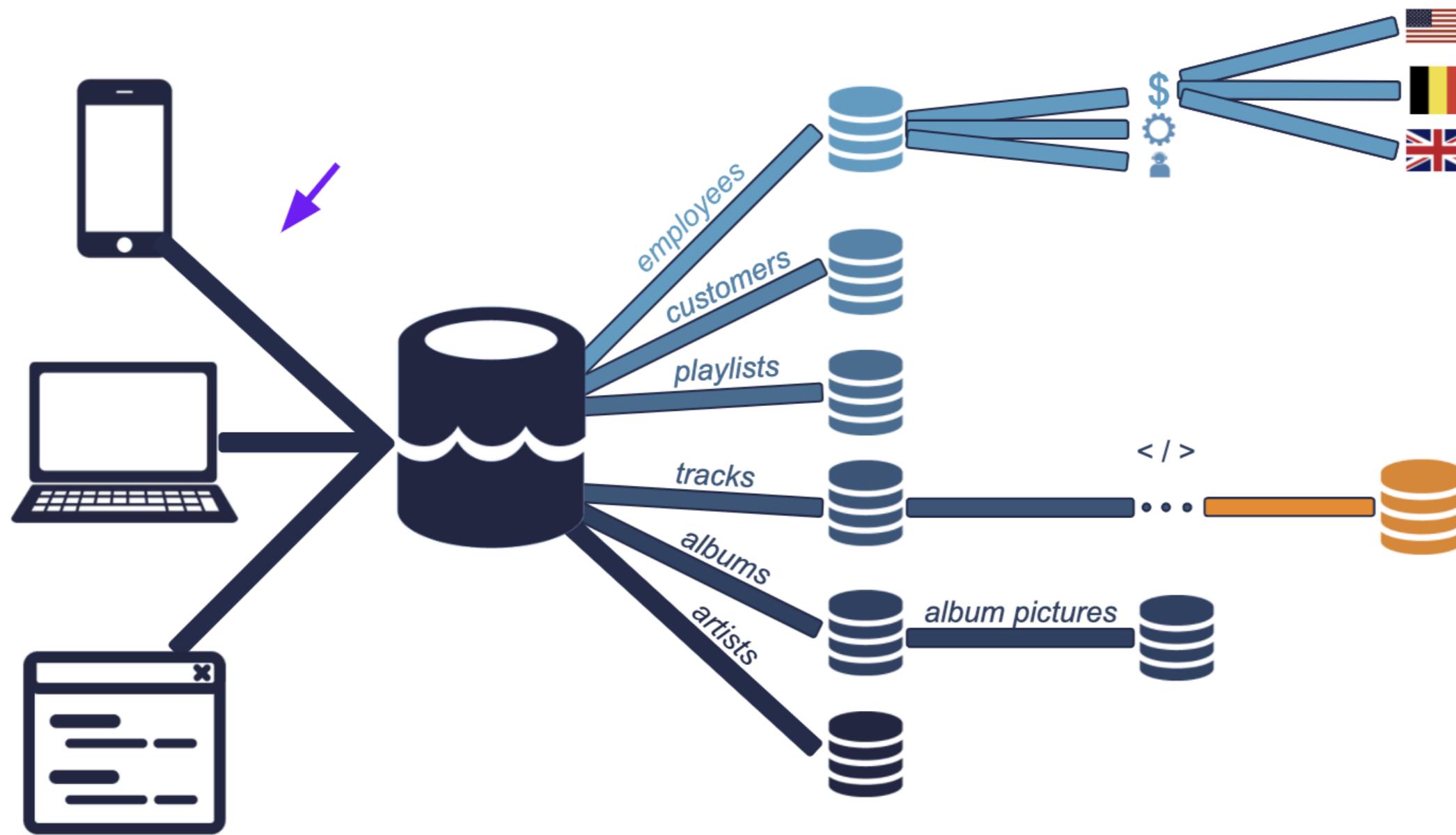
UNDERSTANDING DATA ENGINEERING

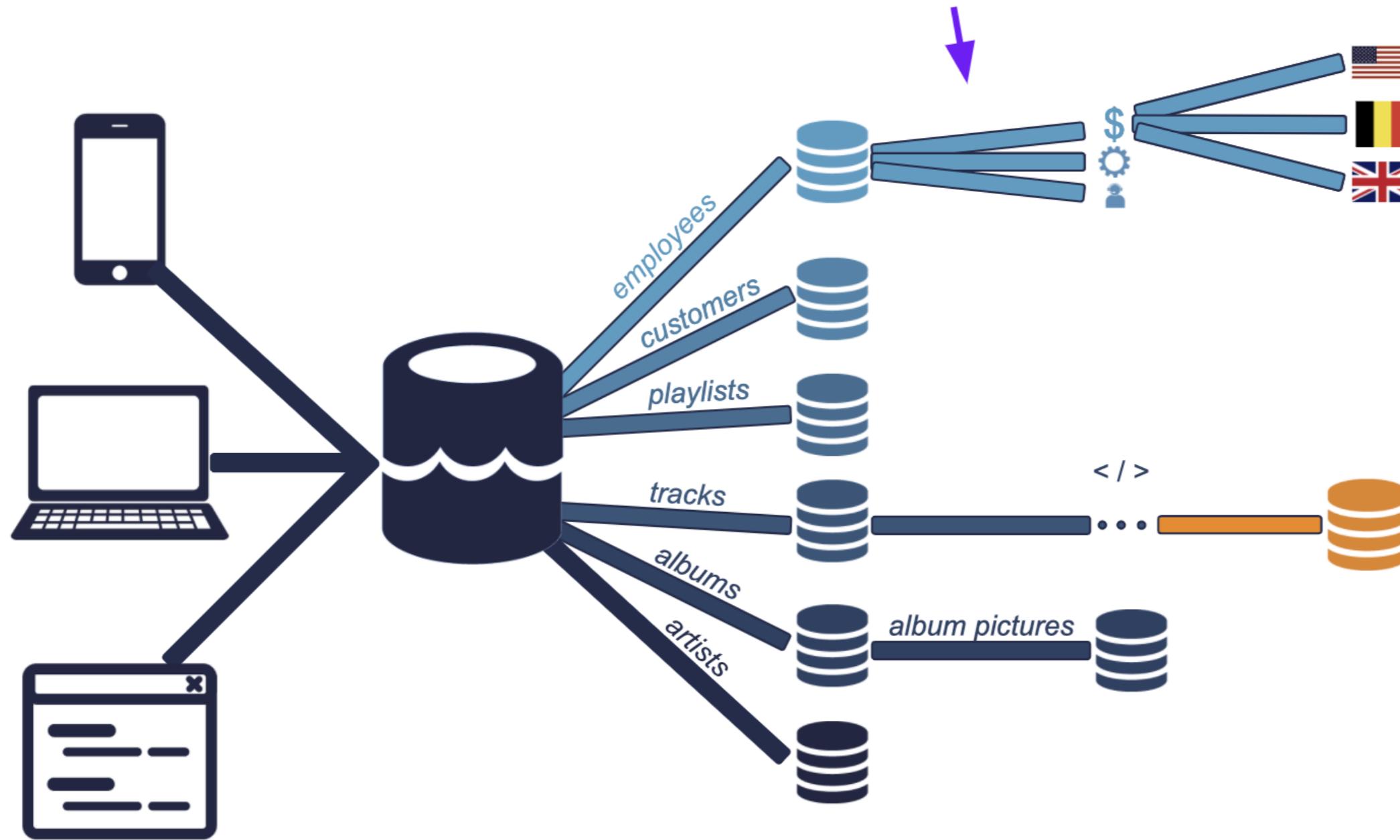
Processing data

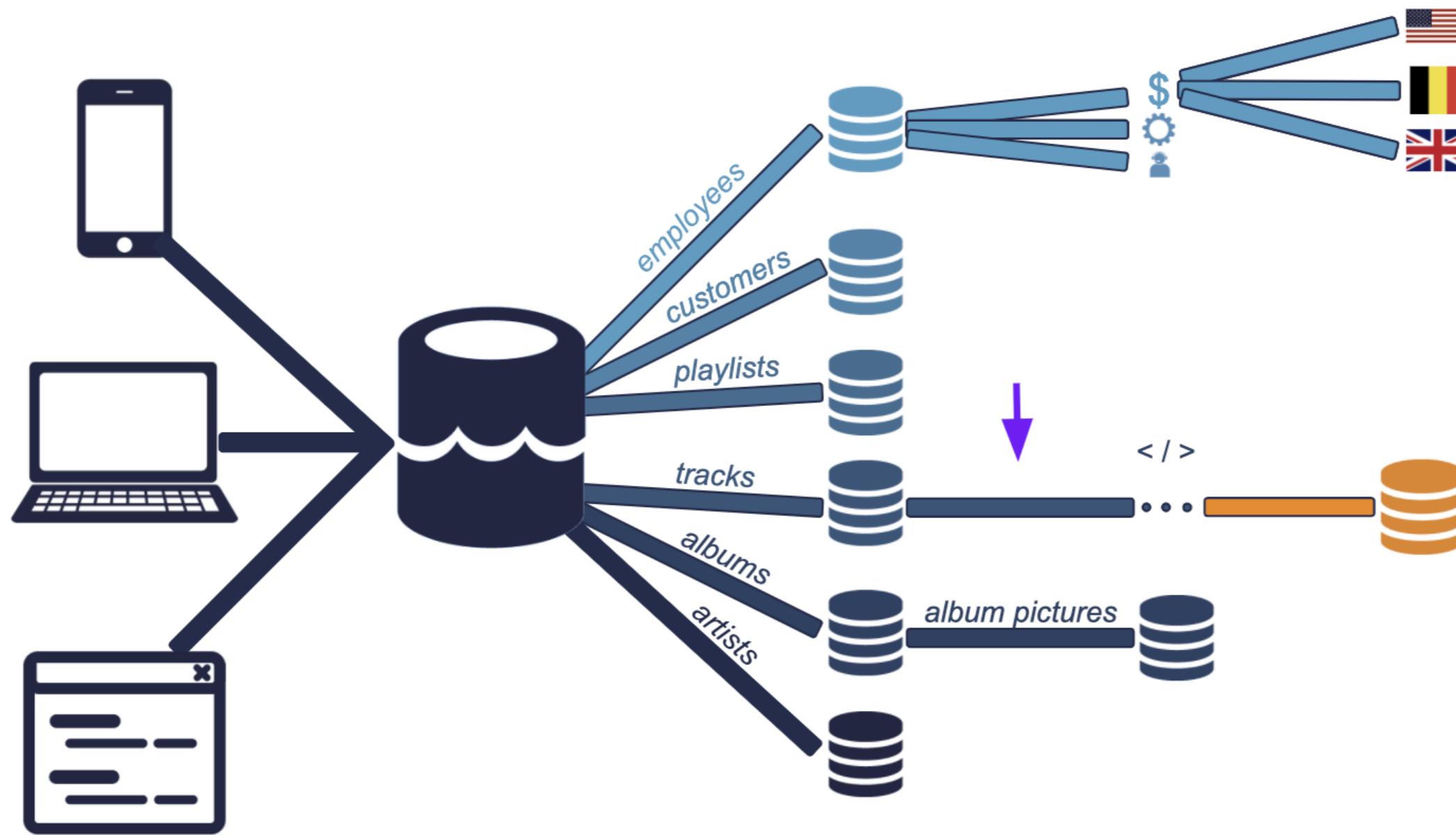
UNDERSTANDING DATA ENGINEERING











A general definition

- Data processing: converting **raw** data into **meaningful** information

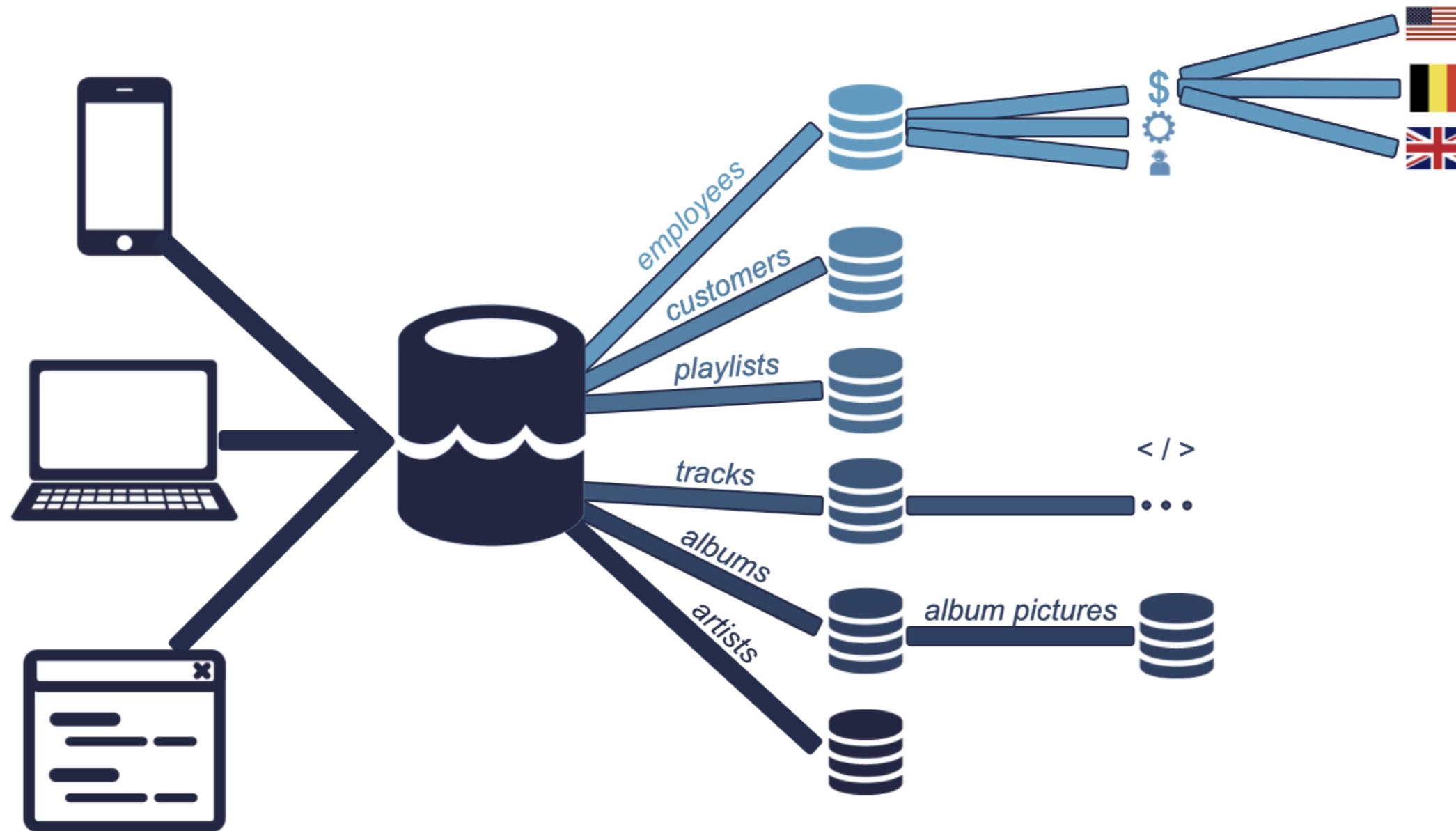
Data processing value

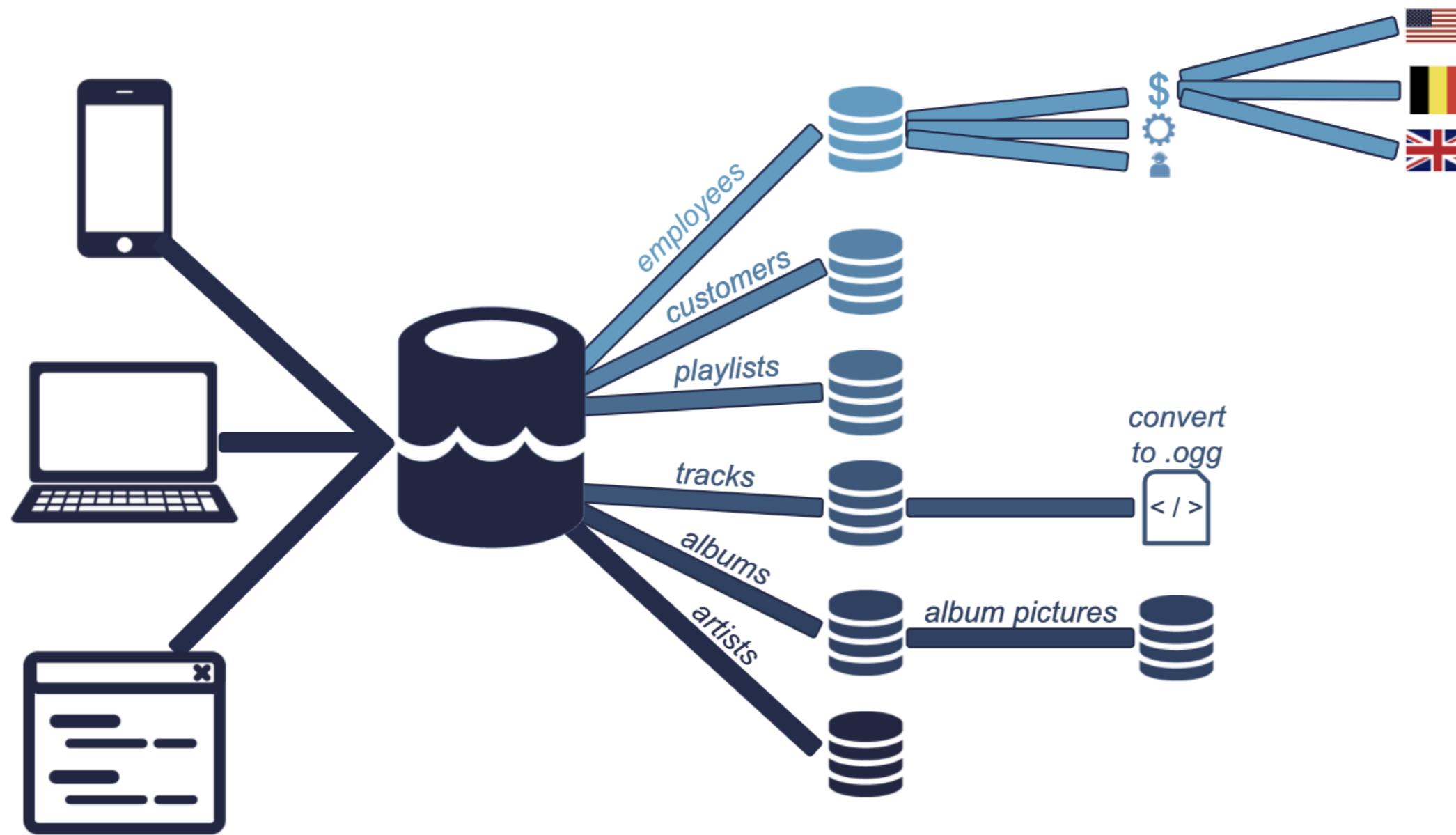
Conceptually

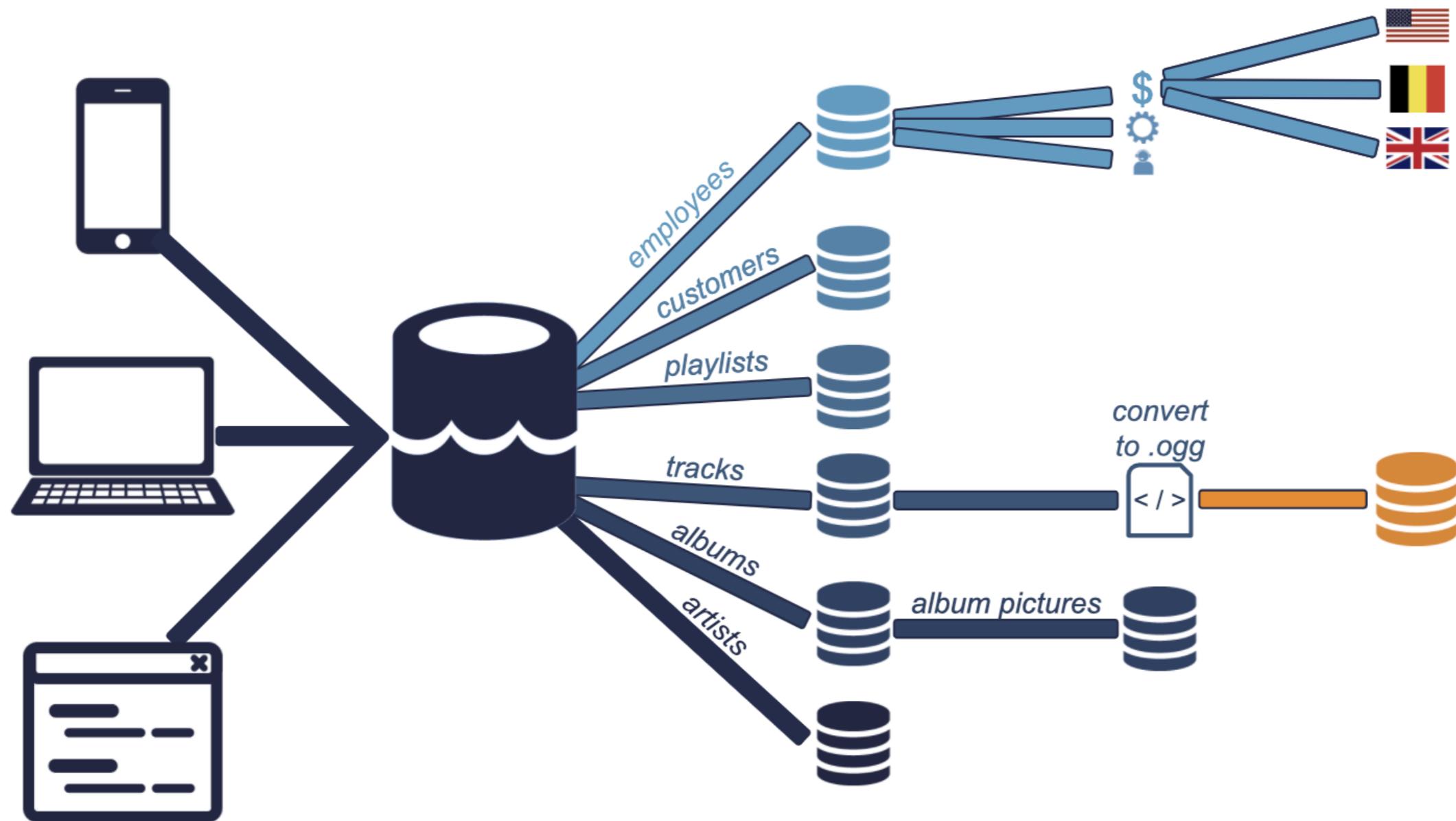
- Remove unwanted data
- Optimize memory, process and network costs
- Convert data from one type to another

At Spotflix

- No long term need for testing feature data
- Can't afford to store and stream files this big







Data processing value

Conceptually

- Remove unwanted data
- To save memory
- Convert data from one type to another
- Organize data
- To fit into a schema/structure
- Increase productivity

At Spotflix

- No need for lossless format
- Can't afford to store files this big
- Convert songs from `.flac` to `.ogg`
- Reorganize data from the data lake to data warehouses
- Employee table example
- Enable data scientists

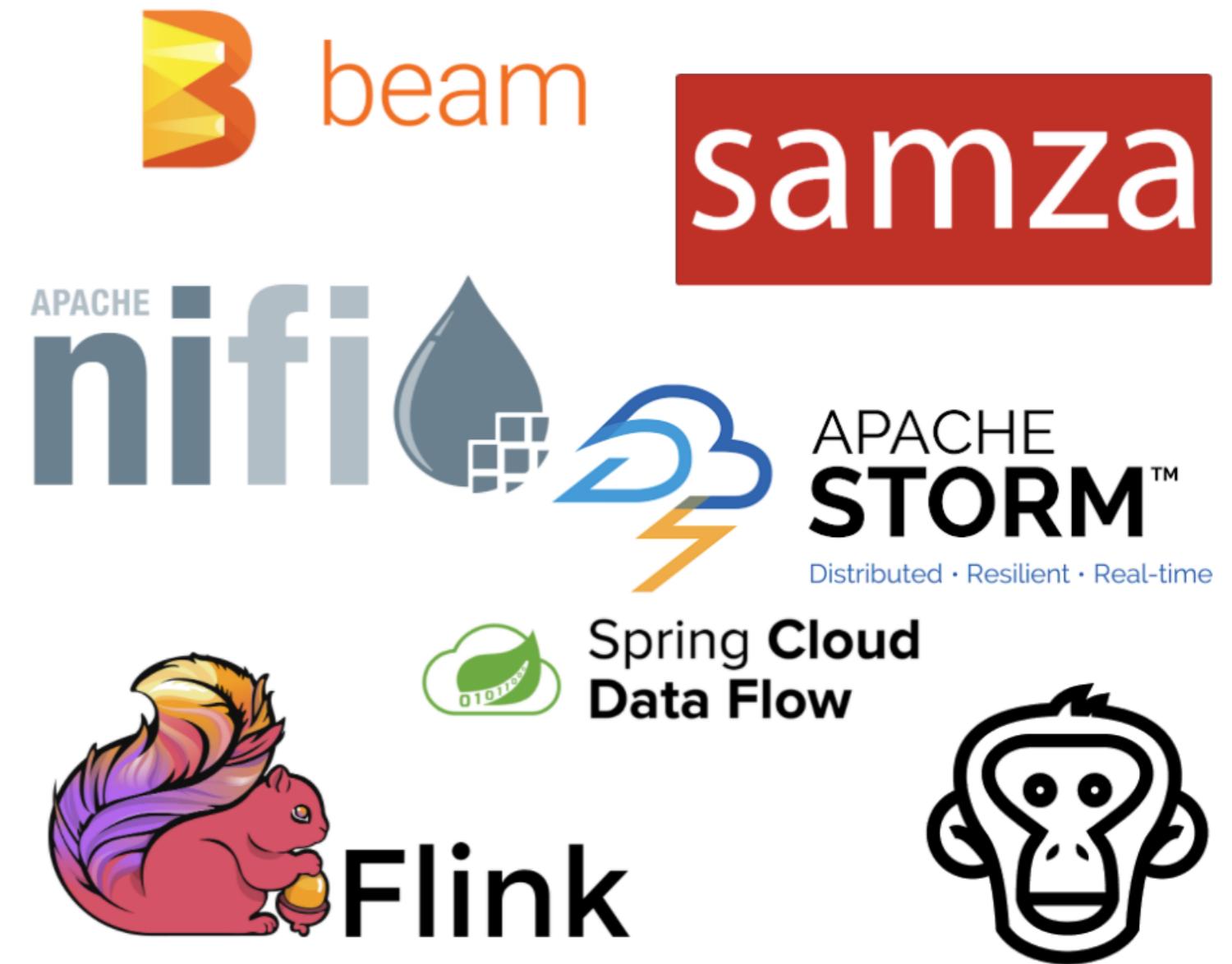
How data engineers process data

- Data manipulation, cleaning, and tidying tasks
 - that can be automated
 - that will always need to be done
- Store data in a sanely structured database
- Create views on top of the database tables
- Optimizing the performance of the database
- Rejecting corrupt song files
- Deciding what happens with missing metadata
- Separate artists and albums tables...
- ...but provide view combining them
- Indexing

Batch processing



Stream processing



¹ The difference between batch and stream will be explained in the next lesson!



Summary

- What data processing is
- Why it's necessary
- What it consists in
- How we process data at Spotflix

Let's practice!

UNDERSTANDING DATA ENGINEERING

Scheduling data

UNDERSTANDING DATA ENGINEERING

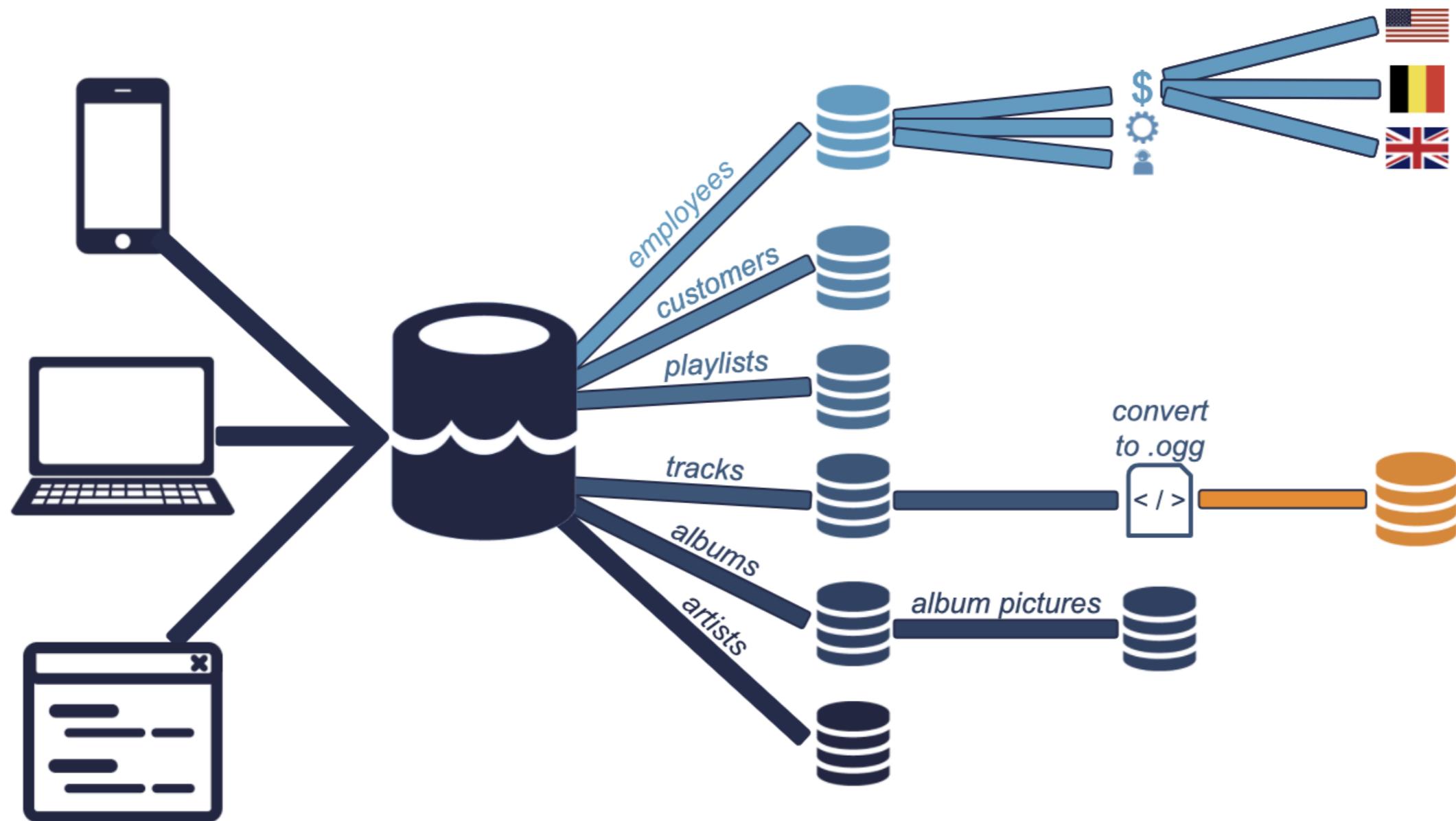


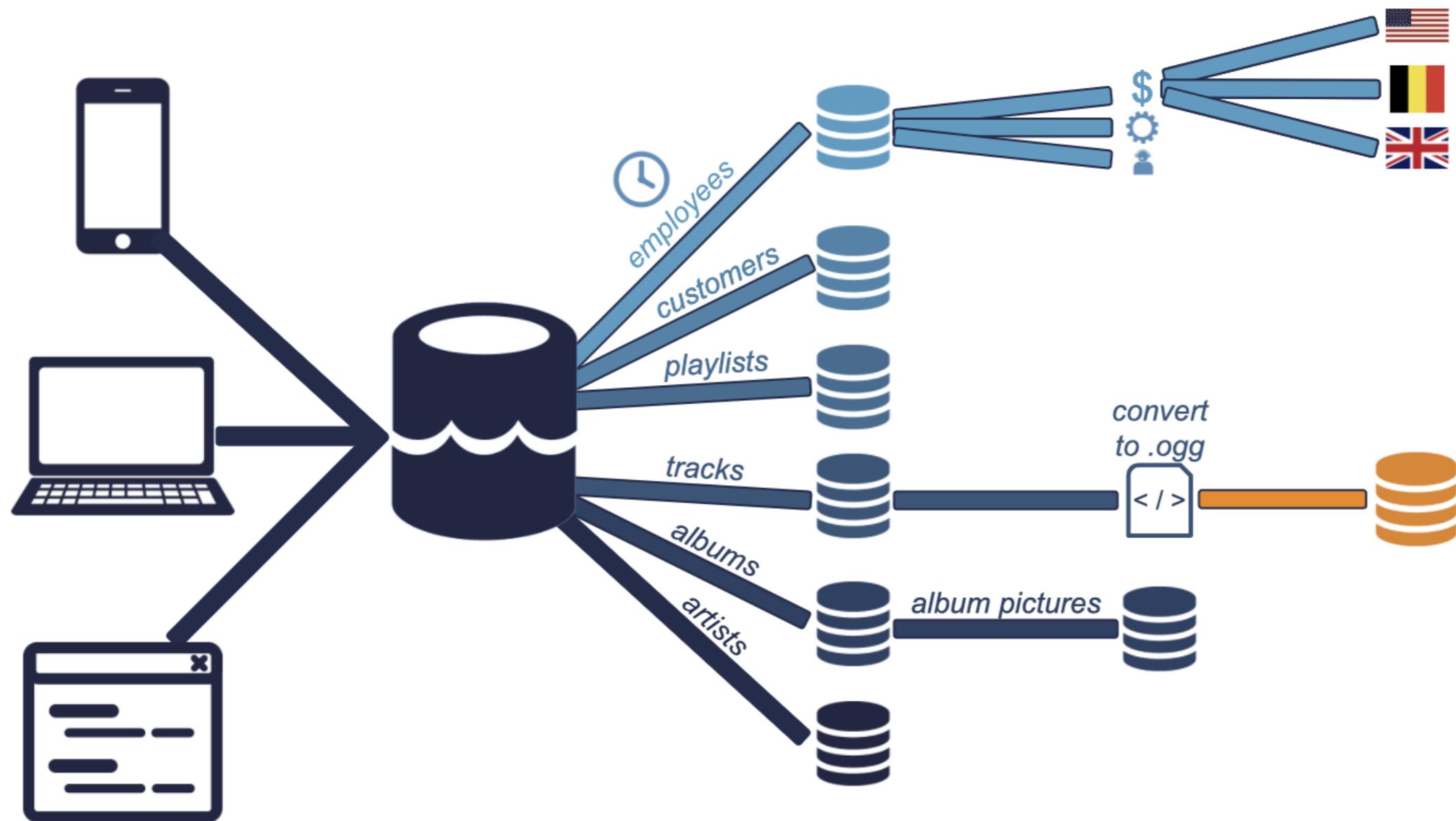
Scheduling

- Can apply to any task listed in data processing
- Scheduling is the glue of your system
- Holds each piece and organize how they work together
- Runs tasks in a specific order and resolves all dependencies

Manual, time and sensor scheduling

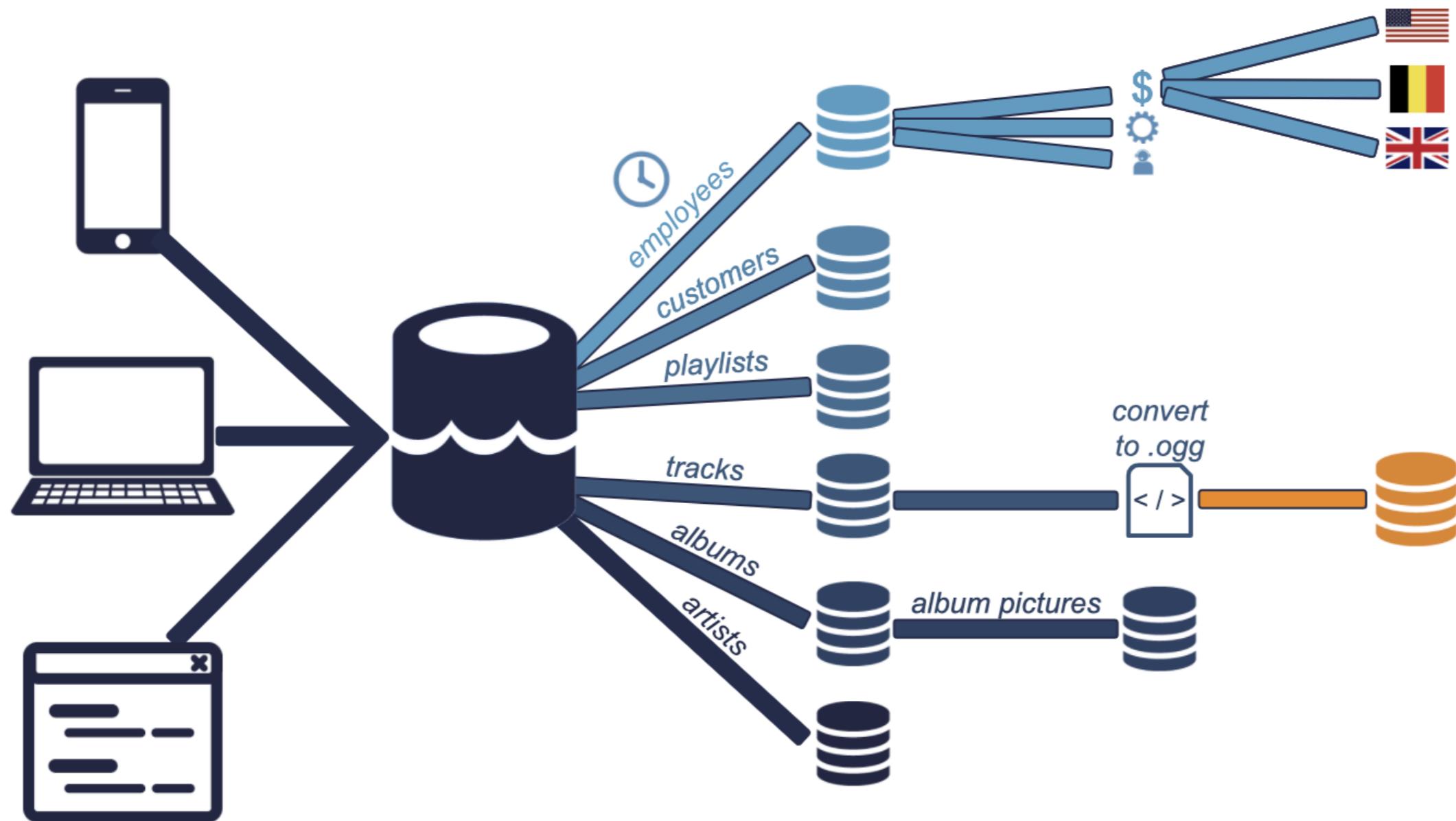
- Manually
 - Manually update the employee table

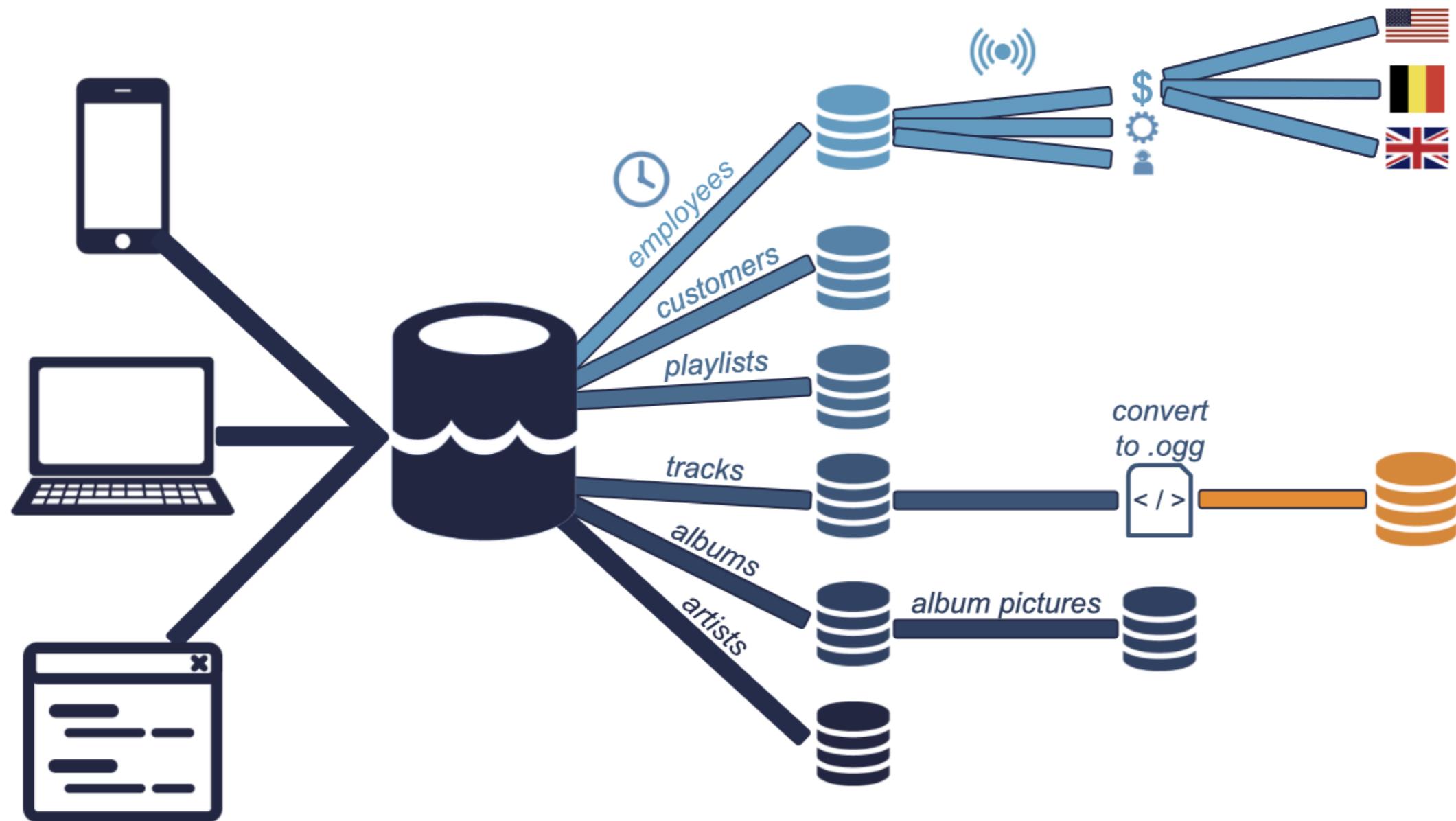




Manual, time and sensor scheduling

- Manually
- Automatically run at a specific time
- Automatically run if a specific condition is met
 - Sensor scheduling
- Manually update the employee table
- Update the employee table at 6 AM





Manual, time, and sensor scheduling

- Manually
- Automatically run at a specific time
- Automatically run if a specific condition is met
 - Sensor scheduling
- Manually update the employee table
- Update the employee table at 6 AM
- Update the department tables if a new employee was added

Batches and streams

- Batches
 - Group records at intervals
 - Often cheaper
- Streams
 - Send individual records right away
 - Songs uploaded by artists
 - Employee table
 - Revenue table
 - New users signing in
 - Another example: online vs. offline listening

Scheduling tools



Summary

- What scheduling is
- Different ways to set it up
- Difference between batches and streams
- How scheduling is implemented at Spotflix
- Airflow, Luigi

Let's practice!

UNDERSTANDING DATA ENGINEERING

Parallel computing

UNDERSTANDING DATA ENGINEERING



Parallel computing

- Basis of modern data processing tools
- Necessary:
 - Mainly because of memory
 - Also for processing power
- How it works:
 - Split tasks up into several smaller subtasks
 - Distribute these subtasks over several computers



x 1,000

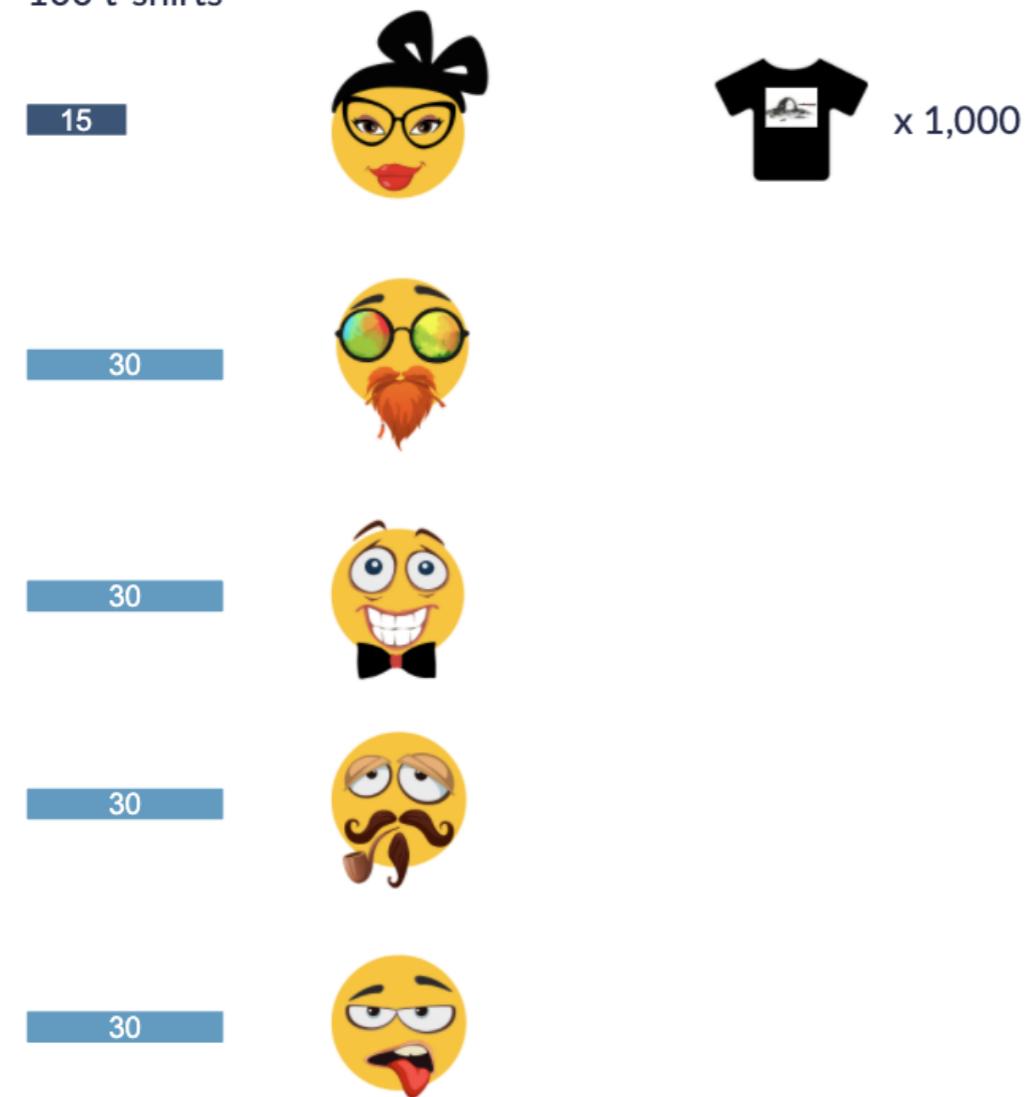
Time for
100 t-shirts

15



x 1,000

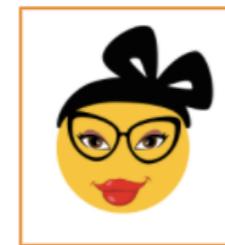
Time for
100 t-shirts



¹ Emojis by Mohamed Hassan

Time for
100 t-shirts

15



x 1,000

30



30



30



30



Time for
100 t-shirts

15



x 1,000

30



x 250

30



x 250

30



x 250

30

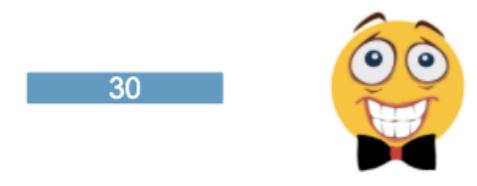
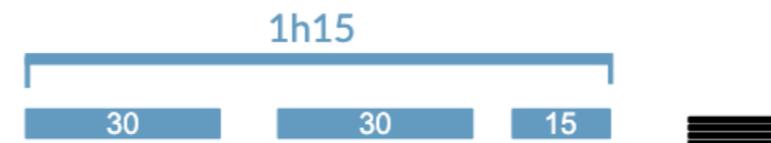


x 250

Time for
100 t-shirts



Time for 1,000 t-shirts



Time for
100 t-shirts



x 1,000

Time for 1,000 t-shirts

2h30



30



x 250



30



x 250



30



x 250



30



x 250



Benefits and risks of parallel computing

- Employees = processing units
- Advantages
 - Extra processing power
 - Reduced memory footprint
- Disadvantages
 - Moving data incurs a cost
 - Communication time

Time for
100 t-shirts



x 1,000

Time for 1,000 t-shirts

2h30



30



x 250



30



x 250



30



x 250



30



x 250



Time for
100 t-shirts



x 1,000

Time for 1,000 t-shirts

2h30



30



0h10
x 250



30



x 250



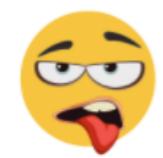
30



x 250



30



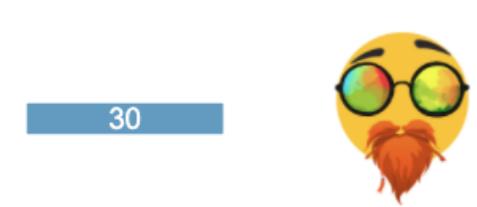
x 250



Time for
100 t-shirts



x 1,000



0h10
x 250



x 250



x 250

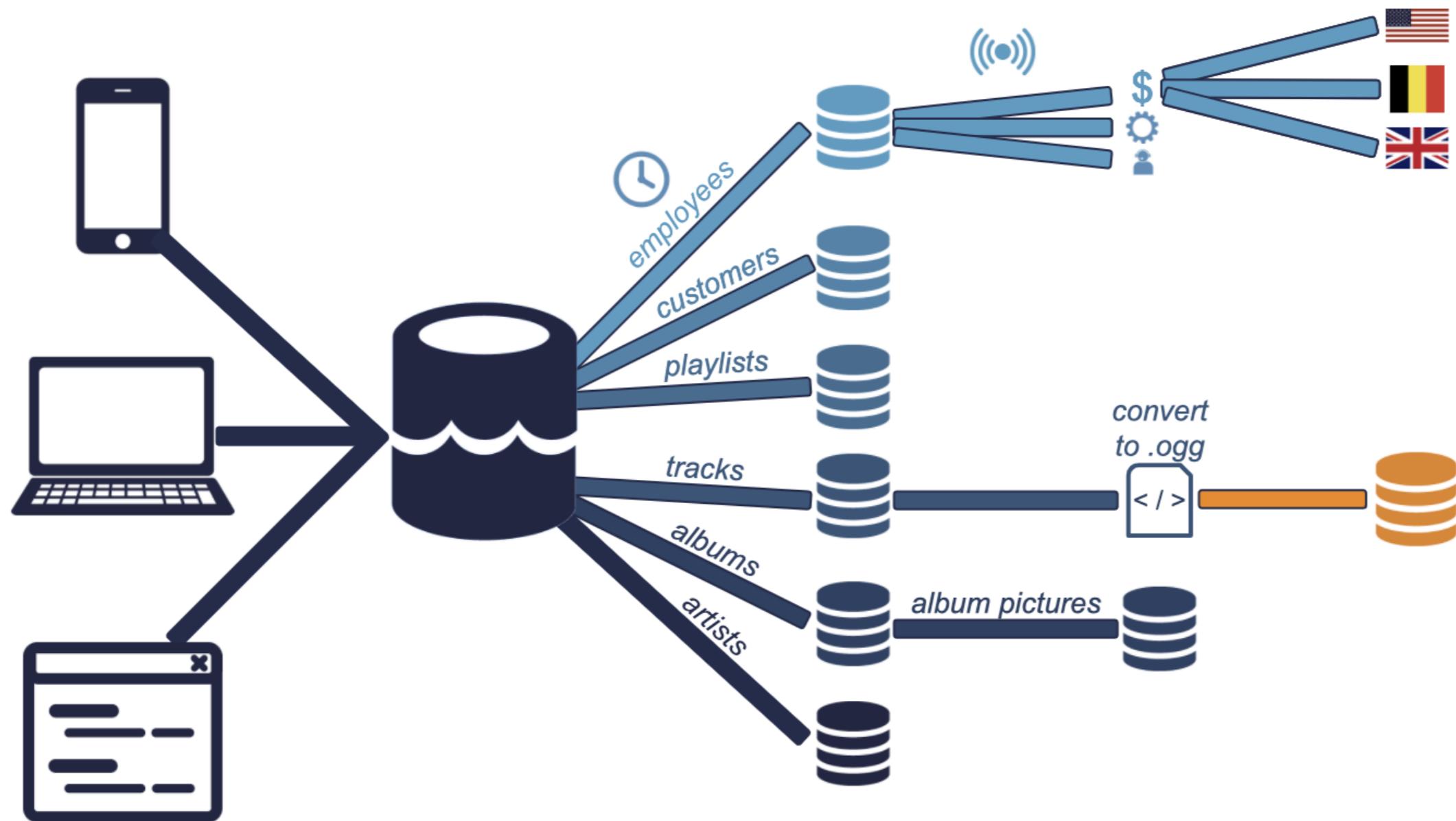


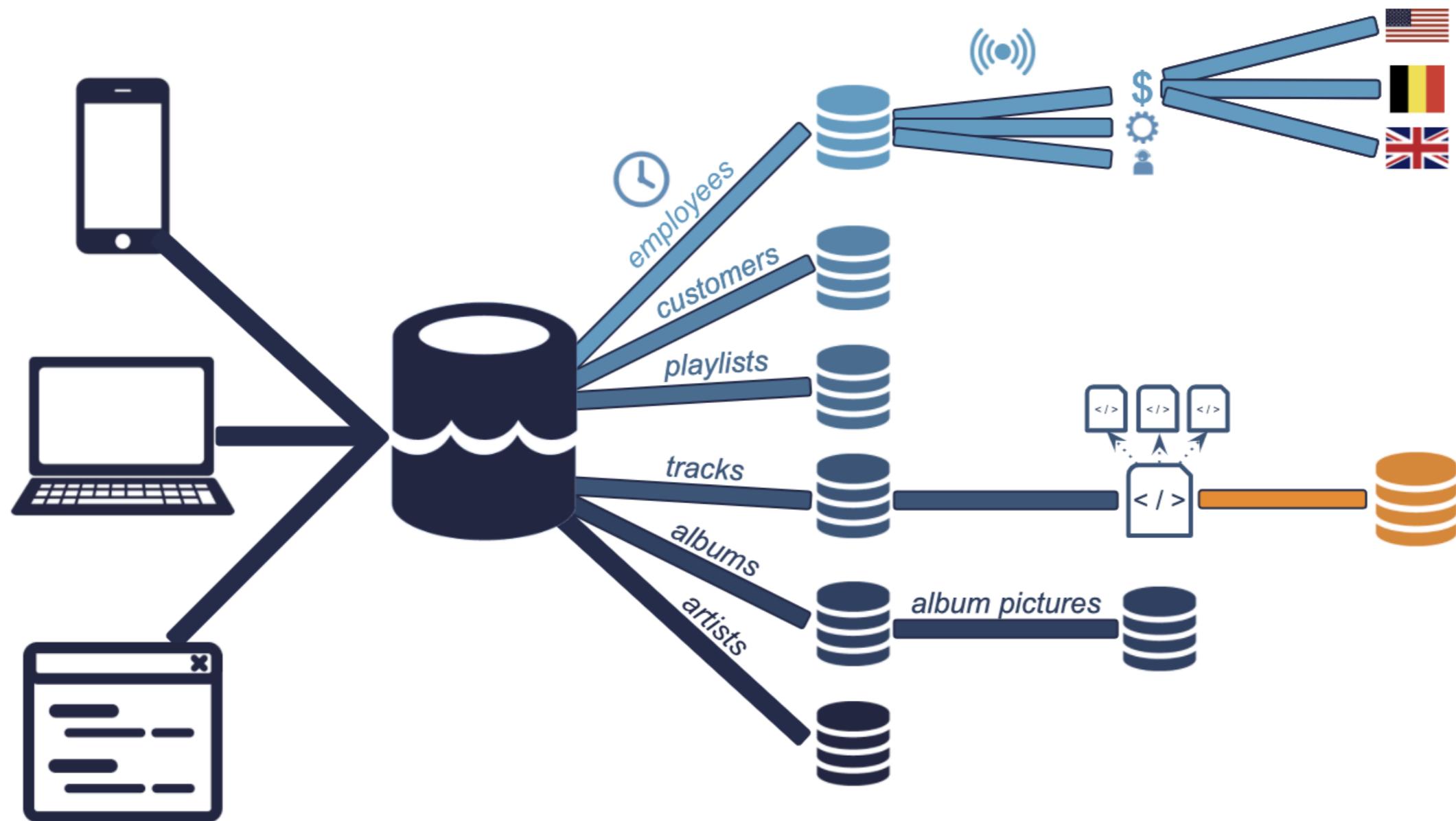
x 250

Time for 1,000 t-shirts

2h30







Summary

- Benefits and risks
- How it's implemented at Spotflix

Let's practice!

UNDERSTANDING DATA ENGINEERING

Cloud computing

UNDERSTANDING DATA ENGINEERING



Cloud computing for data processing

Servers on premises

- Bought
- Need space
- Electrical and maintenance cost
- Enough power for peak moments
- Processing power unused at quieter times

Servers on the cloud

- Rented
- Don't need space
- Use just the resources we need
- When we need them
- The closer to the user the better

Cloud computing for data storage

- Database reliability: data replication
- Risk with sensitive data



32.4%



32.4%



17.6%



32.4%



17.6%



6%

File storage





File storage

AWS S3





File storage

AWS S3



Azure
Blob Storage





File storage

AWS S3



Azure
Blob Storage



Google
Cloud Storage





File storage

AWS S3



Azure
Blob Storage



Google
Cloud Storage



Computation



File storage

AWS S3



Azure
Blob Storage



Google
Cloud Storage



Computation

AWS EC2





File storage

AWS S3



Azure
Blob Storage



Google
Cloud Storage



Computation

AWS EC2



Azure
Virtual Machines





File storage

AWS S3



Azure
Blob Storage



Google
Cloud Storage



Computation

AWS EC2



Azure
Virtual Machines



Google
Compute Engine





File storage

AWS S3



Azure
Blob Storage



Google
Cloud Storage



Computation

AWS EC2



Azure
Virtual Machines



Google
Compute Engine



Databases



File storage

AWS S3



Azure
Blob Storage



Google
Cloud Storage



Computation

AWS EC2



Azure
Virtual Machines



Google
Compute Engine



Databases

AWS RDS





File storage

AWS S3



Azure
Blob Storage



Google
Cloud Storage



Computation

AWS EC2



Azure
Virtual Machines



Google
Compute Engine



Databases

AWS RDS



Azure
SQL Database





File storage

AWS S3



Azure
Blob Storage



Google
Cloud Storage



Computation

AWS EC2



Azure
Virtual Machines



Google
Compute Engine



Databases

AWS RDS

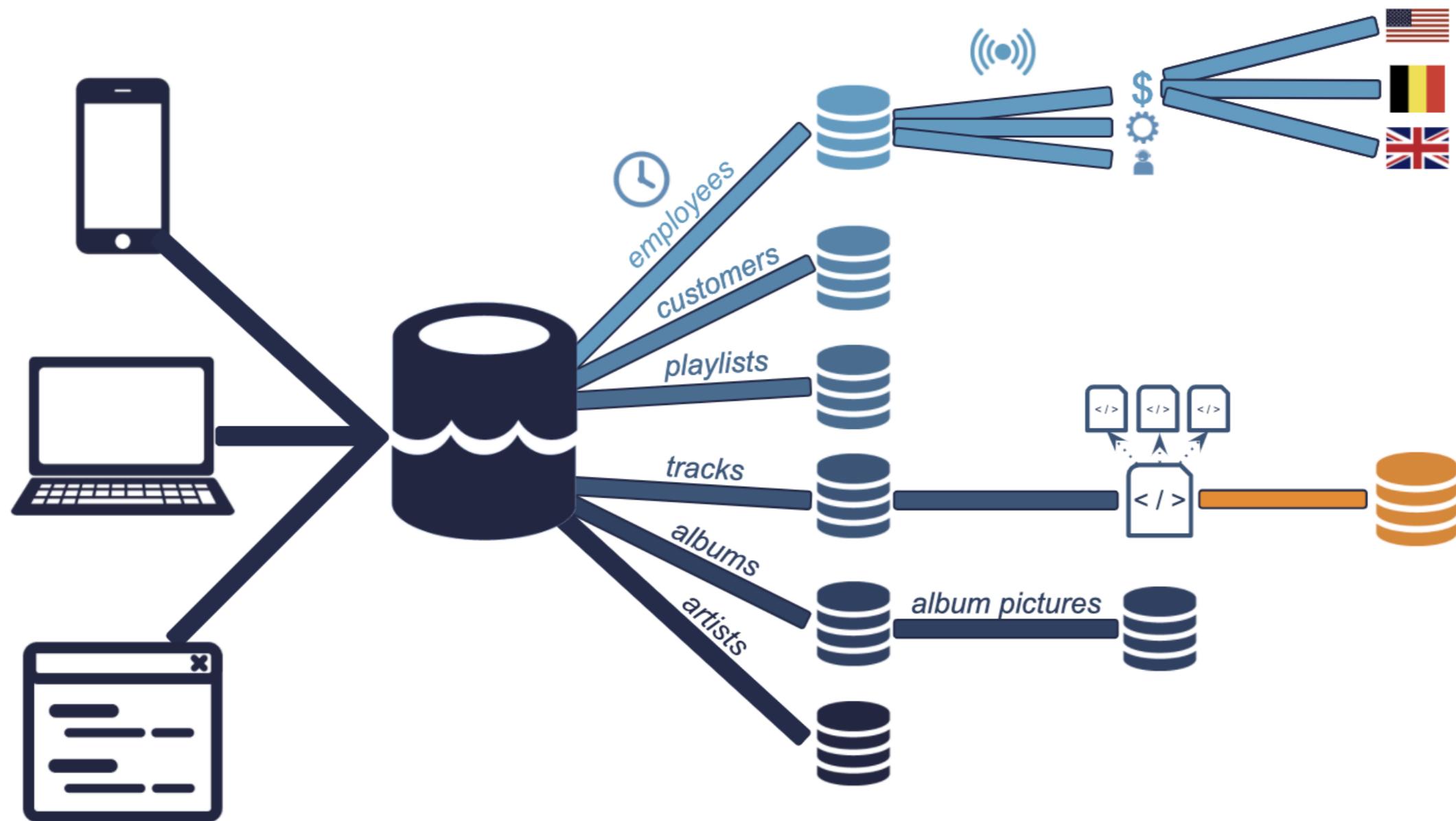


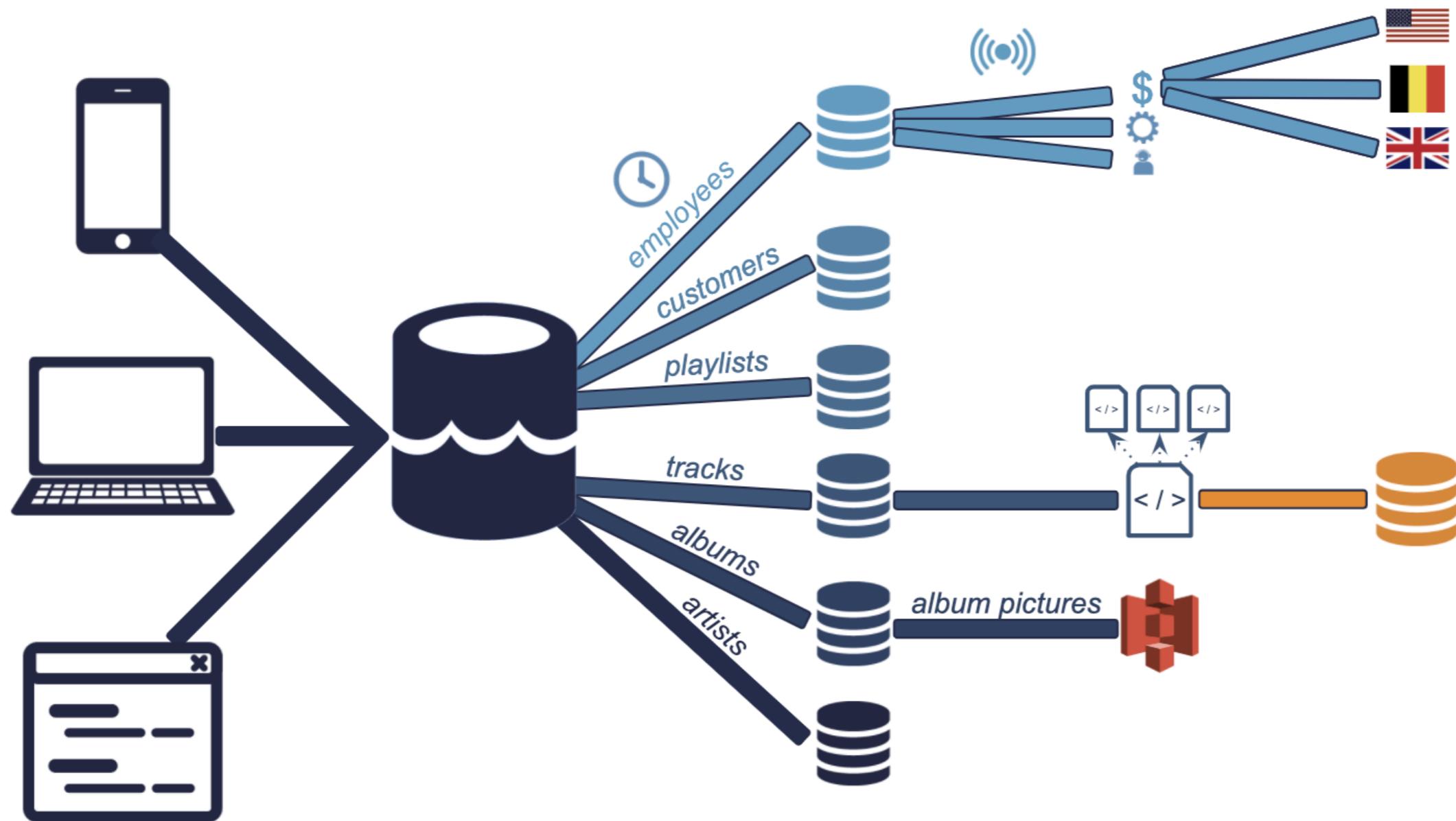
Azure
SQL Database

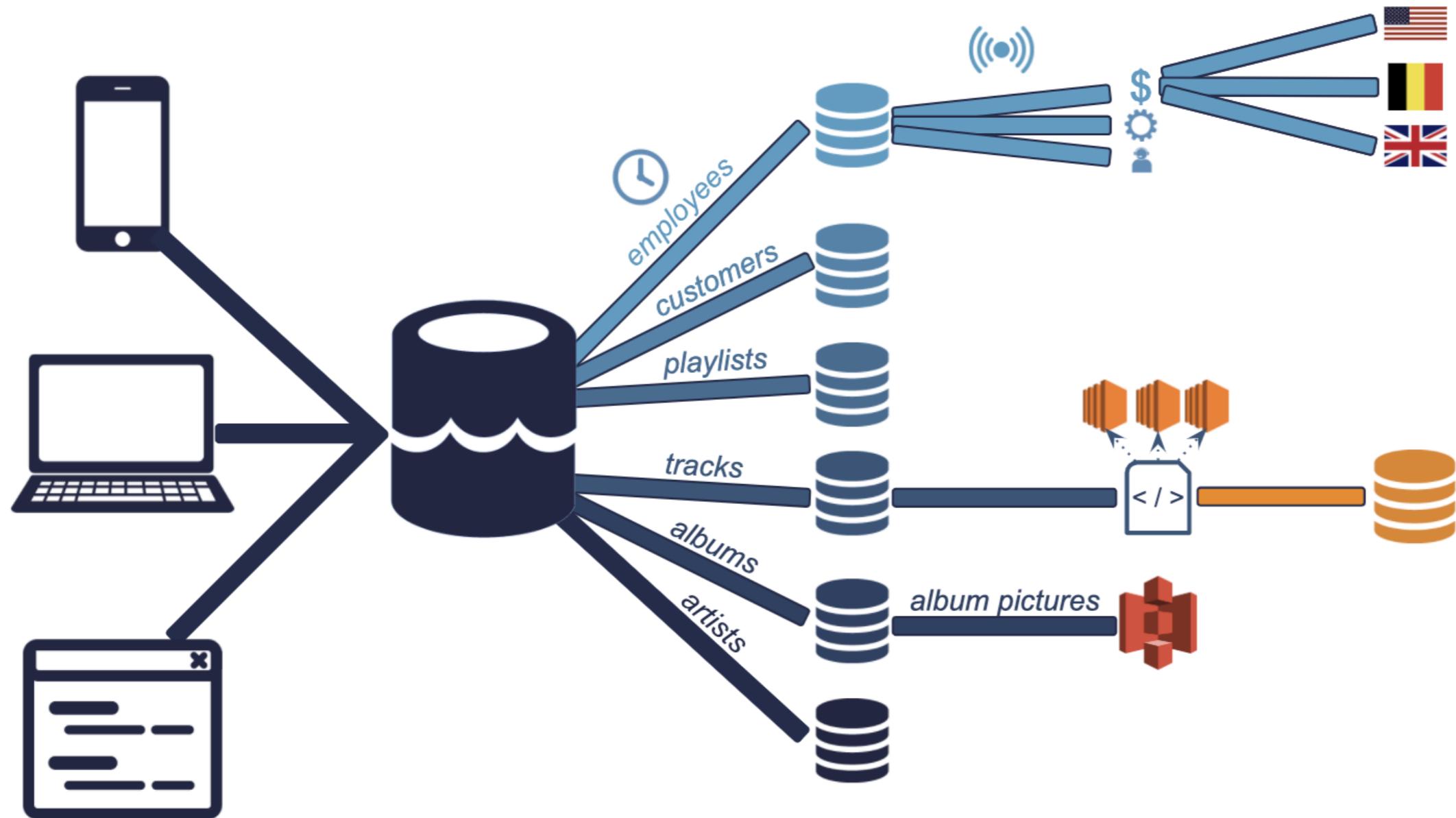


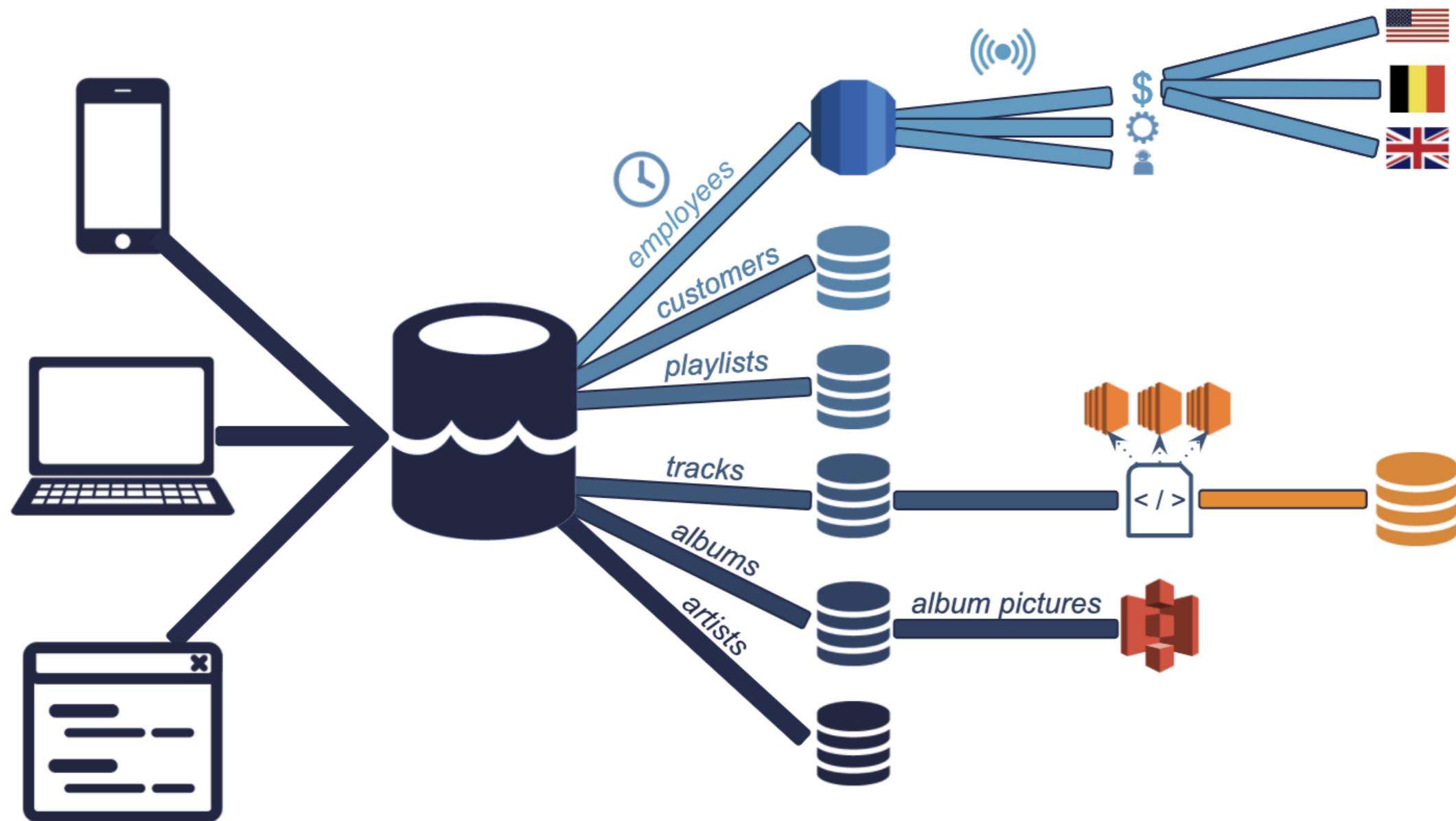
Google
Cloud SQL











Multicloud

Pros

- Reducing reliance on a single vendor
- Cost-efficiencies
- Local laws requiring certain data to be physically present within the country
- Mitigating against disasters

Cons

- Cloud providers try to lock in consumers
- Incompatibility
- Security and governance

Summary

- Benefits and risks of cloud computing
- How it is implemented at Spotflix
- Can cite the main cloud providers and their services

Let's practice!

UNDERSTANDING DATA ENGINEERING

We are the champions

UNDERSTANDING DATA ENGINEERING



Actually, YOU are the champion!



What you learned - chapter 1

- What Data Engineering is
- How important it is
- How data engineers differ from data scientists
- What a data pipeline is and how it works

What you learned - chapter 2

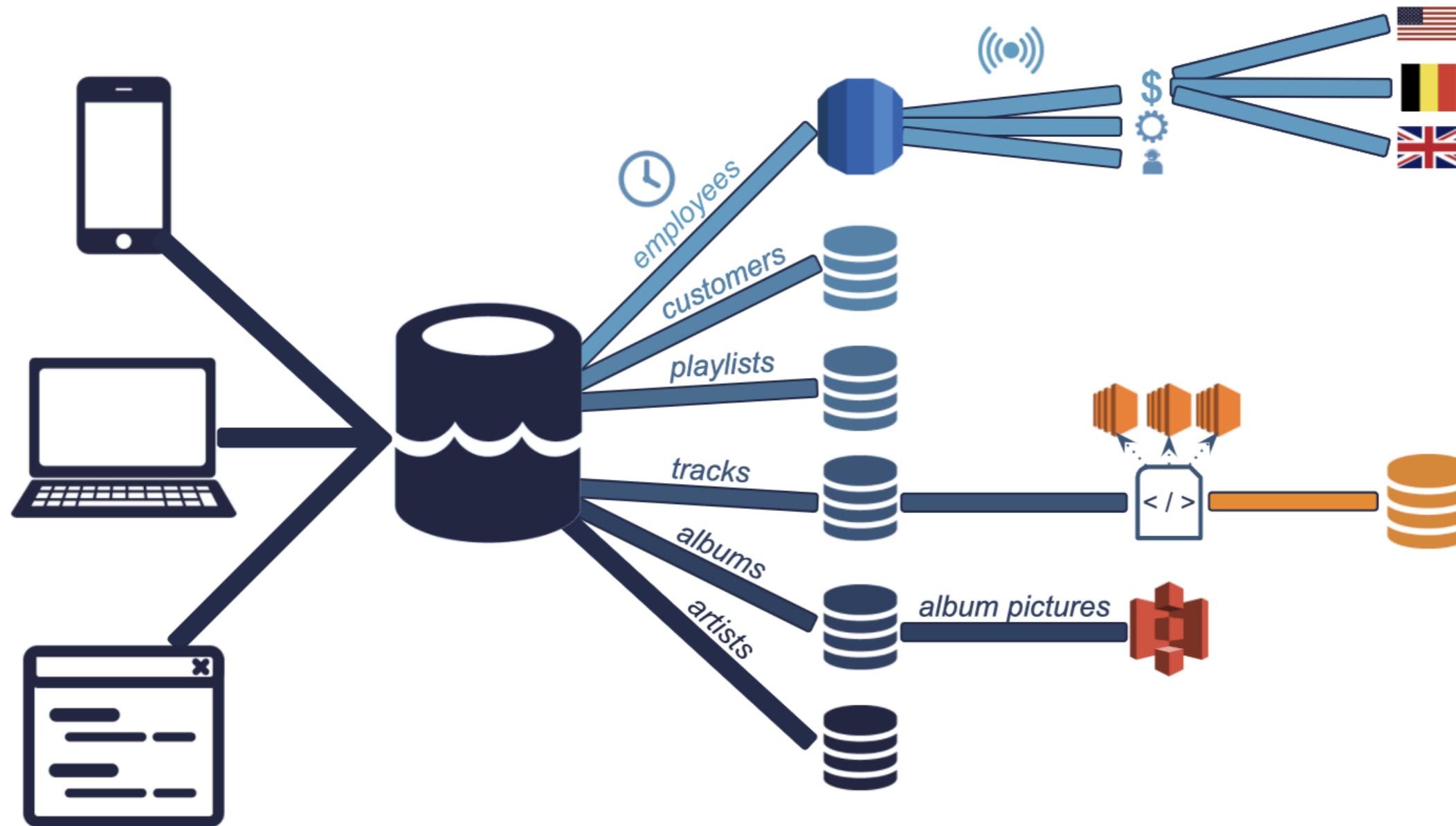
- The different structures data can take
- How fundamental SQL is
- The differences between data lakes, data warehouses and databases

What you learned - chapter 3

- How data is processed
- How scheduling holds it all together
- Parallel computing
- Cloud computing

And some more

- What SQL code actually looks like
- Main tools and technologies used in data engineering
- And some more



Data Engineering for Everyone - Lexicon

Data Engineering for Everyone - Lexicon

- **Airflow**: an open-source workflow management platform used to schedule data engineering tasks.
Started at Airbnb, now maintained by the Apache foundation.
- **AWS**: Amazon Web Services. Amazon's cloud computing services.
- **Azure**: Microsoft's cloud services.
- **Big data**: the systematic storage, management and analysis of datasets that are too large or complex to be dealt with by traditional data-processing application software. Big Data revolves around 4 Vs: volume, variety, velocity, and veracity.
- **Cloud computing**: the use of a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer.

A promise is a promise, DataChamps!

- All the exercises are song titles
- Search for "DataChamps" on Spotify

Congratulations!

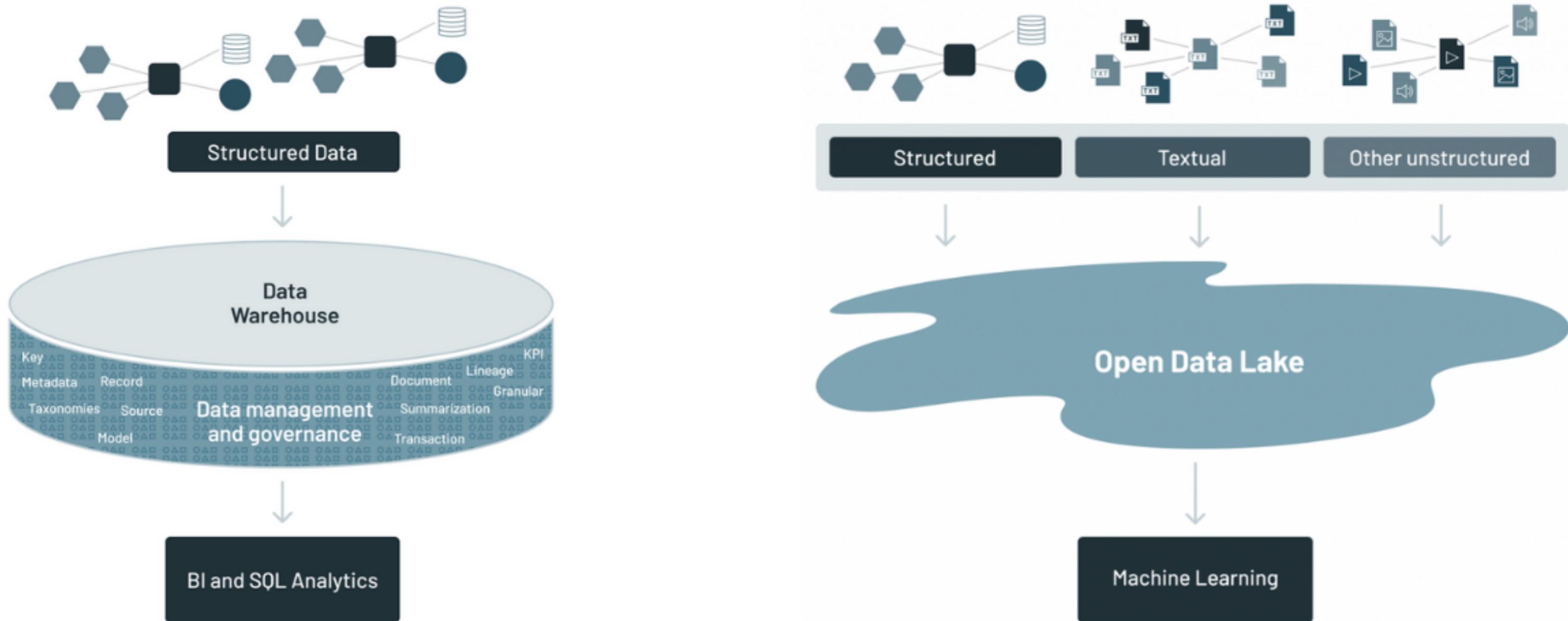
UNDERSTANDING DATA ENGINEERING

The Databricks Data Intelligence Platform

INTRODUCTION TO DATABRICKS

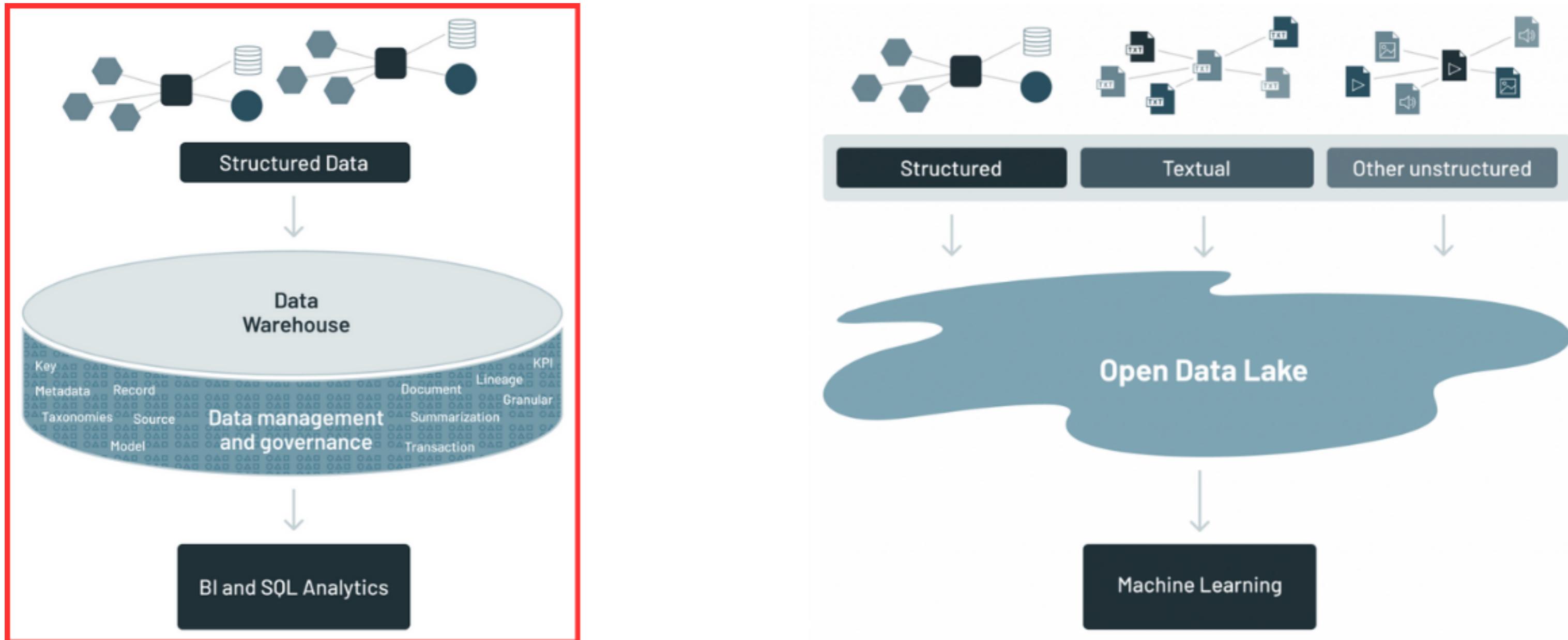


Architecture Options



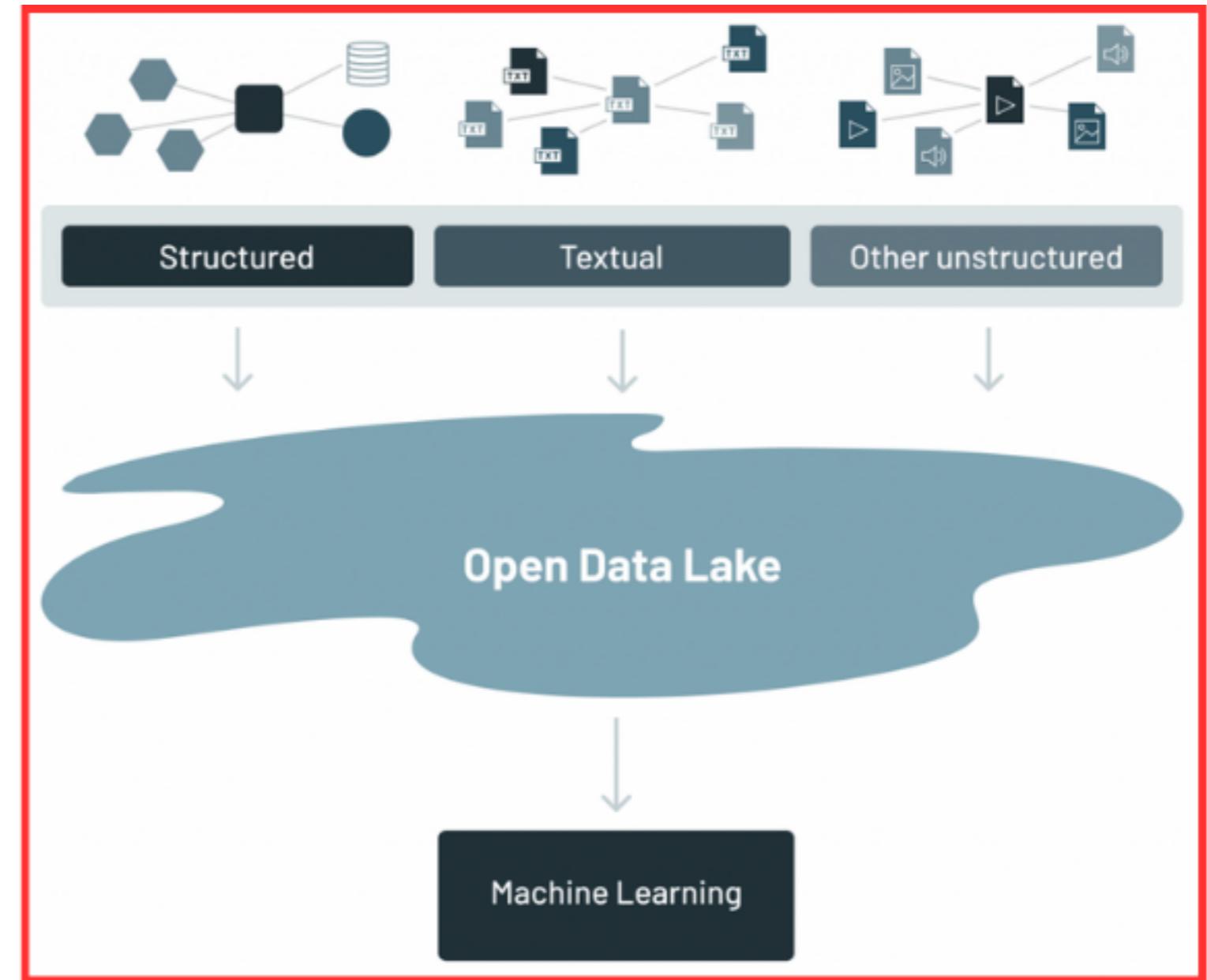
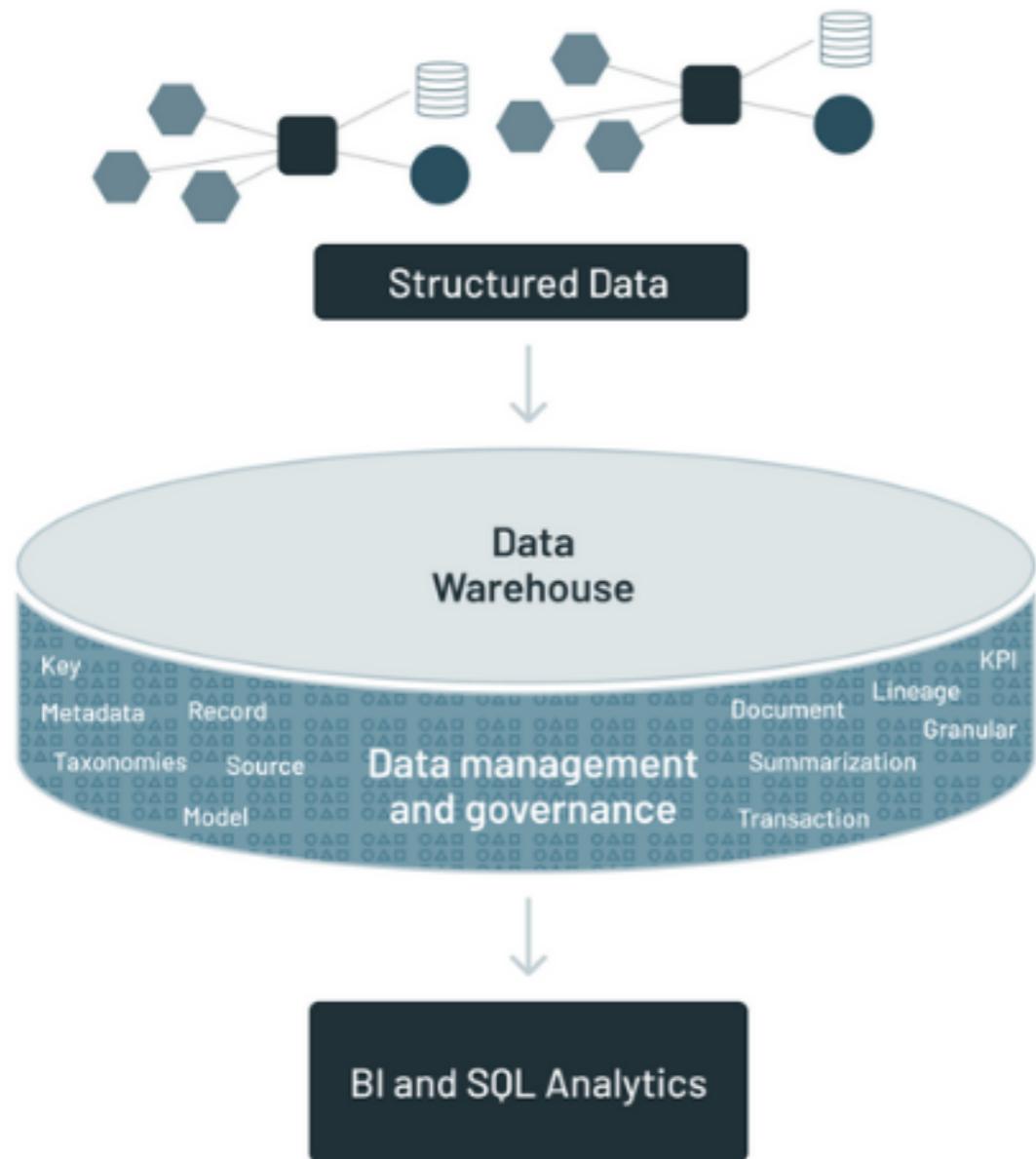
¹ <https://www.databricks.com/blog/2021/05/19/evolution-to-the-data-lakehouse.html>

Birth of the Lakehouse



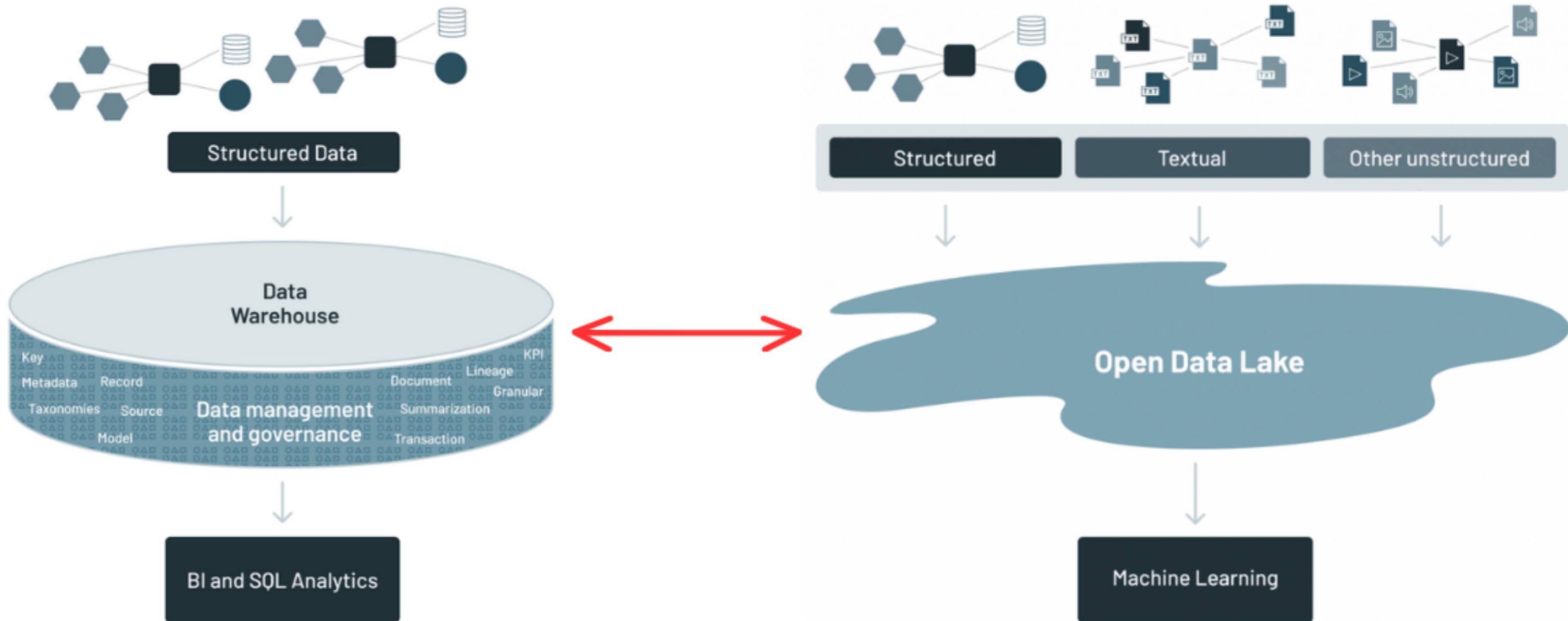
¹ <https://www.databricks.com/blog/2021/05/19/evolution-to-the-data-lakehouse.html>

Birth of the Lakehouse



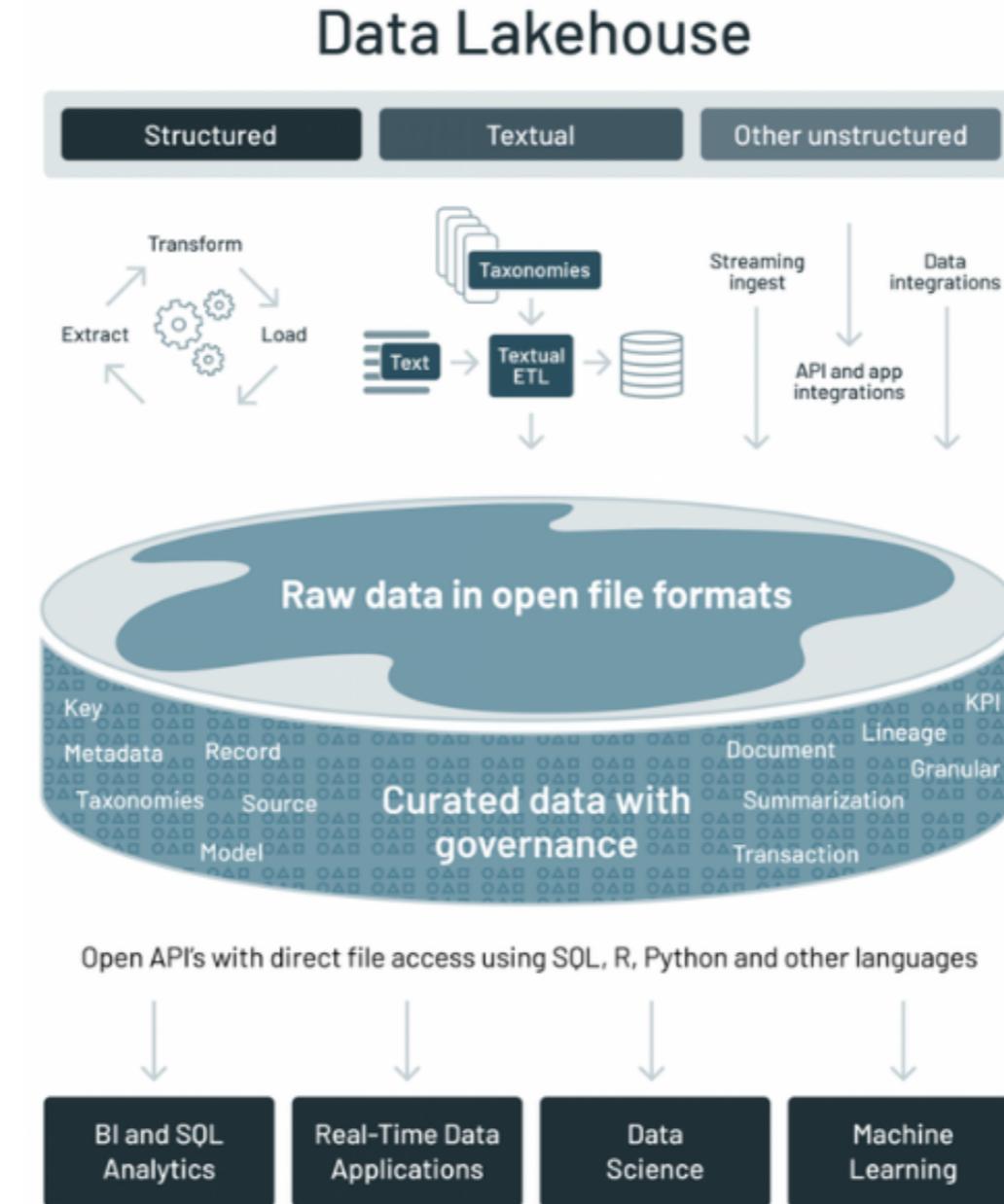
¹ <https://www.databricks.com/blog/2021/05/19/evolution-to-the-data-lakehouse.html>

Birth of the Lakehouse



¹ <https://www.databricks.com/blog/2021/05/19/evolution-to-the-data-lakehouse.html>

Birth of the Lakehouse

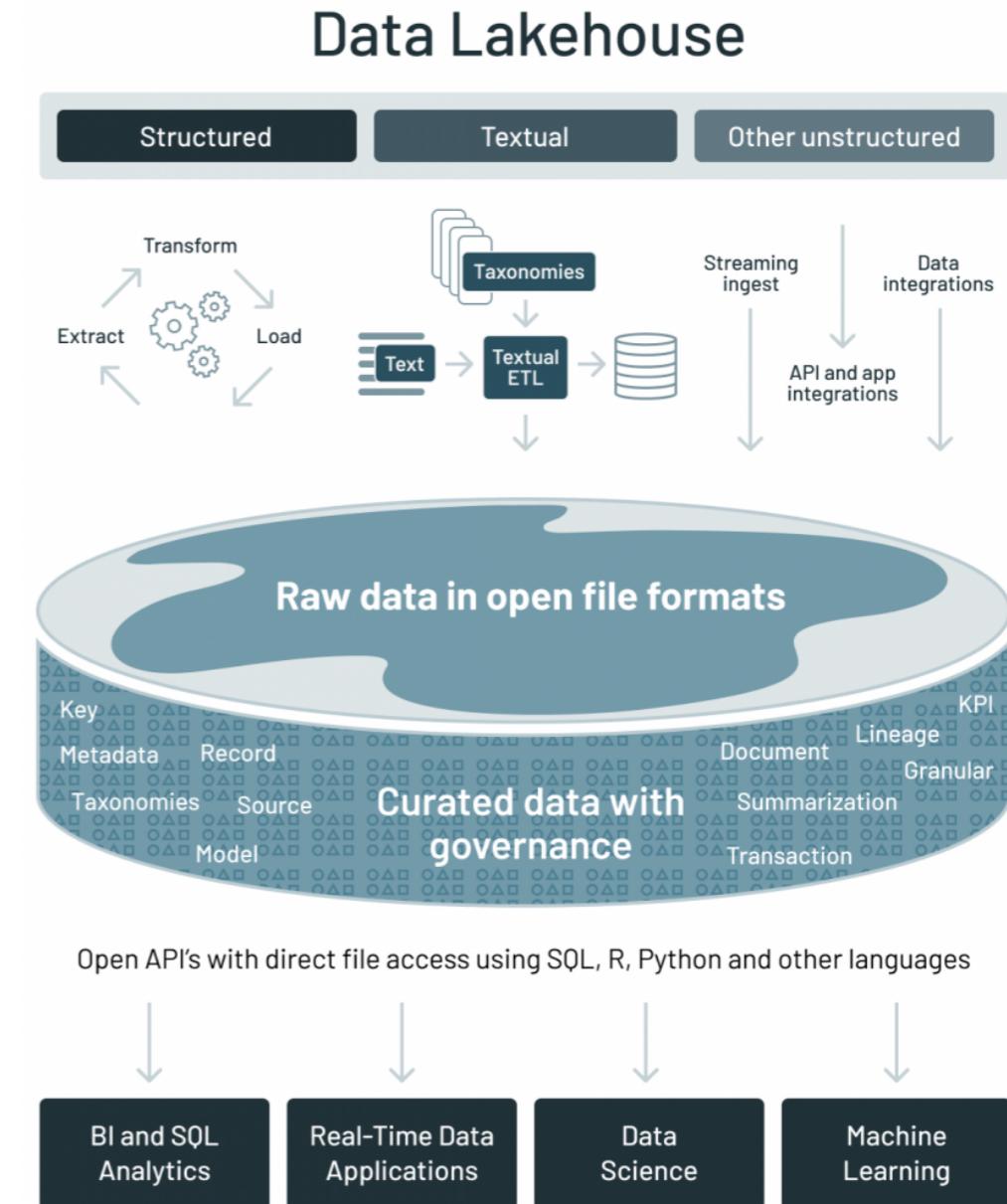


¹ <https://www.databricks.com/blog/2021/05/19/evolution-to-the-data-lakehouse.html>

The Databricks Lakehouse

The Databricks Lakehouse Platform

- Single platform for all data workloads
- Simplified architecture
- Collaborative environment



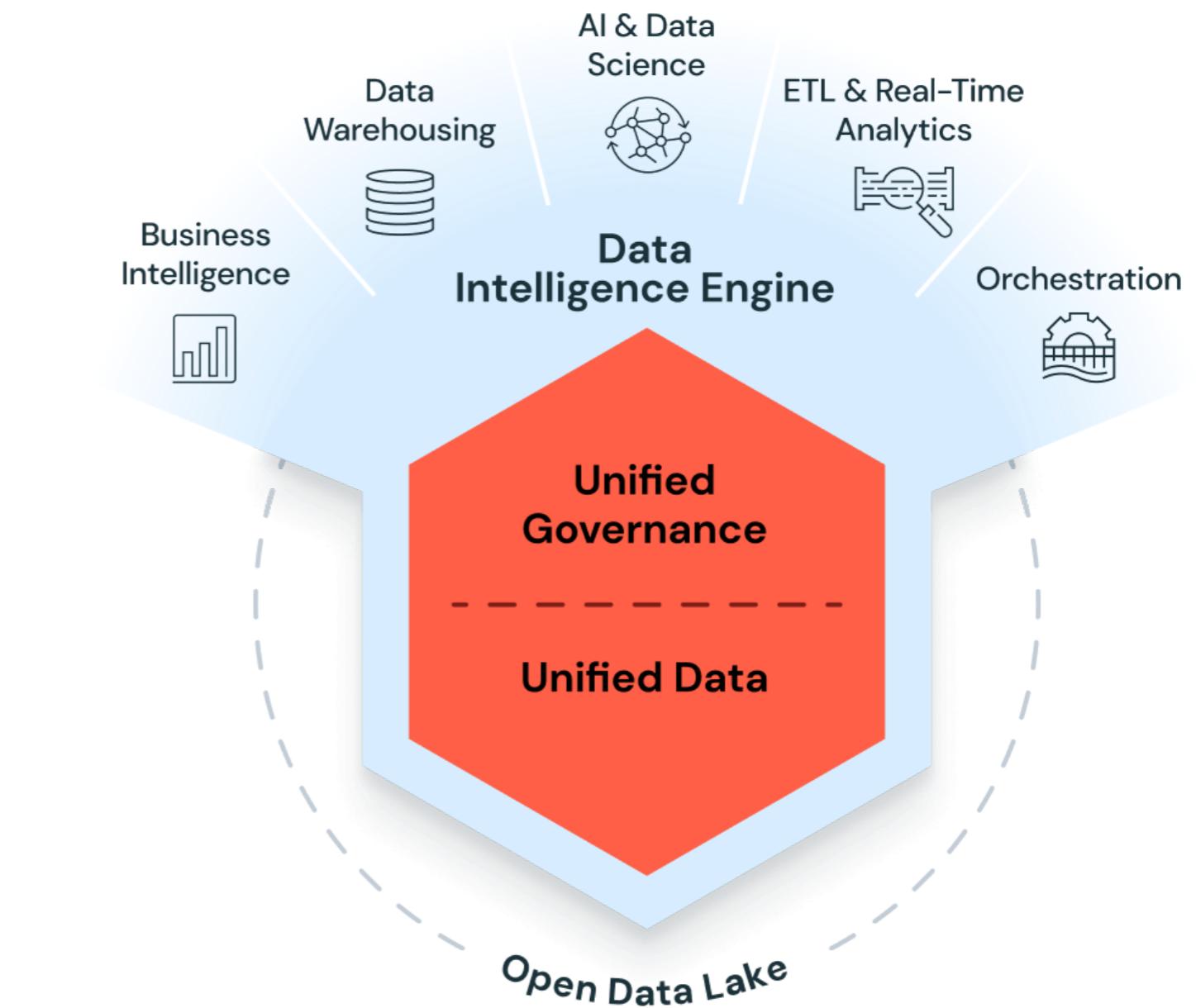
¹ <https://www.databricks.com/blog/2021/05/19/evolution-to-the-data-lakehouse.html>

The Databricks Data Intelligence Platform

Evolution from the Lakehouse

Databricks has expanded the Lakehouse vision to create the first *Data Intelligence Platform*

- Same core architecture of the Lakehouse
- Built-in AI
- First-class support for custom AI applications



¹ <https://www.databricks.com/product/data-intelligence-platform>

Databricks Architecture Benefits

Unification

- Every use case from AI to BI
- Benefits of data warehouse and data lake



Multi-Cloud

- Bring powerful platform to your data
- No lock-in to a specific cloud platform



Databricks Development Benefits

Collaborative

- Every data persona
- Ability to work in same platform in real-time



Open-Source

- Underpinned by *Apache Spark*
- Support for most popular languages (Python, R, Scala, SQL)

A blurred screenshot of a computer monitor displaying a large amount of code. The code is written in a programming language, likely Python, and includes several functions and variables. The text is in a monospaced font with color-coded syntax highlighting for different elements like keywords, comments, and strings.

Let's practice!

INTRODUCTION TO DATABRICKS

Setting up a Databricks workspace example

INTRODUCTION TO DATABRICKS



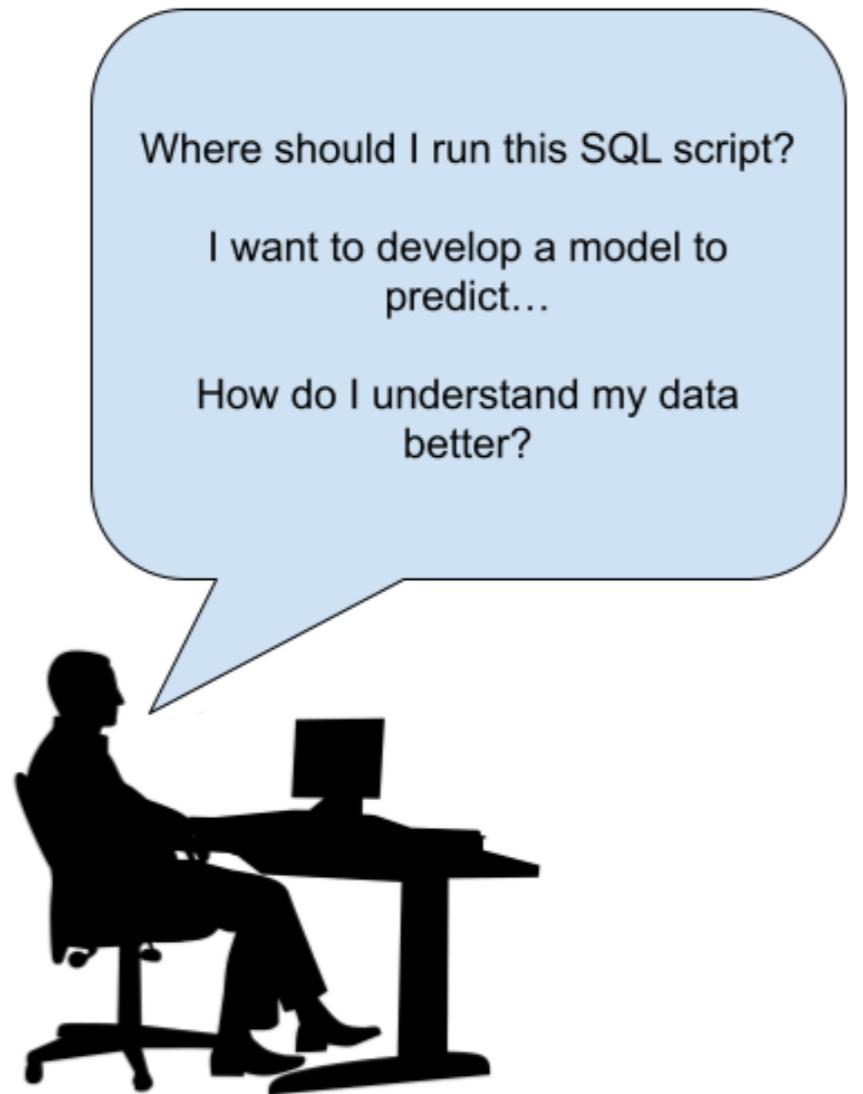
Let's practice!

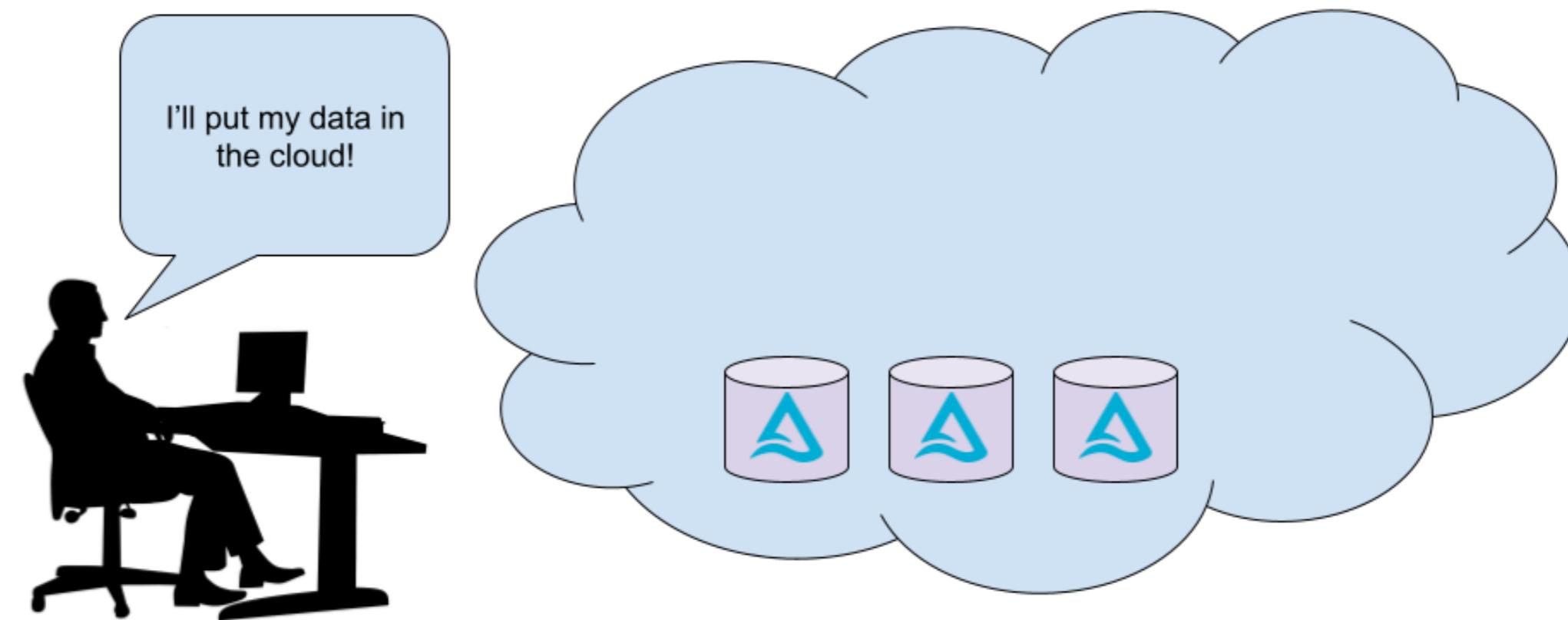
INTRODUCTION TO DATABRICKS

Databricks Architecture

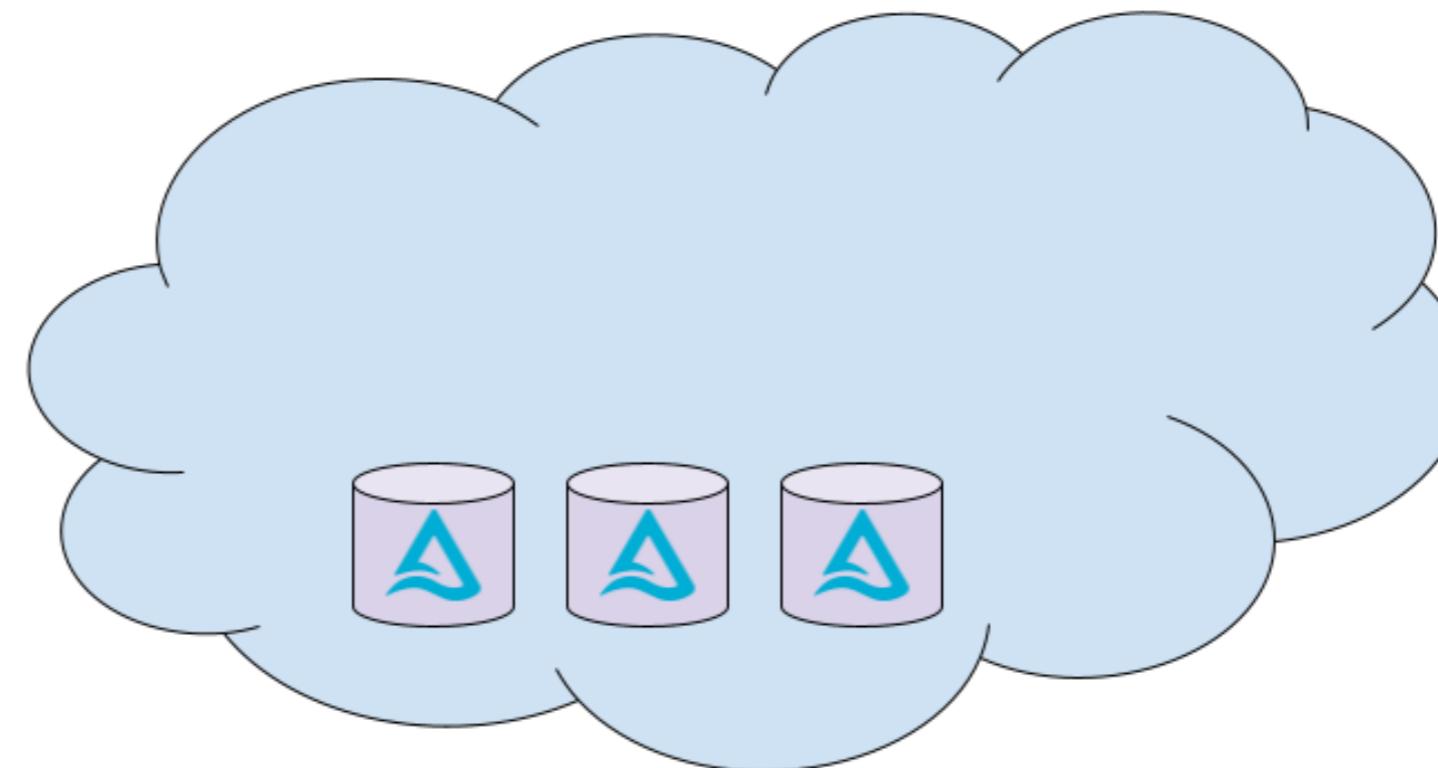
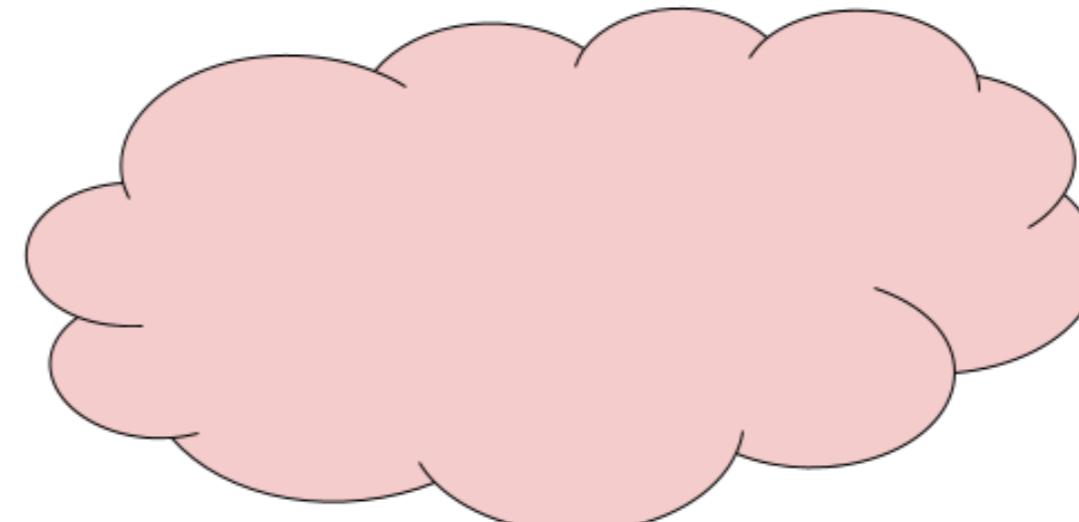
INTRODUCTION TO DATABRICKS





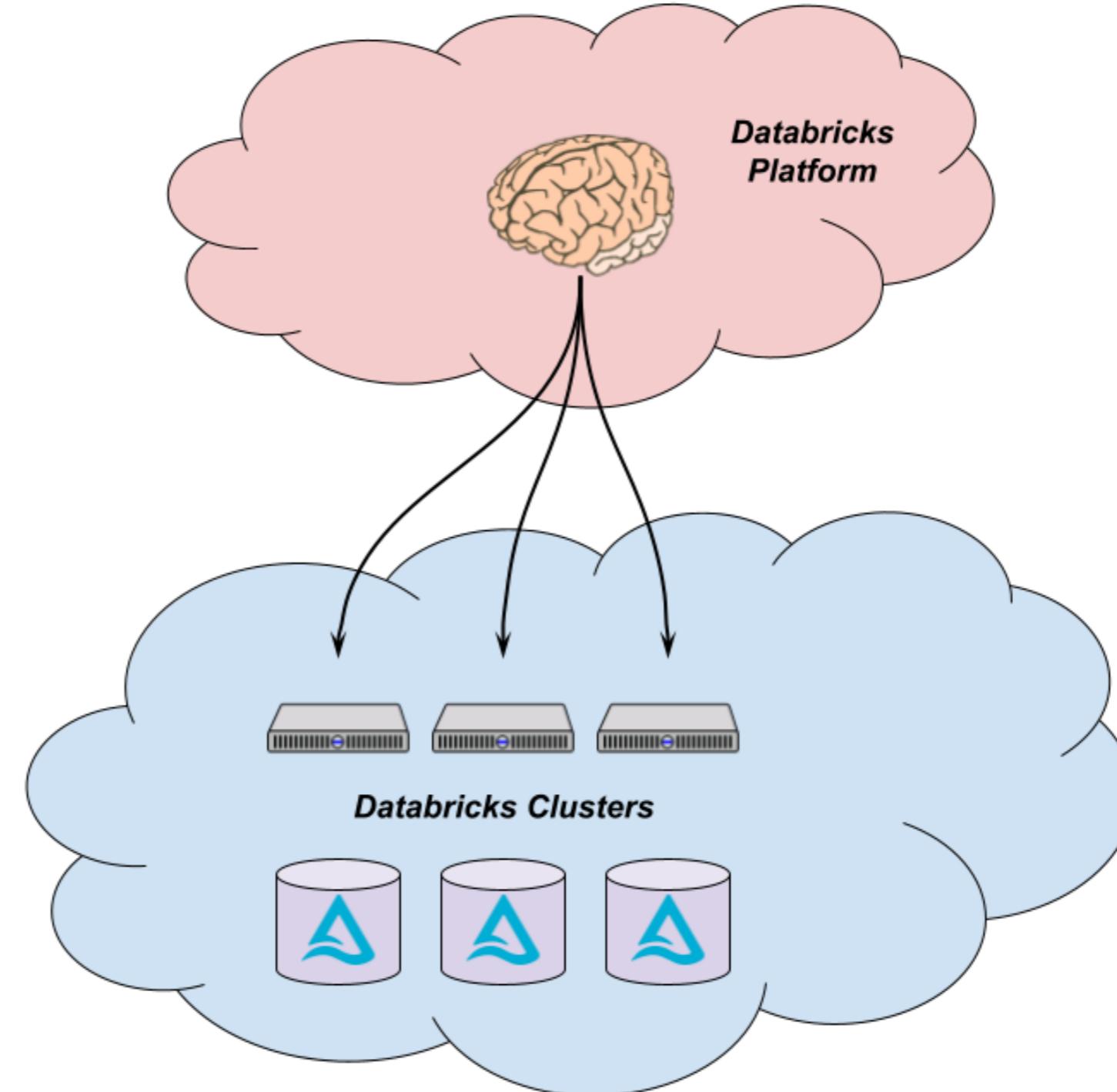


INTRODUCTION TO DATABRICKS





databricks



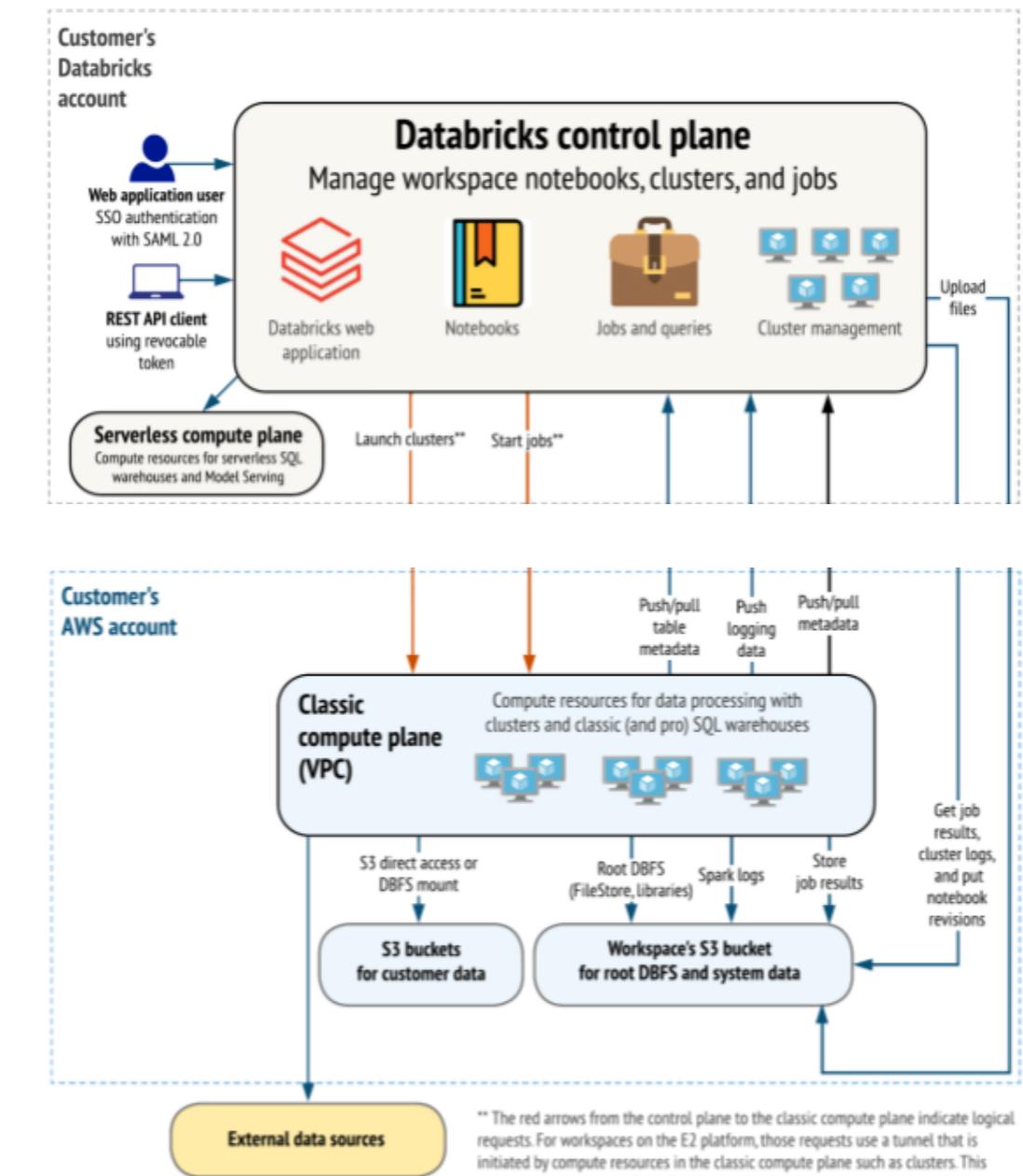
The Databricks Architecture

Control Plane

- Databricks owned environment in your cloud / region
- Hosts the UI, notebooks, and general code
- Orchestrates compute nodes for processing

Compute Plane

- Customer owned environment
- Location for data storage
- Customer networking, applications, etc.



¹ <https://docs.databricks.com/en/getting-started/overview.html#high-level-architecture>

Let's review!

INTRODUCTION TO DATABRICKS

Administering a Databricks workspace

INTRODUCTION TO DATABRICKS



Account Administrators

Key Responsibilities:

- Creating and managing workspaces
- Governing access to workspaces
- Managing the account subscription



Account Console



Account console

Manage your Databricks account at scale



Workspaces

Configure workspace settings. Workspaces contain notebooks, libraries, queries, and workflows



Data

Manage metastores as your top-level container for data, catalogs, schemas (also called databases), views and tables



Users & groups

Manage identities for use with jobs, automated tools and systems



Settings

Configure your Databricks account user provisioning and other settings

Workspaces

The screenshot shows the Databricks Account console interface. On the left is a dark sidebar with four icons: a cluster, a user, a group, and a gear. The main area has a header "Account console" and a subtitle "Manage your Databricks account at scale". Below are four cards:

- Workspaces** (highlighted with a red border): Configure workspace settings. Workspaces contain notebooks, libraries, queries, and workflows.
- Data**: Manage metastores as your top-level container for data, catalogs, schemas (also called databases), views and tables.
- Users & groups**: Manage identities for use with jobs, automated tools and systems.
- Settings**: Configure your Databricks account user provisioning and other settings.

<https://accounts.cloud.databricks.com/>

Data

The screenshot shows the Databricks Account console interface. On the left is a vertical sidebar with four icons: a network icon, a cluster icon, a gear icon, and a settings gear icon. The main area has a header "Account console" and a subtitle "Manage your Databricks account at scale". Below this are four cards:

- Workspaces**: Manage workspace settings. Workspaces contain notebooks, libraries, queries, and workflows.
- Data**: Manage metastores as your top-level container for data, catalogs, schemas (also called databases), views and tables. This card is highlighted with a red border.
- Users & groups**: Manage identities for use with jobs, automated tools and systems.
- Settings**: Configure your Databricks account user provisioning and other settings.

<https://accounts.cloud.databricks.com/>

Users & Groups

The screenshot shows the Databricks Account console interface. On the left is a vertical sidebar with icons for Workspaces, Data, Settings, and Help. The main area has four cards:

- Workspaces**: Manage workspace settings. Workspaces contain notebooks, libraries, queries, and workflows.
- Data**: Manage metastores as your top-level container for data, catalogs, schemas (also called databases), views and tables.
- Users & groups**: Manage identities for use with jobs, automated tools and systems. This card is highlighted with a red border.
- Settings**: Configure your Databricks account user provisioning and other settings.

<https://accounts.cloud.databricks.com/>

Workspace Administrators

Key Responsibilities:

- Managing identities in your workspace
- Creating and managing compute resources

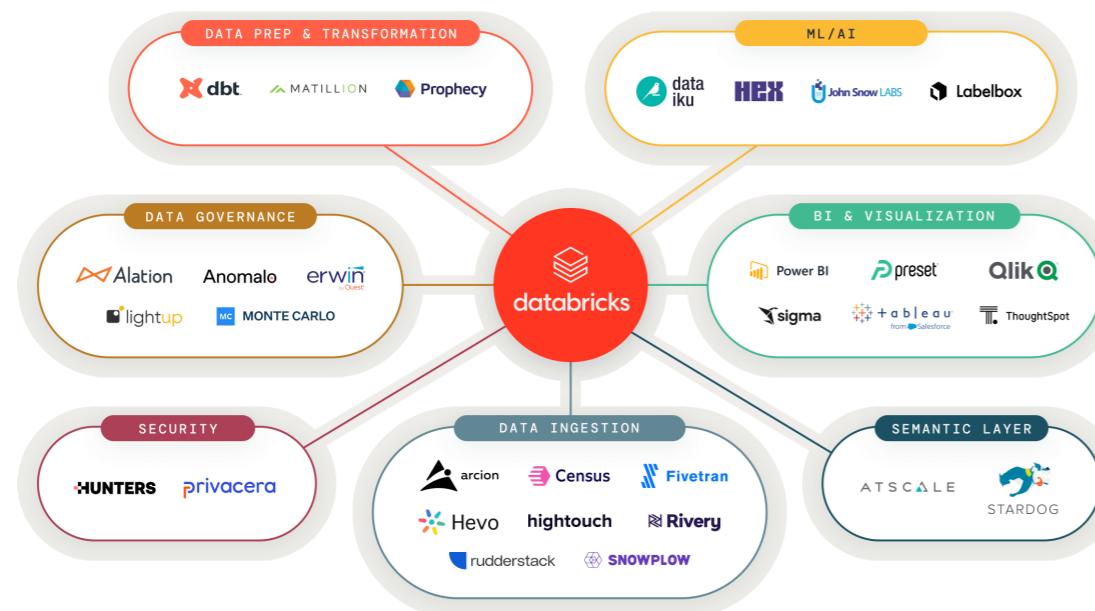
Admin Settings

[Users](#) [Service principals](#) [Groups](#) [Global init scripts](#) [Workspace settings](#) [SQL settings](#) [Notification destinations](#) [SQL warehouse settings](#)

Other Administrative Activities

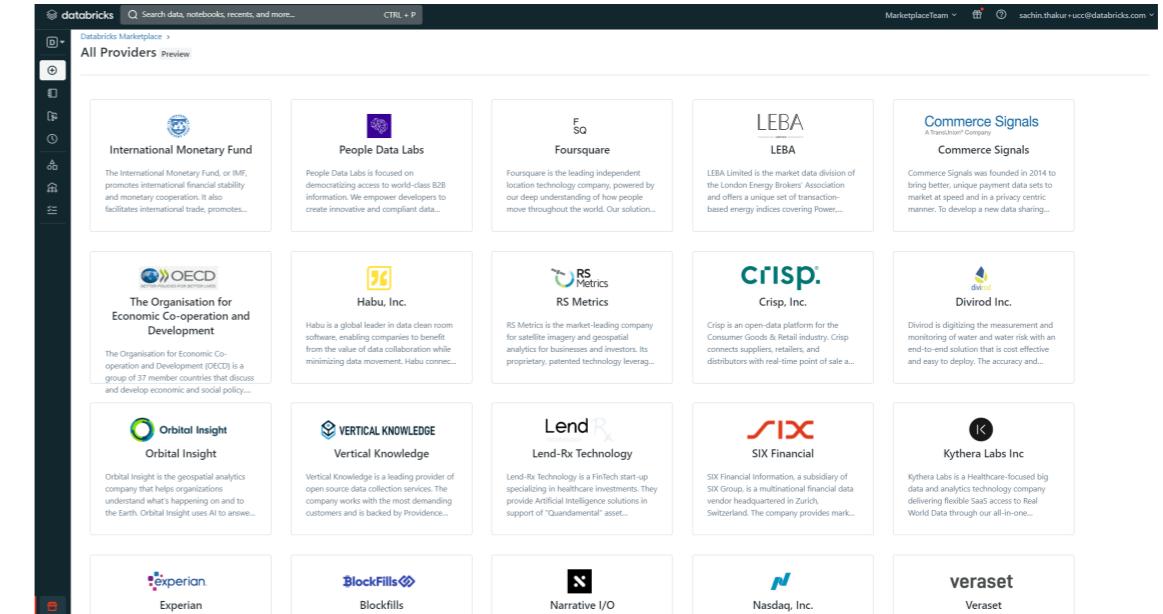
Partner Connect

- UI-based connection to partner technologies
- BI connections, ingestion tools, etc.



Databricks Marketplace

- Discover and access third-party datasets
- Integrate directly into your data catalogs



¹ <https://www.databricks.com/partnerconnect>

Let's review!

INTRODUCTION TO DATABRICKS

Data Intelligence Platform - Data

INTRODUCTION TO DATABRICKS



Why do organizations care about data management?

Protection and security



Confidence in data



Kinds of data

Structured

- Most common and understood
- Typical rows and columns
- Examples:
 - database tables
 - .CSV
 - Parquet
 - Delta

id	name	occupation	location
1	Kevin	Data Scientist	California
2	Tom	Architect	Arizona
3	Sally	Lawyer	Texas
4	Tina	Surgeon	Florida
5	Joe	Engineer	New York

Kinds of data

Semi-structured

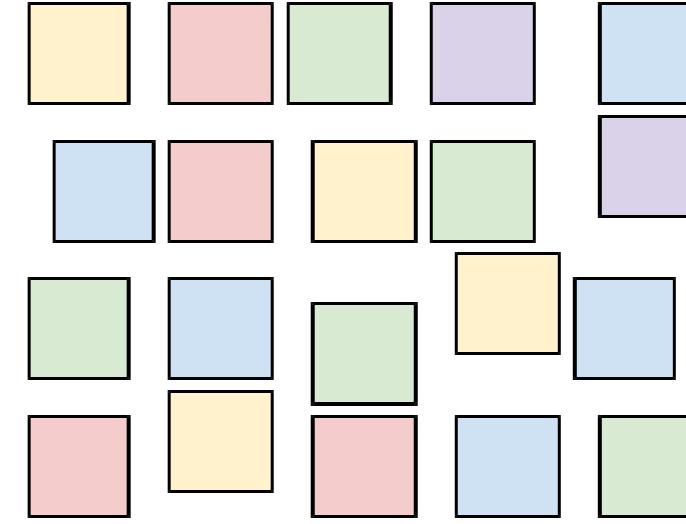
- Common with web-based devices
- Some structure, but more flexible in content
- Examples:
 - JSON
 - XML
 - HTML

```
{  
  "people": [ {  
    "id": 1,  
    "name": "Kevin",  
    "occupation": "Data Scientist",  
    "location": "California"},  
    {  
      "id": 2,  
      "name": "Tom",  
      "occupation": "Architect",  
      "location": "Arizona"}]  
}
```

Kinds of data

Unstructured

- Common with smart devices, cameras, etc.
- Little structure, information-rich
- Examples:
 - JPEG
 - PNG
 - MP4
 - PDF
 - DOC



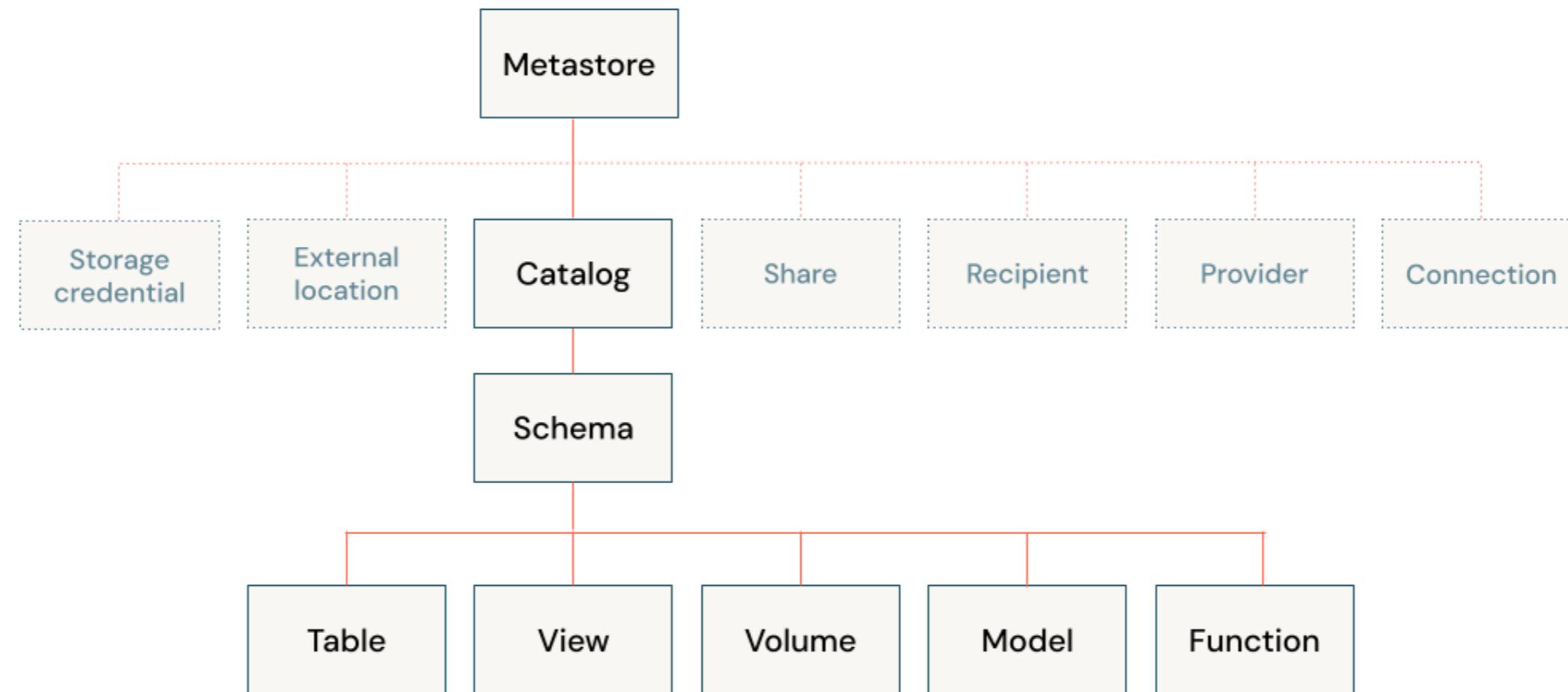
Delta

delta.io

- Open-source storage format
- Collection of parquet tables
- JSON transaction log
- Fully ACID compliant
- Batch and streaming datasets

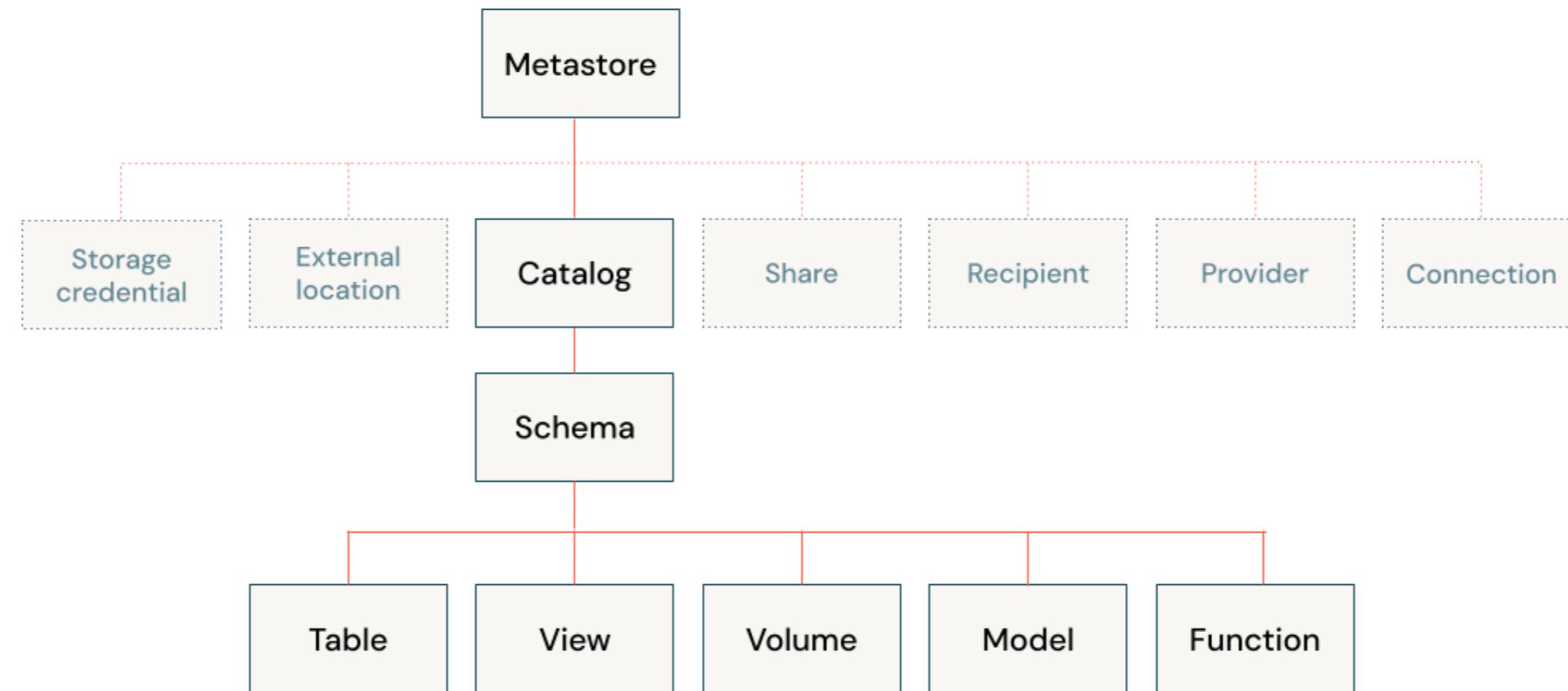


Unity Catalog



¹ <https://docs.databricks.com/en/data-governance/unity-catalog/index.html#the-unity-catalog-object-model>

Unity Catalog



GRANT, SHOW, REVOKE, USE ...

Catalog Explorer

- Single location to explore all data assets
- UI to discover data
- Manage Unity Catalog permissions
- View data lineage and related assets

The screenshot shows the Databricks Catalog Explorer interface. On the left, there is a sidebar titled "Catalog" with a "Type to filter" input field and a dropdown menu. The sidebar lists several catalogs: "databricks_ws_094b2e73_d2f4_4e66_9dd7_4e7a89942f2f" (expanded), "default", "information_schema", "hive_metastore" (expanded), "samples" (expanded), "default", "nyctaxi", "tpch", "system" (expanded), and "information_schema". On the right, the main area is titled "Catalogs" and shows a table of "4 catalogs". The table has columns for "Name", "Owner", and "Created at". The data is as follows:

Name	Owner	Created at
databricks_ws_094b2e73_d2f4_4e66_9dd7_4e7a89942f2f	_workspace_admins_databricks_ws_094b2e73_...	2024-03-21 15:40:44
hive_metastore		
samples		
system	System user	2024-02-01 01:25:41

Let's practice!

INTRODUCTION TO DATABRICKS

Managing Data Catalogs

INTRODUCTION TO DATABRICKS



Let's practice!

INTRODUCTION TO DATABRICKS

Data Intelligence Platform - Compute

INTRODUCTION TO DATABRICKS



Why do organizations care about compute?



Apache Spark

- Created by Databricks co-founders
- Open source framework
- Highly efficient distributed computing
- APIs for Python, SQL, Scala, R
- Great for all use cases:
 - data engineering to machine learning and business intelligence

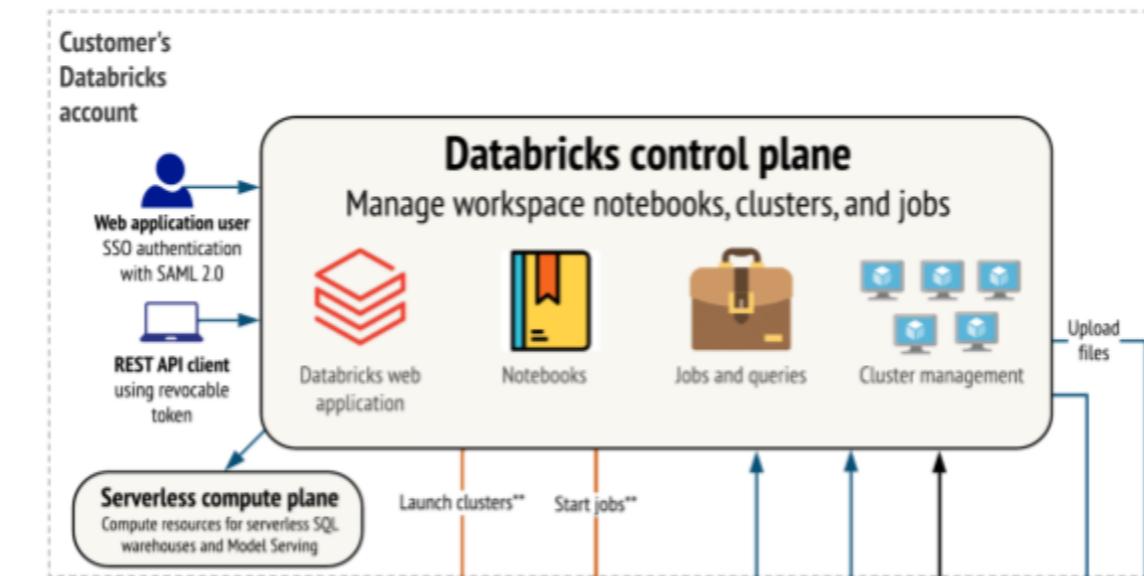
Check out some of the [Apache Spark courses](#) on DataCamp!



Cluster Types

Classic

- Compute resources (virtual machines) are created in the Compute Plane
- Databricks provides configuration to your cloud
- *Pros:* compute and security in your environment, leverage pre-existing compute pools, etc.
- *Cons:* slow startup time

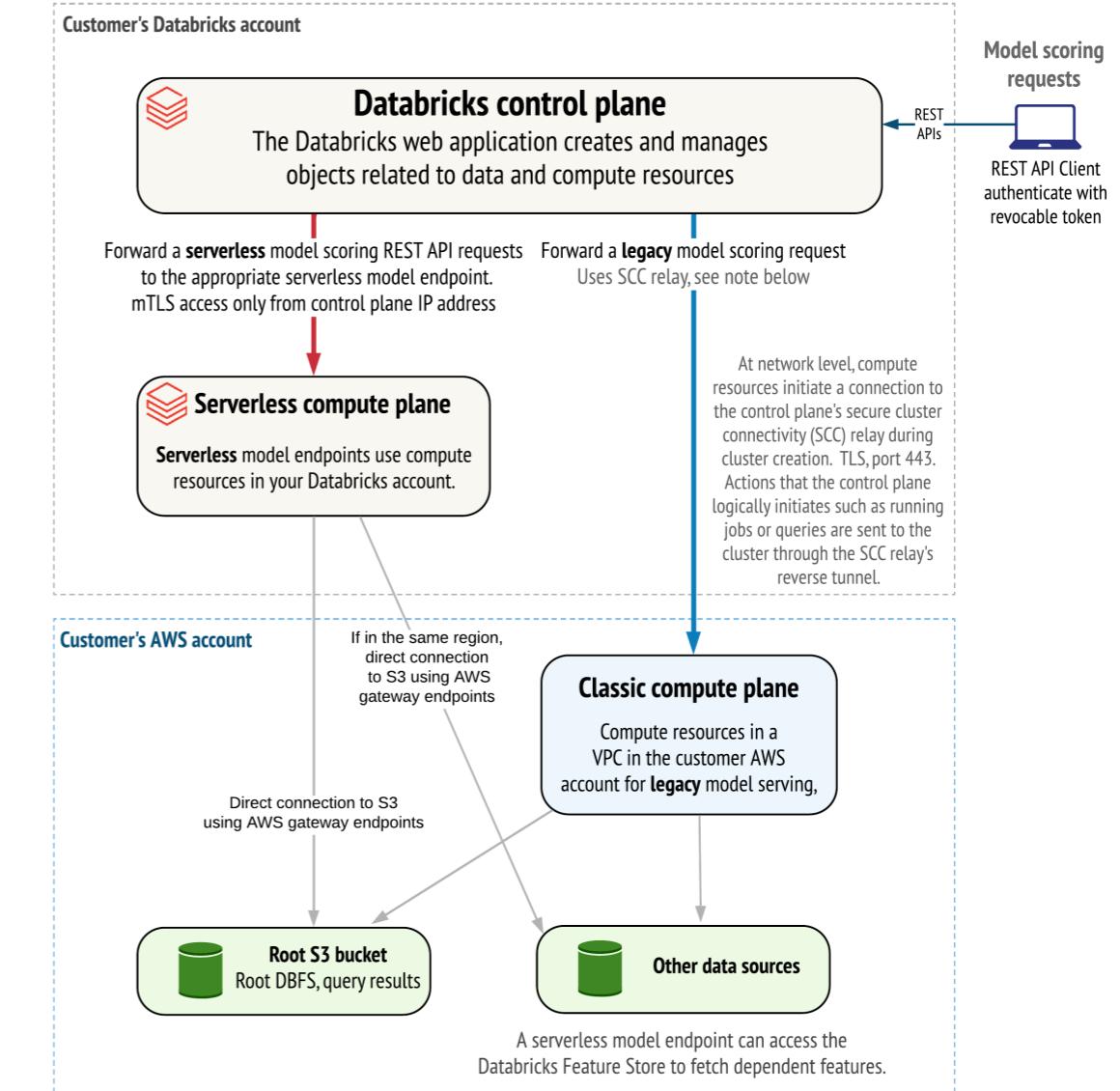


Cluster Types

Serverless

- Compute resources (virtual machines) are created in the Control Plane
- Databricks provides access to your users
- **Pros:** Fast startup time, the latest and greatest feature, the fastest performance, Databricks improves performance over time
- **Cons(?)**: compute not in your environment

Compare classic and serverless compute planes for Model Serving



Single-node vs. Multi-node

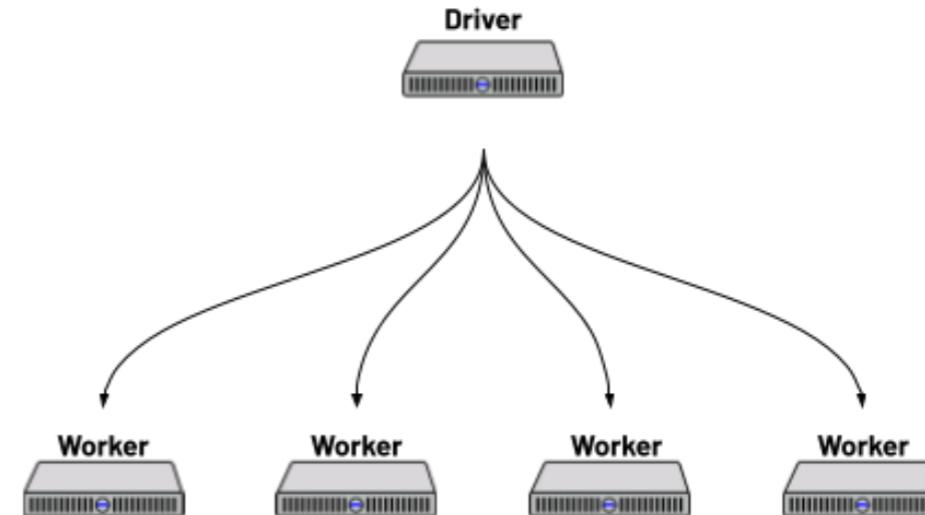
Single-node

- Cluster with just a Driver Node
- Can still run Spark
- Can also run single-node frameworks (i.e., pandas)
- Great for smaller datasets



Multi-node

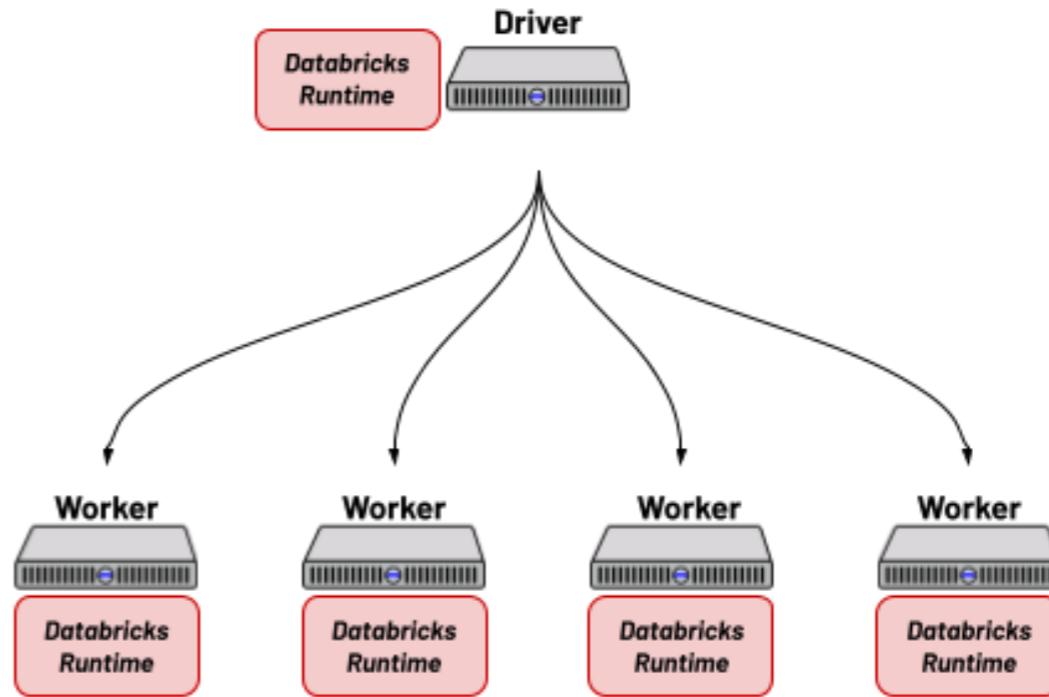
- Cluster with a Driver Node and one or more Worker Nodes
- Spark can distribute work across multiple nodes
- Great for larger datasets



Databricks Runtime

- Installed on every Databricks cluster
 - Optimized version of Apache Spark
 - Photon for faster SQL queries
 - Common libraries (e.g., pandas, dplyr, sci-kit learn)
 - Logic to connect with Databricks services

General recommendation: Use the most recent Long Term Support (LTS) version of the Runtime



Let's practice!

INTRODUCTION TO DATABRICKS

Data Intelligence Platform - Analytics

INTRODUCTION TO DATABRICKS



Why do organizations care about analytics?



Supported Languages

Scala

- Based on Java
- Generally used for data engineering



Python

- Used for all use cases



SQL

- Used for data engineering and BI

R

- Used for data science use cases

Databricks Notebooks

Based on Jupyter notebooks, Databricks Notebooks are an optimized and enhanced version.

The screenshot shows a Databricks Notebook interface. On the left is a sidebar with various icons for data management. The main area has a header "Exploratory Analysis" and a tab "test3 Python". A message bar at the top right says "Last edit was 10 minutes ago" and "Give feedback".

Code Editor (Cmd 9):

```
1 import pandas as pd; import numpy as np
2 # Step: Keep rows where entity is one of: United States
3 filtered = df.loc[df['entity'].isin(['United States'])]
4
5 # Step: Pivot dataframe from long to wide format using the variable column 'indicator' and the value column 'value'
6 wide_df = filtered.set_index(['entity', 'iso_code', 'date', 'indicator'])['value'].unstack(-1).reset_index()
7 wide_df.columns.name = ''
8
9 # Step: Replace missing values
10 wide_df = wide_df.fillna(0)
11
12 # Step: Change data type of date to Datetime
13 wide_df['date'] = pd.to_datetime(wide_df['date'], infer_datetime_format=True)
```

Comment Area:

isaac gritz 9/29/2022, 3:49:22 PM
This is really helpful, can we refactor this and share the code with the rest of the team?

rafi.kurlansik@datab... 9/29/2022, 3:52:44 PM
Sure thing, I'll create a Python module with these functions and work with Afsana on the unit tests.

Code Preview (Cmd 10):

Show code

Hospitalizations: 2020-2022

The chart displays the number of patients (Y-axis, 0 to 150k) against date (X-axis, Jan 2021 to Jul 2022). It includes six data series: Daily ICU occupancy (blue), Daily ICU occupancy per million (orange), Daily hospital occupancy (green), Daily hospital occupancy per million (purple), Weekly new hospital admissions (yellow), and Weekly new hospital admissions per million (cyan). The chart shows two major peaks in hospitalizations around January 2021 and January 2022.

Legend (Indicator):

- Daily ICU occupancy
- Daily ICU occupancy per million
- Daily hospital occupancy
- Daily hospital occupancy per million
- Weekly new hospital admissions
- Weekly new hospital admissions per million

SQL Editor

The screenshot shows the Databricks SQL Editor interface. On the left is a dark sidebar with various icons. The main area has a title bar with 'Catalog' and a search bar 'Type to filter'. Below this is a list of databases: 'For you' (hive_metastore, ml, samples, system) and 'All' (samples, tpch). The central part of the screen displays a query editor with the following code:

```
1 SELECT
2   o_orderdate AS Date,
3   o_orderpriority AS Priority,
4   sum(o_totalprice) AS `Total Price`
5 FROM
6   `samples`.`tpch`.`orders`
7 WHERE
8   o_orderdate > '1994-01-01'
9   AND o_orderdate < '1994-01-31'
10 GROUP BY
11   1,
12   2
13 ORDER BY
14   1,
15   2
```

Below the code is a results table titled 'Raw results' with columns: Date, Priority, Total Price. The data is as follows:

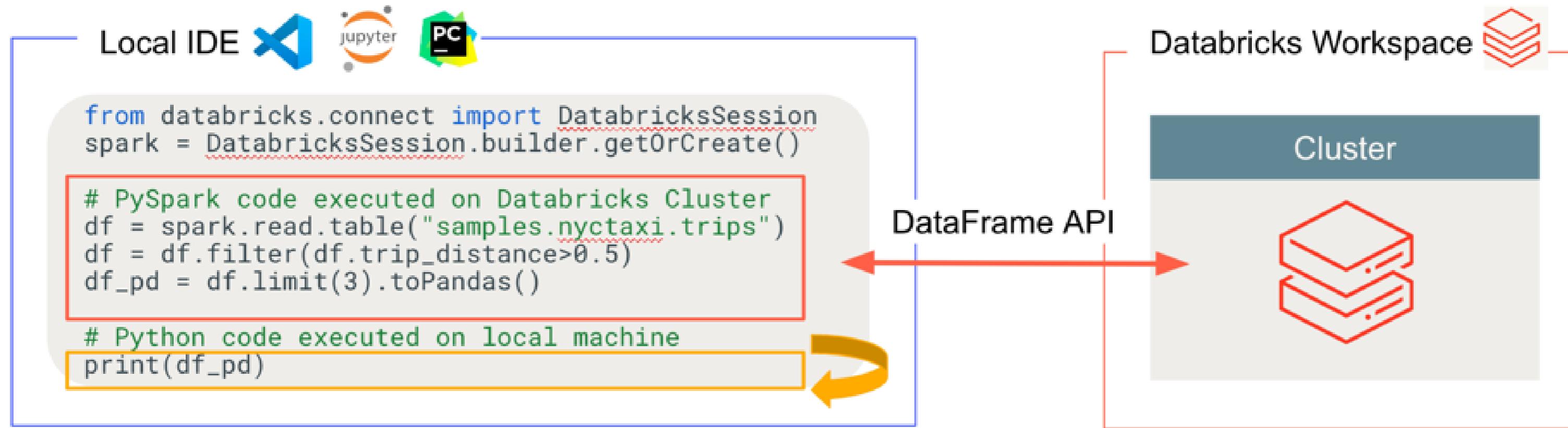
	Date	Priority	Total Price
1	1994-01-02	1-URGENT	96444609.82
2	1994-01-02	2-HIGH	93497904.94
3	1994-01-02	3-MEDIUM	88800085.02
4	1994-01-02	4-NOT SPECIFIED	97955477.98
5	1994-01-02	5-LOW	98015661.37
6	1994-01-03	1-URGENT	92534508.96
7	1994-01-03	2-HIGH	92286715.43
8	1994-01-03	3-MEDIUM	93521575.91
9	1994-01-03	4-NOT SPECIFIED	97569521.46

At the bottom, it says '12 s 751 ms | 145 rows returned' and 'Refreshed a minute ago'.

¹ <https://docs.databricks.com/en/sql/user/sql-editor/index.html>

Databricks Connect

Bring your own IDE! Leverage the power of Databricks Clusters while working in your favorite coding environment.



¹ <https://docs.databricks.com/en/dev-tools/databricks-connect/index.html>

Let's practice!

INTRODUCTION TO DATABRICKS

SQL in the Data Intelligence Platform

INTRODUCTION TO DATABRICKS

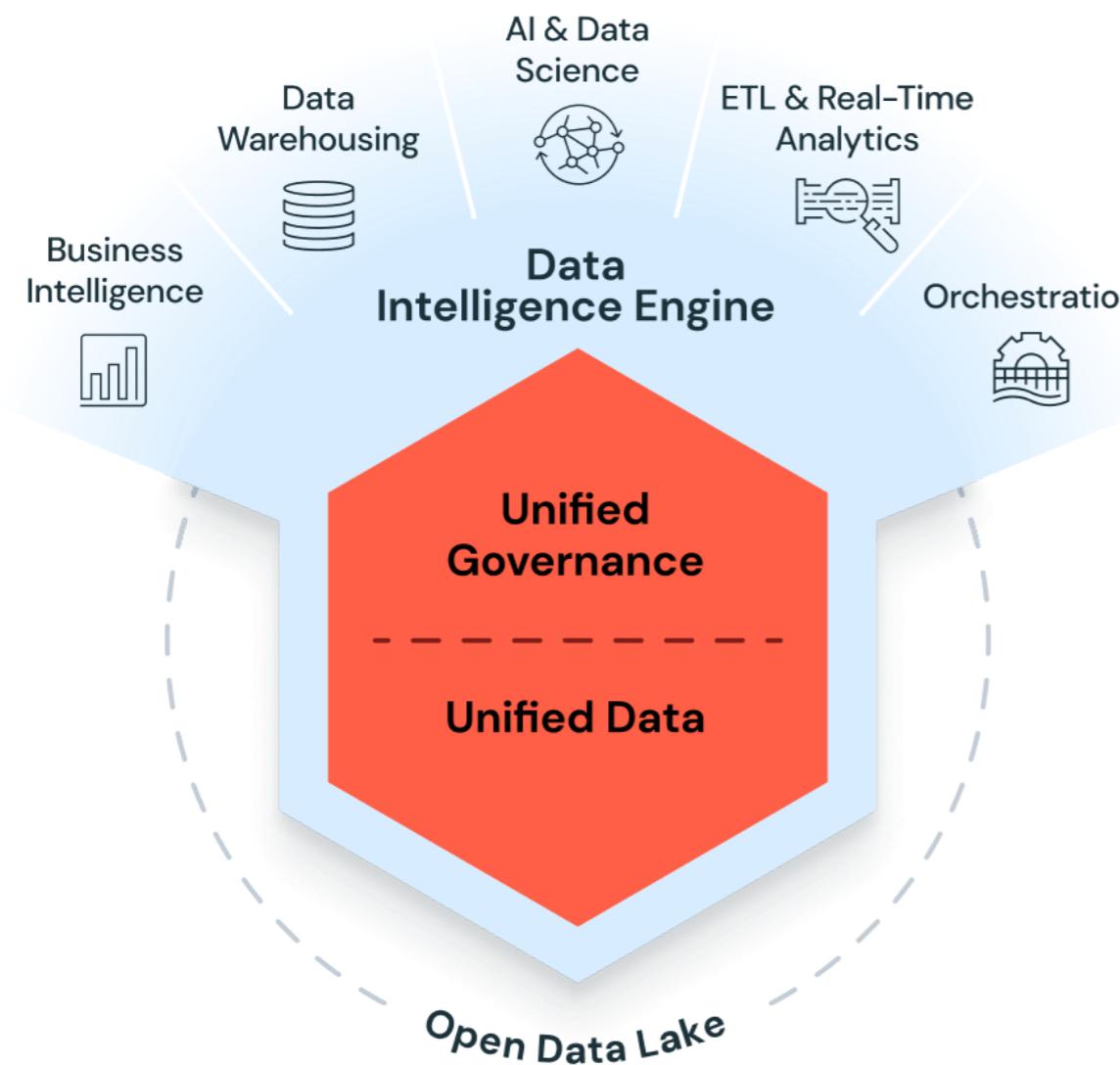


Databricks SQL

The Data Intelligence Platform is designed to run your most challenging data warehousing workloads.



Databricks for SQL Users

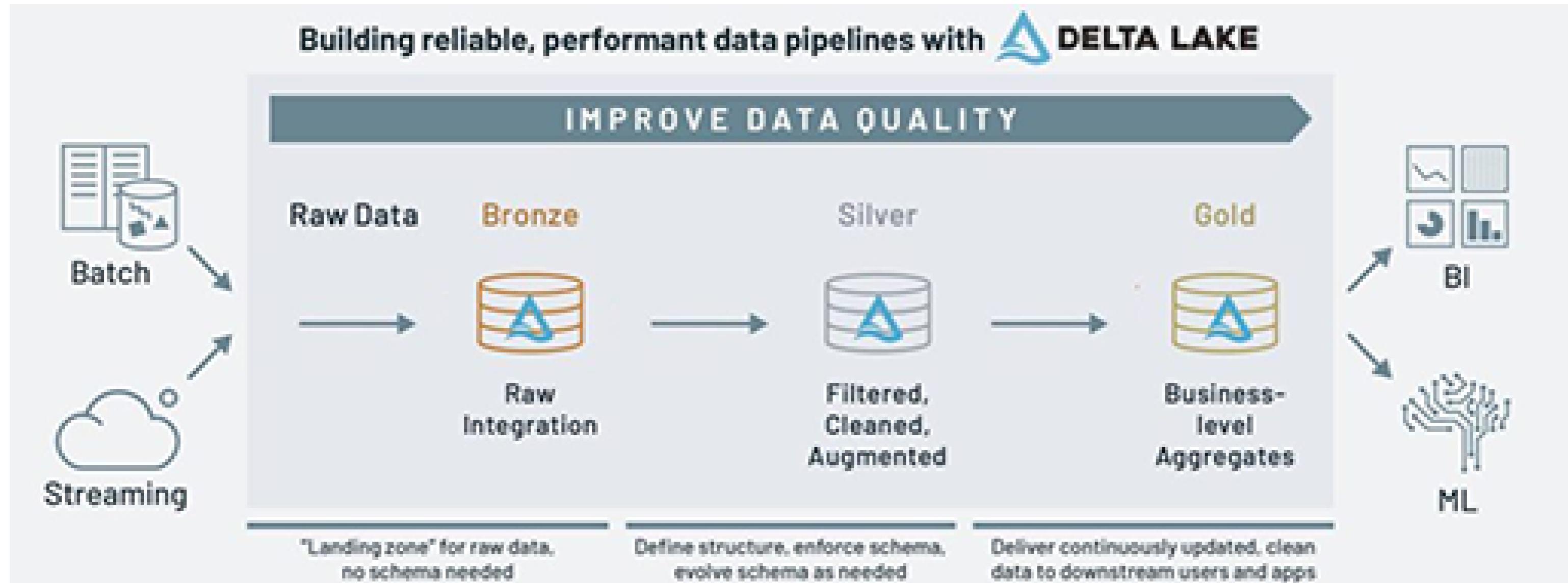


Databricks SQL

- Data Warehousing for the Lakehouse
- Familiar environment for SQL users
- SQL-optimized performance (Photon)
- Connect to your favorite BI tools

Comes built into the platform!

SQL in the Data Intelligence Platform



SQL in the Data Intelligence Platform

Benefits

- Fully integrated
- Scalable, performant compute
- Great UI for analysts

Key Features

- ANSI SQL
- Enhanced Photon engine
- Built-in visualizations

```
SELECT *
FROM json.`/Volumes/
catalog_name/schema_name/
volume_name/path/to/data`  
SELECT *
FROM catalog_name.schema_name.table_name  
CREATE TABLE
catalog_name.schema_name.table_name
AS
SELECT *
FROM ...
```

Let's review!

INTRODUCTION TO DATABRICKS

Congratulations!

INTRODUCTION TO DATABRICKS



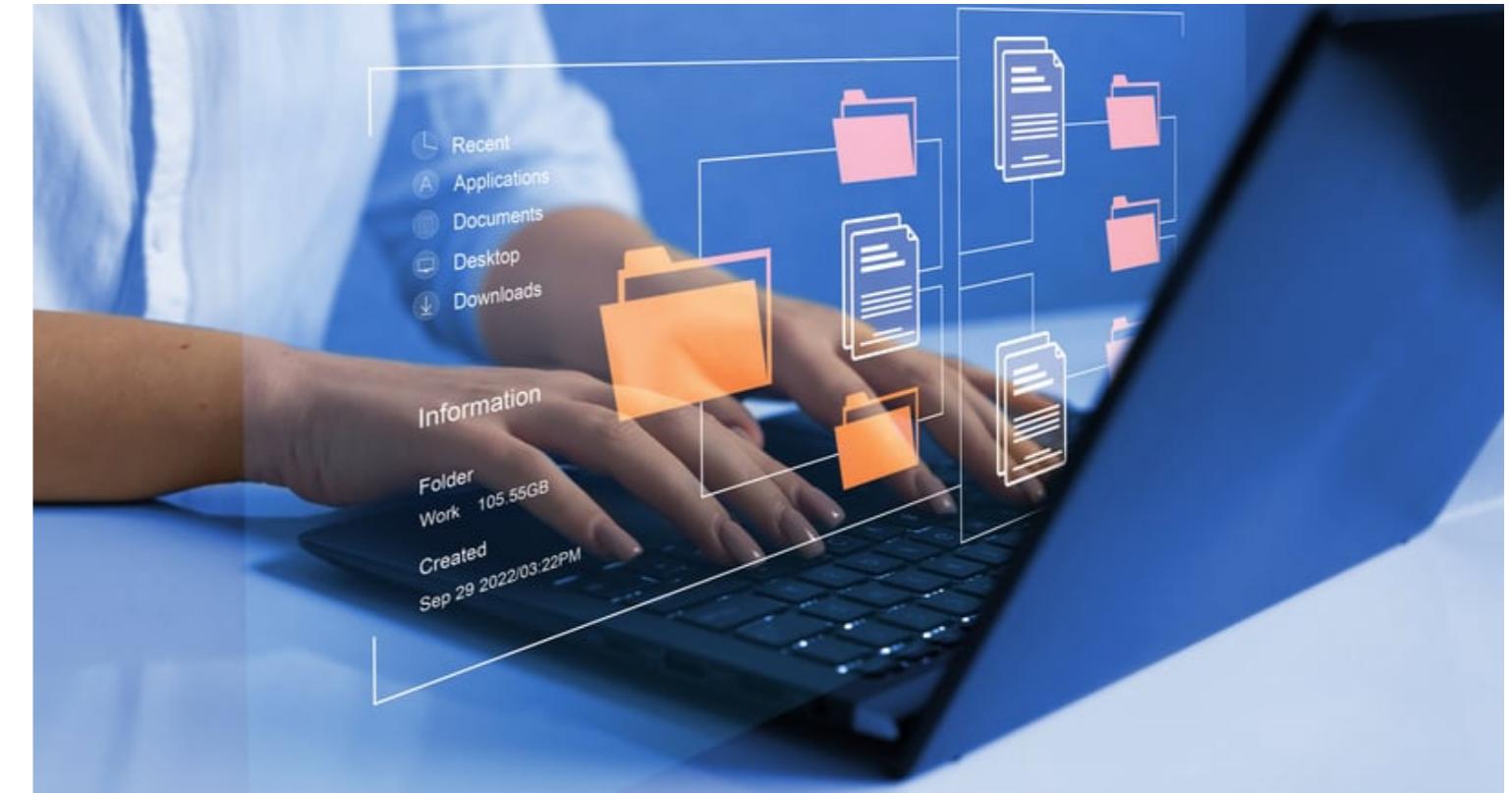
Chapter 1

- Introduction to the Data Intelligence Platform
- Key benefits of building on the platform
- Understanding the underlying architecture



Chapter 2

- Learning about Unity Catalog and Delta
- Managing organizational data
- Utilizing clusters and SQL Warehouses for scalable compute power



Chapter 3

- Conducting analytical processes in Databricks
- Using Databricks as an enterprise Data Warehouse
- Exploring Databricks SQL for analytics



Just the start



Congratulations!

INTRODUCTION TO DATABRICKS

SQL in the Data Intelligence Platform

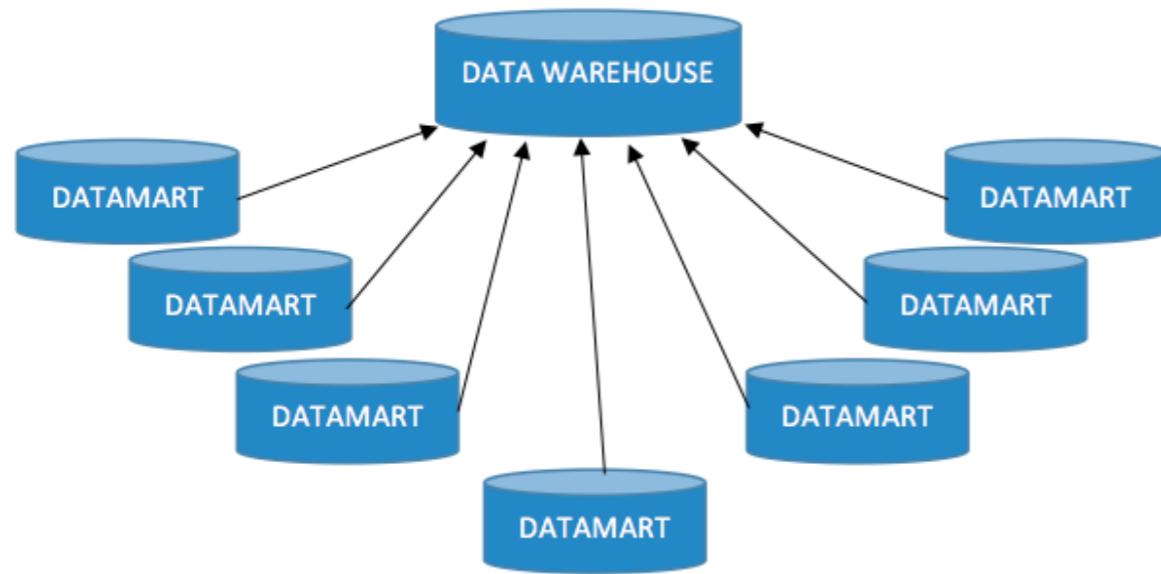
INTRODUCTION TO DATABRICKS SQL



Motivation

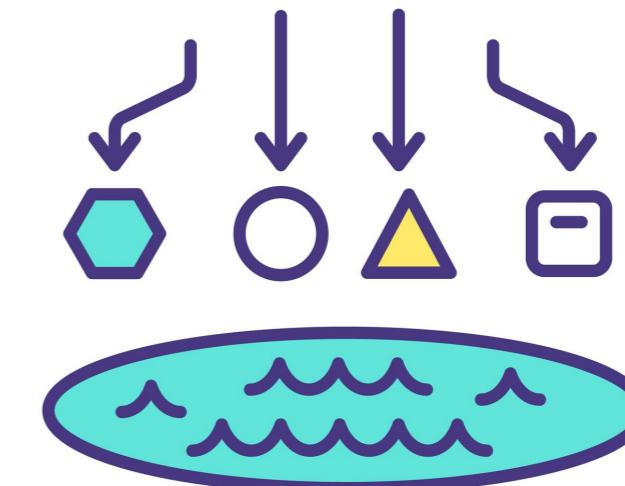
Data Warehouses

- Great for SQL workloads
- Typically expensive
- Proprietary technologies
- Limited capabilities and integrations

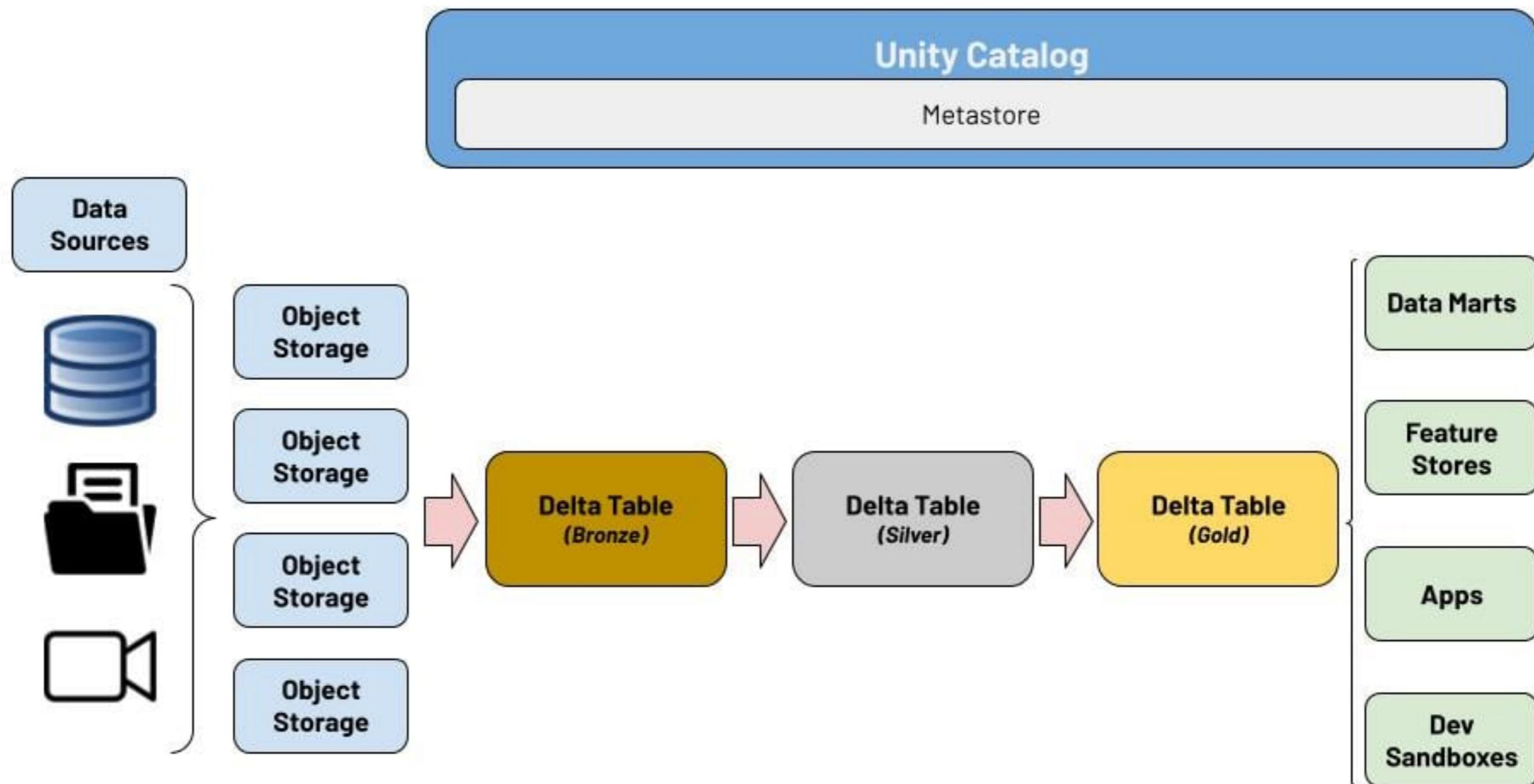


Data Lakes

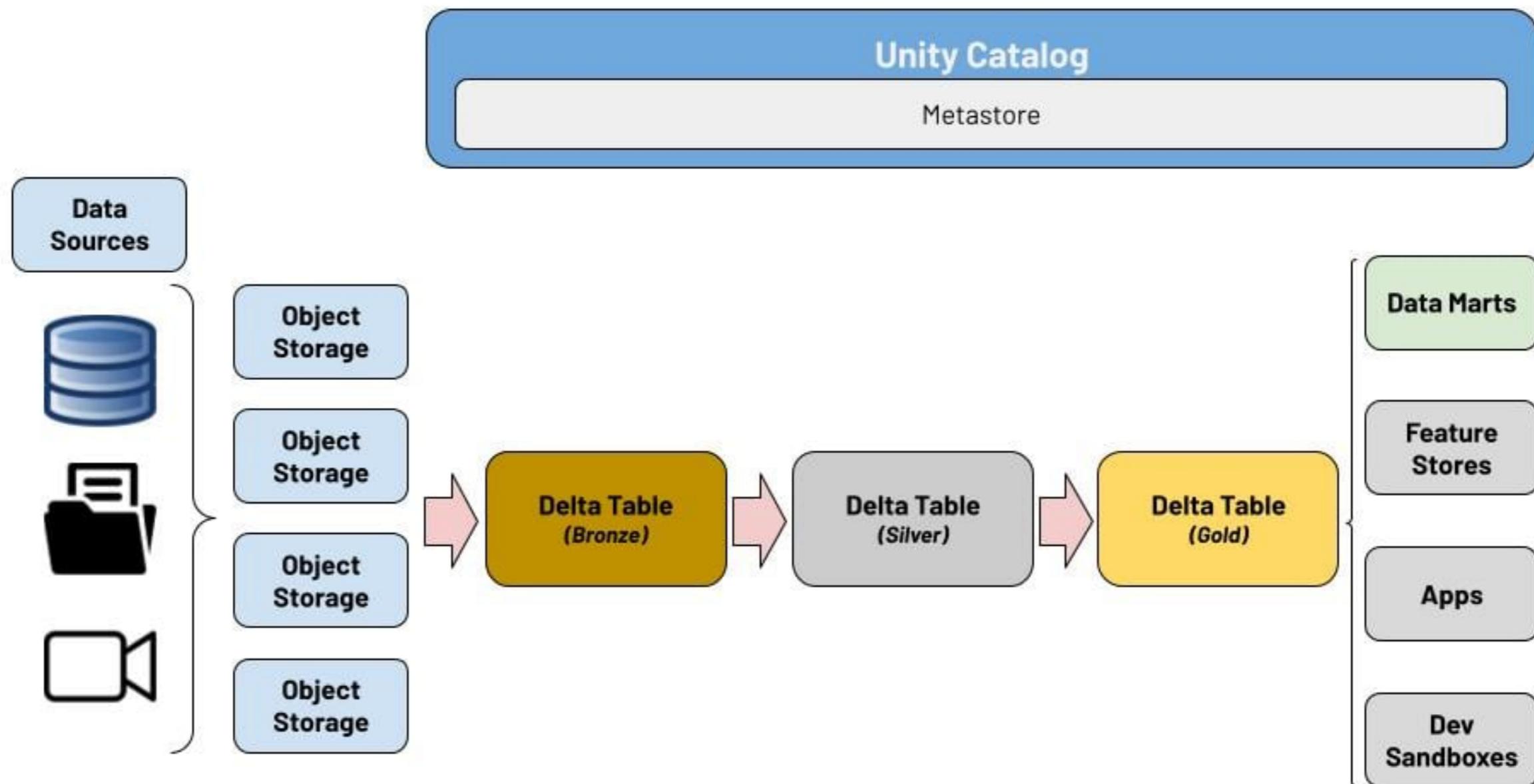
- Great for non-SQL workloads
- Cost effective, lackluster performance
- Open-source technologies
- Unlimited capabilities



Data warehousing in the Lakehouse

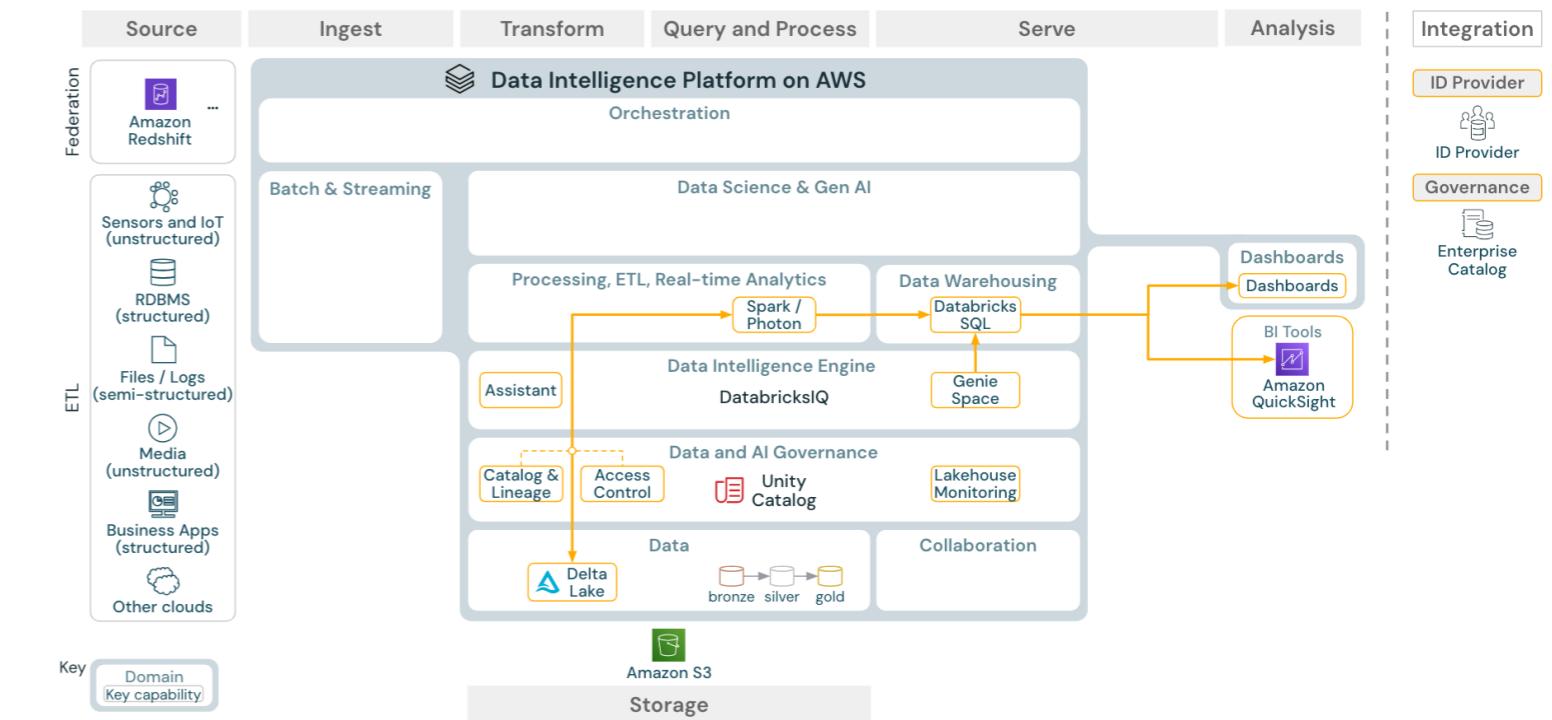


Data warehousing in the Lakehouse



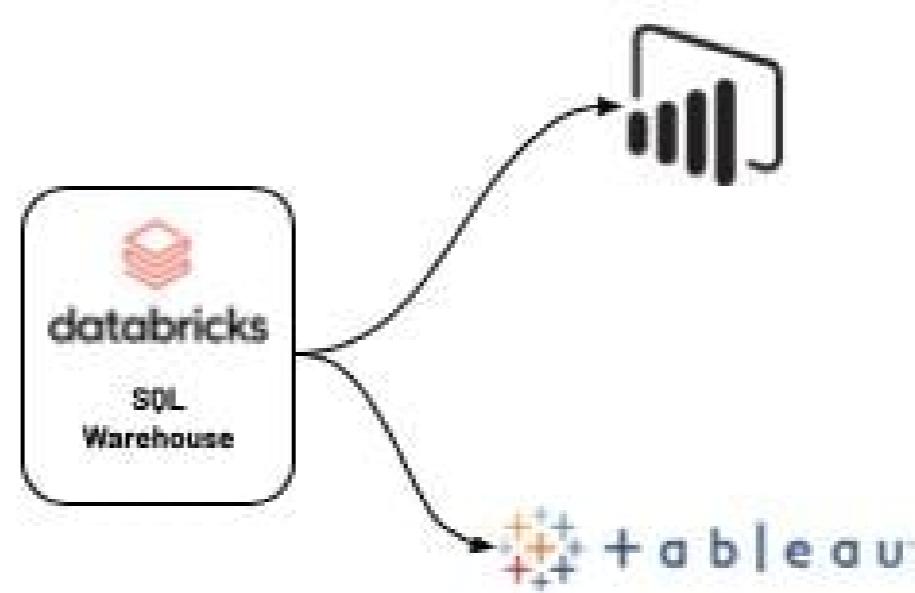
Benefits

- Single architecture for all workloads
- Flexibility and ownership with data
- Open-source technologies
 - Delta
 - ANSI SQL
- Cost effective solution



Business Intelligence ecosystem

- Integrate directly with your BI tool of choice
 - Partner Connect
 - Databricks Connect
 - JDBC / ODBC
- Performance and scalability
- Keep users where they are



Let's practice!

INTRODUCTION TO DATABRICKS SQL

Exploring Databricks SQL

INTRODUCTION TO DATABRICKS SQL



Databricks SQL key assets

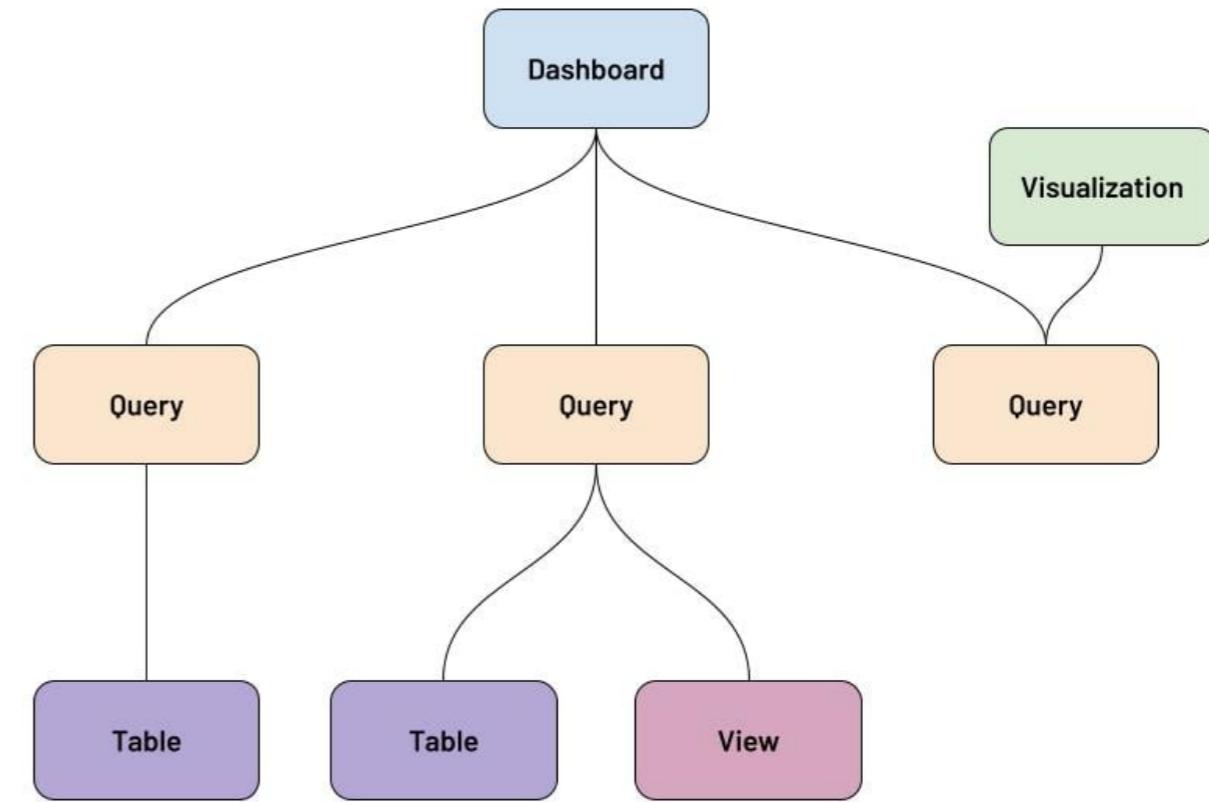
INTRODUCTION TO DATABRICKS SQL



Helpful analogy

A tree consists of many different components, all of which make up the entire entity

In Databricks SQL, different components combine into a data warehouse solution



Query

- The base "unit" of analysis in Databricks SQL
- Runs SQL code against compute
- Uses ANSI SQL standard
- Process data from:
 - Unity Catalog
 - Delta tables
 - Data lake files
 - Data streams

```
SELECT
```

```
orderdate AS Date,  
orderpriority AS Priority  
sum(totalprice) AS TotalPrice
```

```
FROM sfdc.sales.orders
```

```
GROUP BY
```

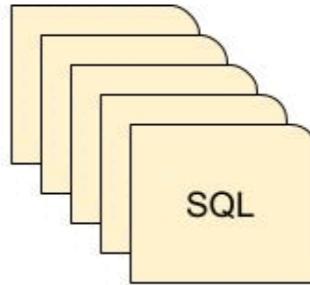
```
1, 2
```

```
ORDER BY
```

```
1, 2
```

SQL Warehouse

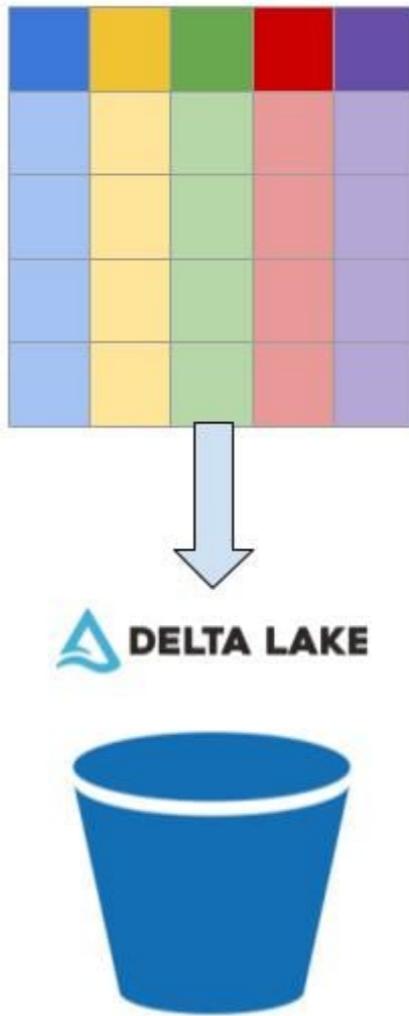
- Compute cluster dedicated for SQL
- Optimizations (e.g. Photon)
- Simpler administration
- Easy scaling
- Queries and BI tools



Tables versus views

Tables

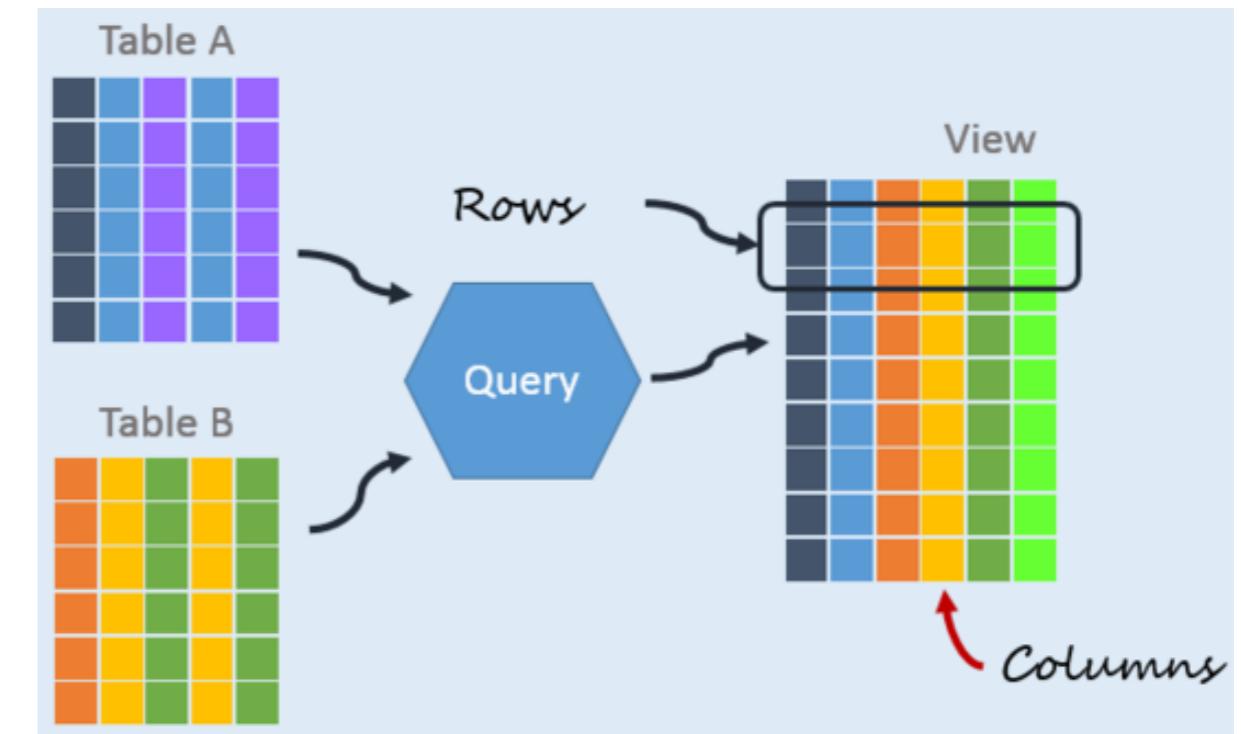
- Physical manifestations of datasets
- Written in Delta format
- Readable and accessible outside of the data pipeline
- Can optimize data layout (partitioning, etc.)



Tables versus views

Views

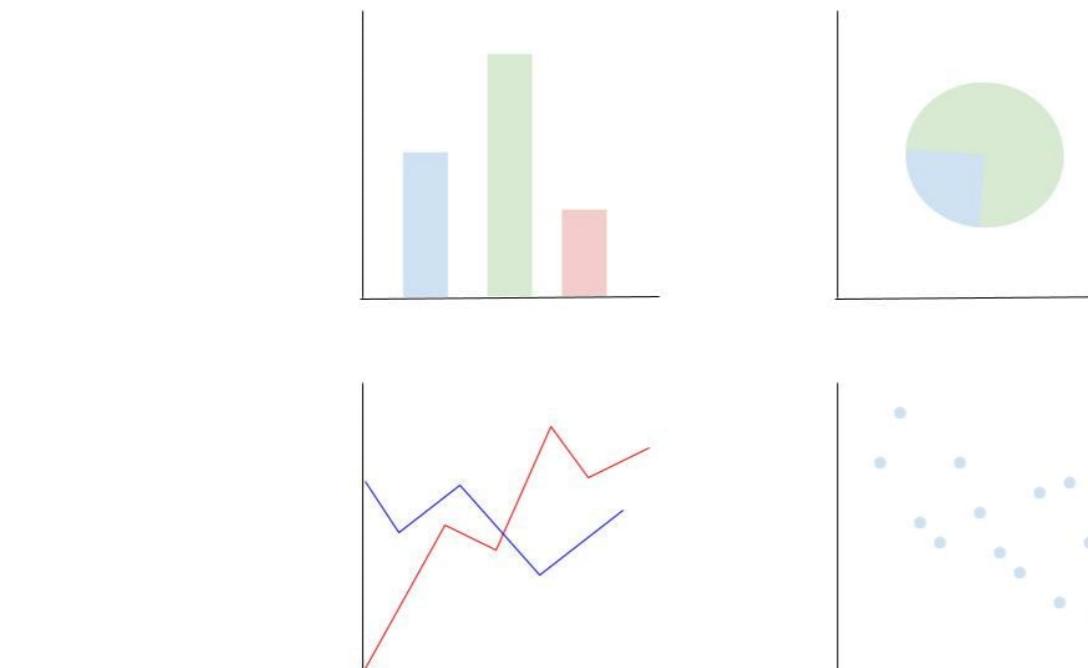
- Virtual representations of query results in Unity Catalog
- Fast performance for reading data
- Great for simplifying downstream queries
 - Source query has many joins, filters, etc.
- Incremental data processing available



Visualizations and dashboards

Visualizations

- Visual representations of a query result
- Created relative to a single query



Dashboards

- Collection of several visualizations
- Across multiple datasets / query results



Let's practice!

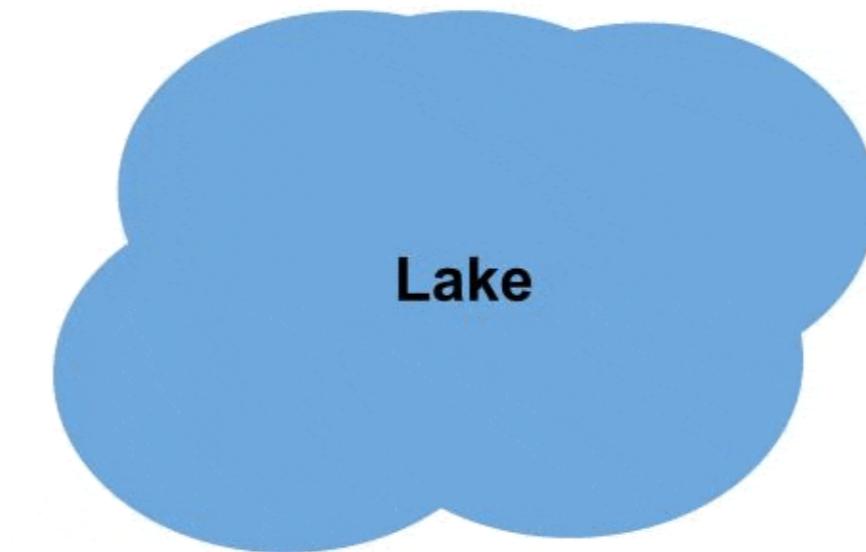
INTRODUCTION TO DATABRICKS SQL

Ingesting Data

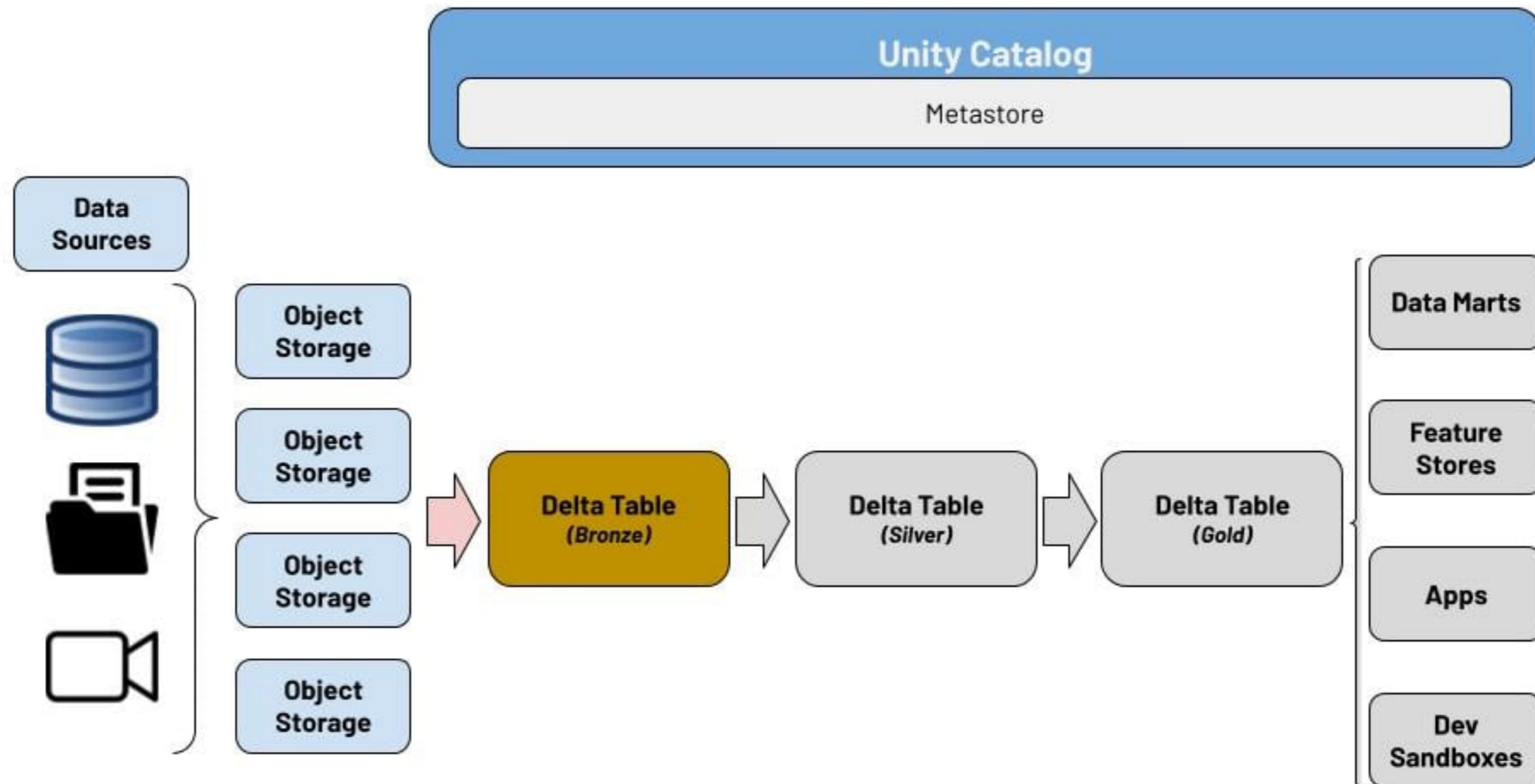
INTRODUCTION TO DATABRICKS SQL



Motivation



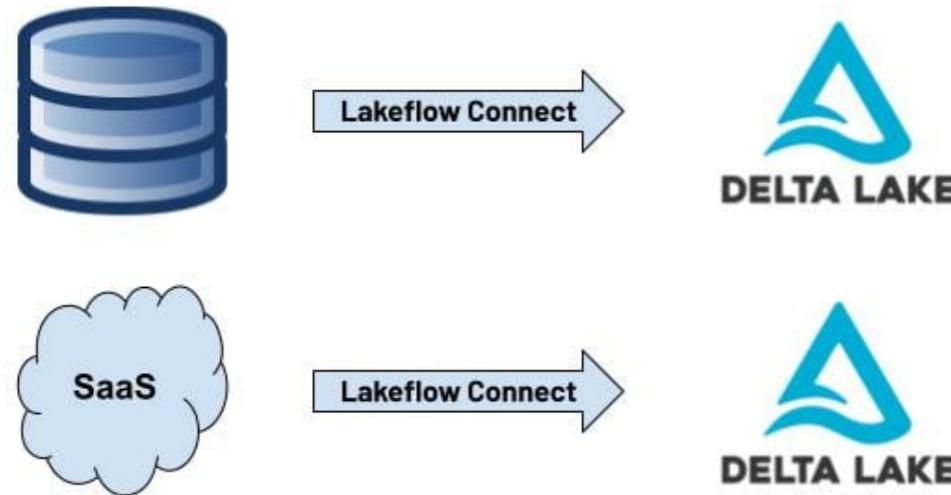
Creating the lakehouse



GUI-based options

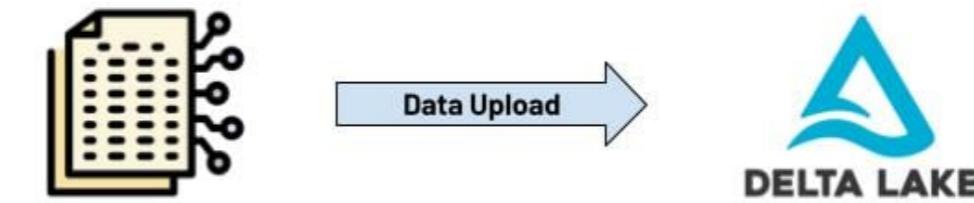
Lakeflow Connect

- Built-in connectors for ingesting data
 - Databases
 - SaaS Applications
- Creates pipelines to keep data up to date



Data upload

- Manually upload your files
 - CSV, Parquet, etc.
- Quickly create new Delta tables
- Great for ad hoc data upload



Bringing data into the lakehouse

COPY INTO

- Copy data from cloud object storage directly into Delta tables
- Better for more static datasets
- Can run natively in *SQL Editor*

```
COPY INTO my_table  
FROM '/path/to/files'  
FILEFORMAT = PARQUET  
FORMAT_OPTIONS ('mergeSchema' = 'true')  
COPY_OPTIONS ('mergeSchema' = 'true')
```

Auto Loader

- Automatically ingests new data files from cloud storage
- Better for larger and changing datasets
- Leverages *Delta Live Tables* in SQL

```
CREATE TABLE customers  
AS SELECT *  
FROM cloud_files(  
  "/path/to/files",  
  "csv")
```

Let's practice!

INTRODUCTION TO DATABRICKS SQL

Hydrating the lakehouse

INTRODUCTION TO DATABRICKS SQL



Let's practice!

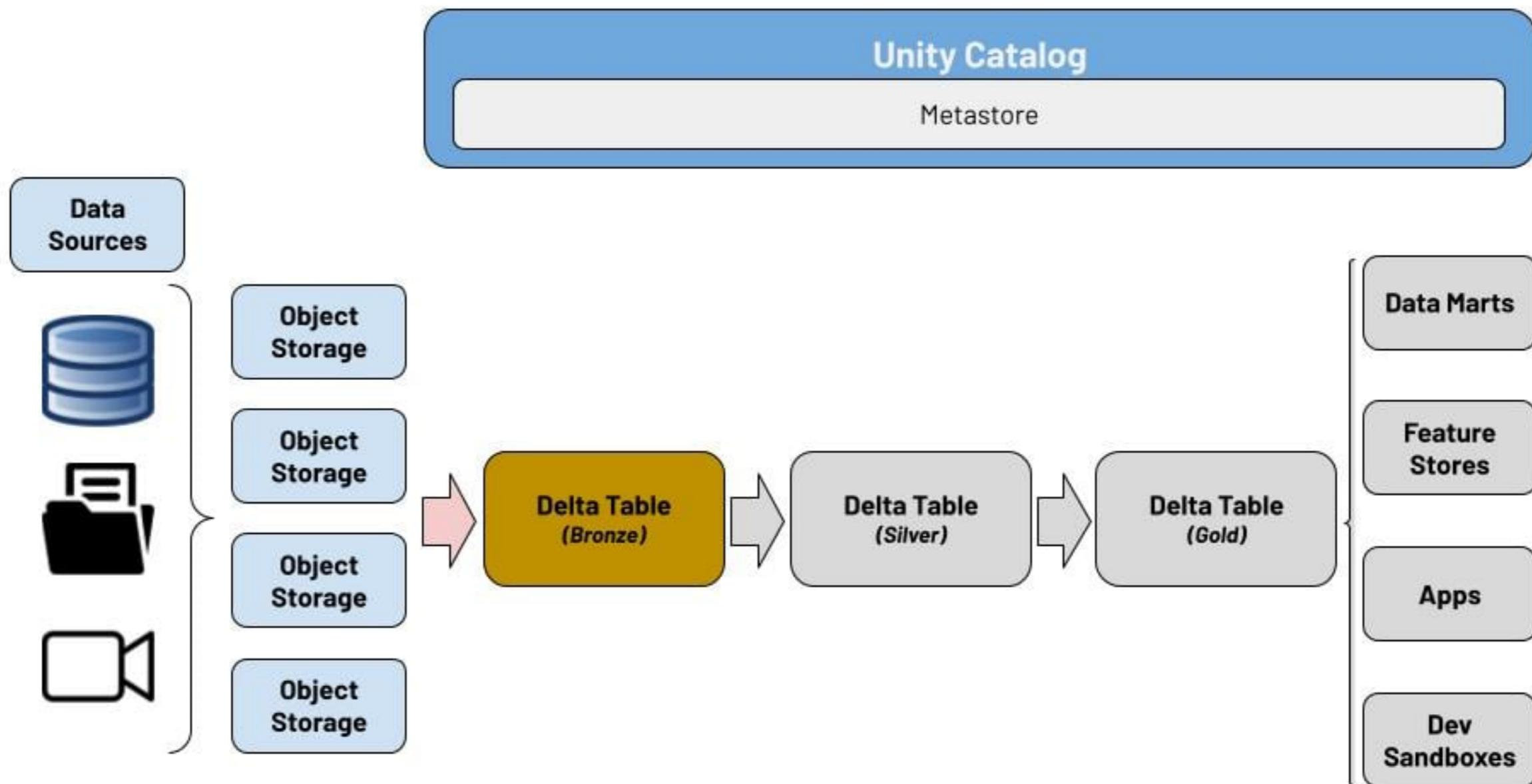
INTRODUCTION TO DATABRICKS SQL

Transforming data

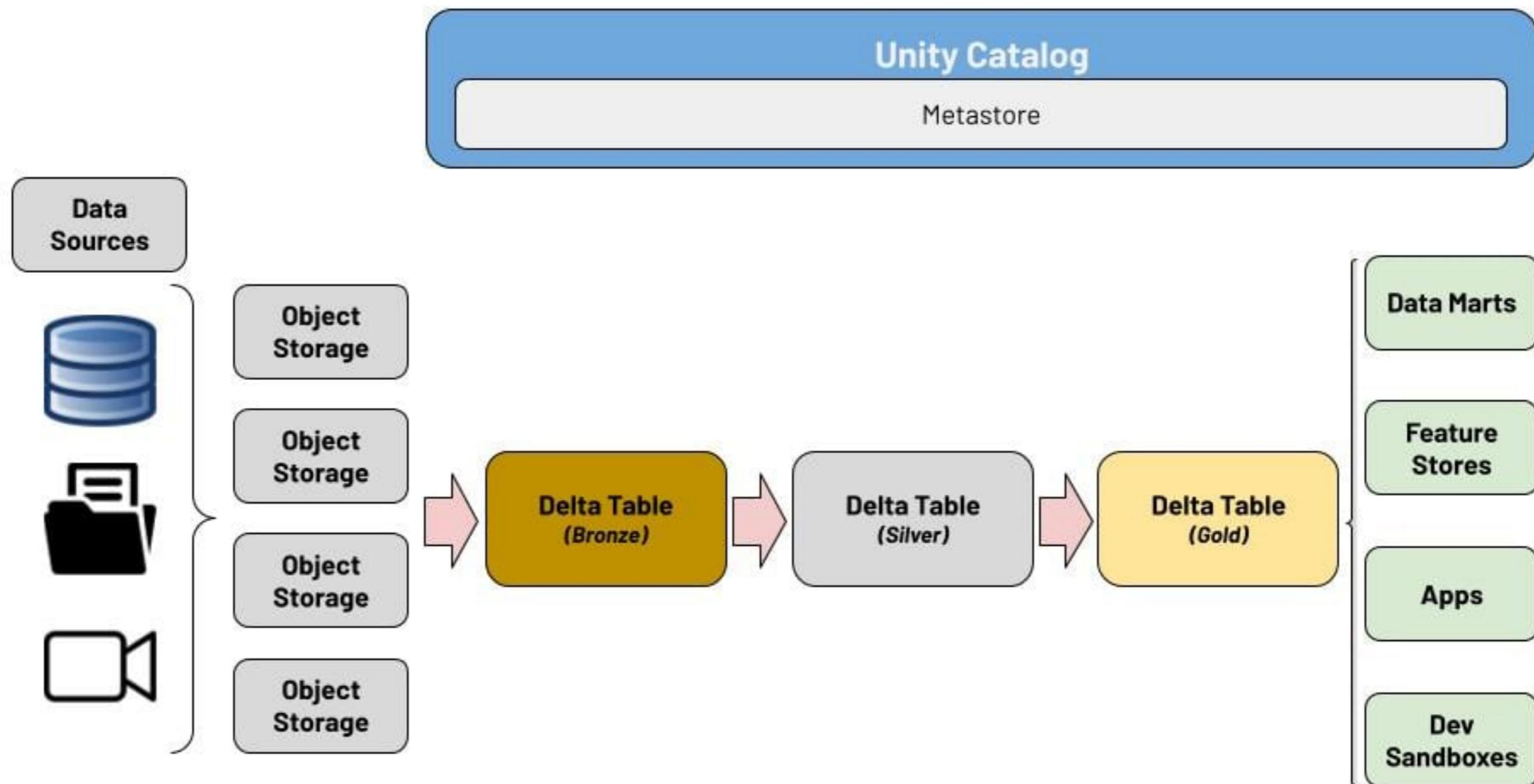
INTRODUCTION TO DATABRICKS SQL



Motivation



Transformation in the lakehouse



Cleaning and transforming data

- Cleaning data in the Bronze (raw) layer into the Silver (analytics-ready) layer
- Important step for downstream data tables
- Common activities
 - Removing NULL values
 - Standardize values
 - Adjusting data types

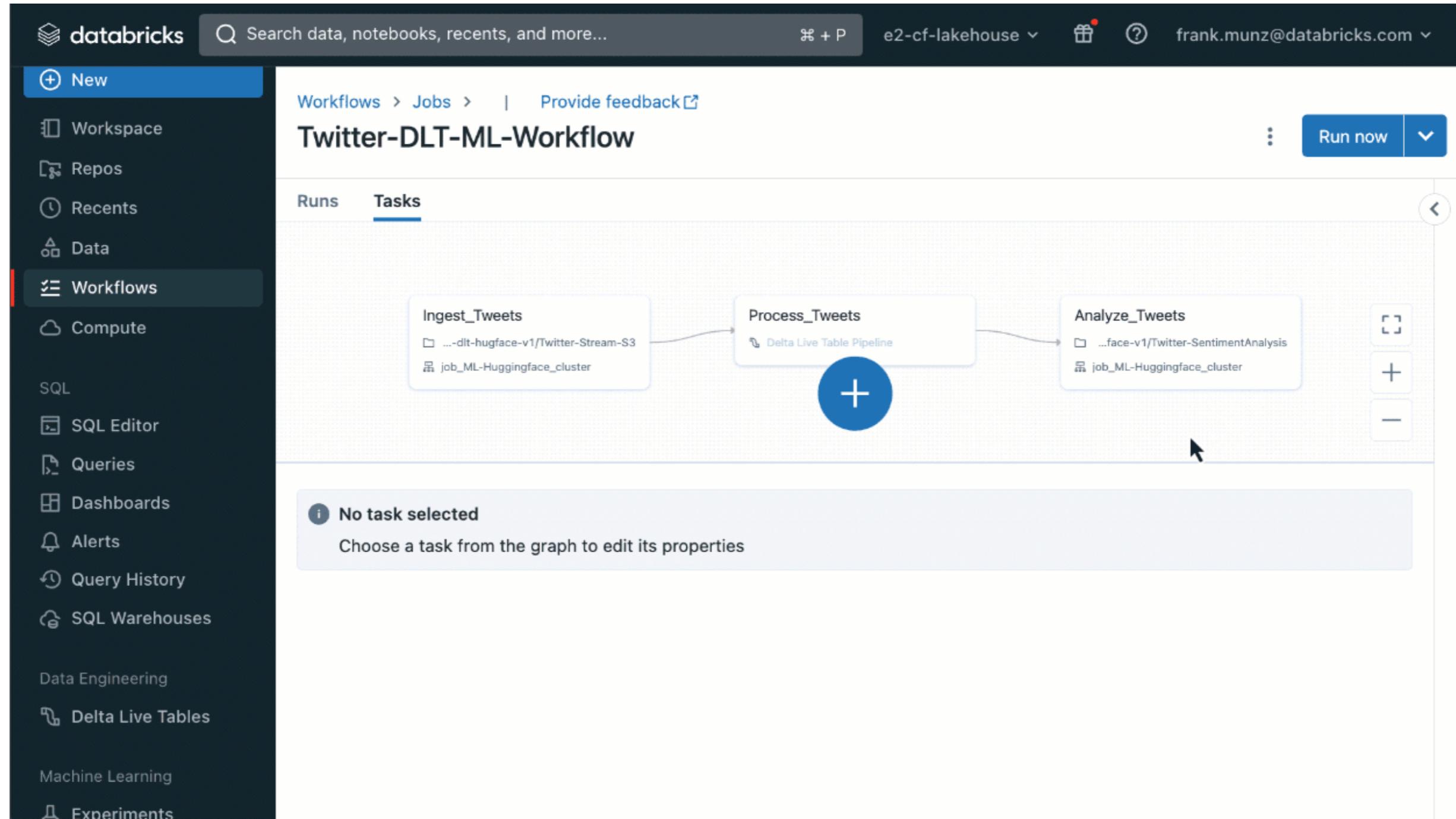
```
CREATE TABLE silver_layer AS (
SELECT DISTINCT c.id,
    c.last_name || ' , ' || c.first_name
        AS name,
    format(s.date, 'YYYY-mm-dd')
        AS sale_date,
    round(s.price, 2)
        AS sale_price
    s.item_name
FROM sales_data s
LEFT JOIN contacts c on c.id = s.id)
```

Aggregating data

- Combining and simplifying data from Silver layer into Gold (BI-ready) layer
- Meant for a specific business intelligence need
 - Great candidate for views
- Common activities
 - Removing extraneous columns
 - Aggregating across dimensions
 - Calculating metrics / KPIs

```
CREATE VIEW q3_revenue AS (
SELECT sum(revenue) AS total_rev,
       count(*) AS total_count,
       total_rev / total_count AS avg_sale,
       category,
       item
  FROM silver_layer
 WHERE date BETWEEN '2024-07-01'
   AND '2024-09-30'
 GROUP BY category, item)
```

Automating tasks



Let's practice!

INTRODUCTION TO DATABRICKS SQL

Creating a coffee data layer

INTRODUCTION TO DATABRICKS SQL



Let's practice!

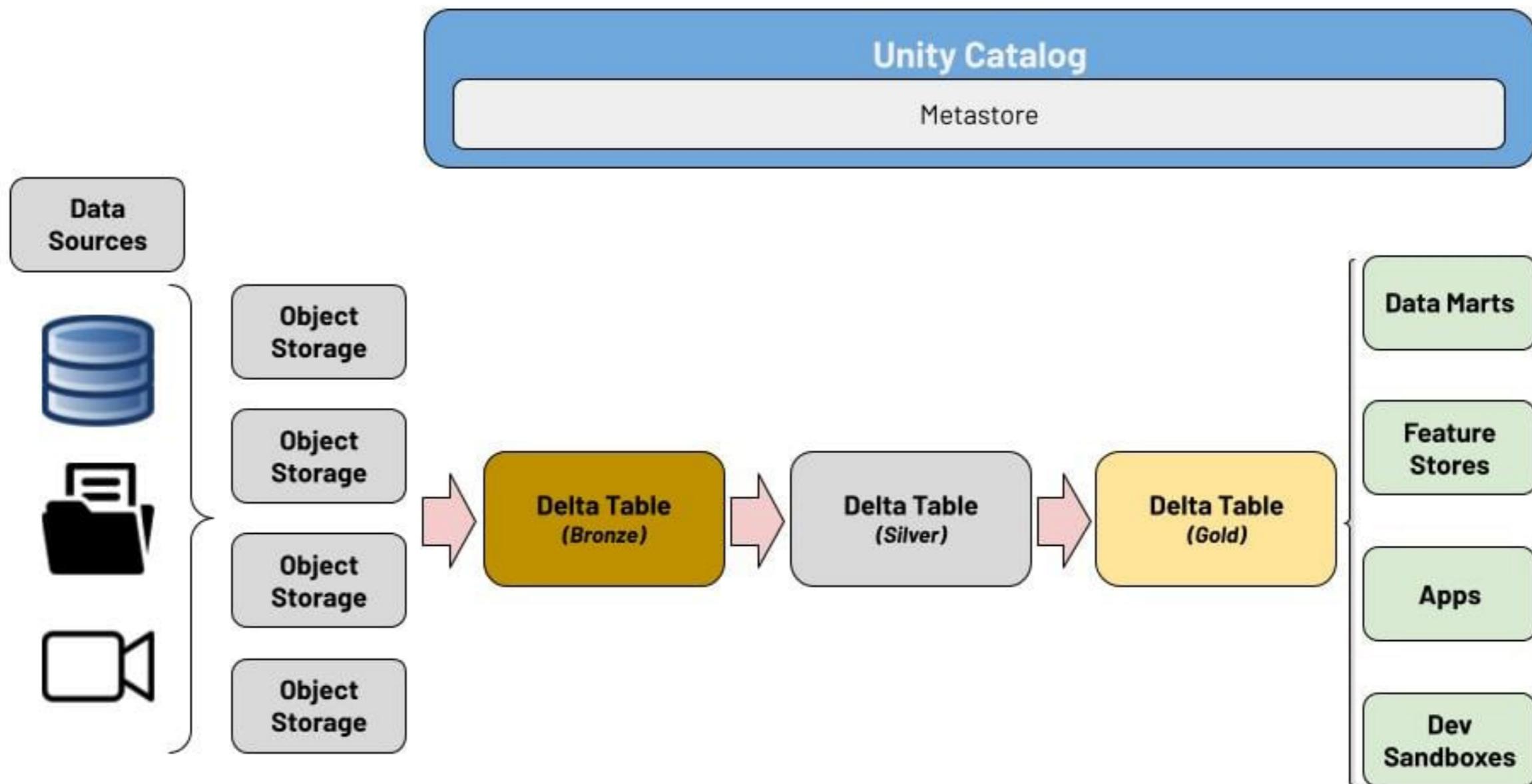
INTRODUCTION TO DATABRICKS SQL

Querying in the Data Intelligence Platform

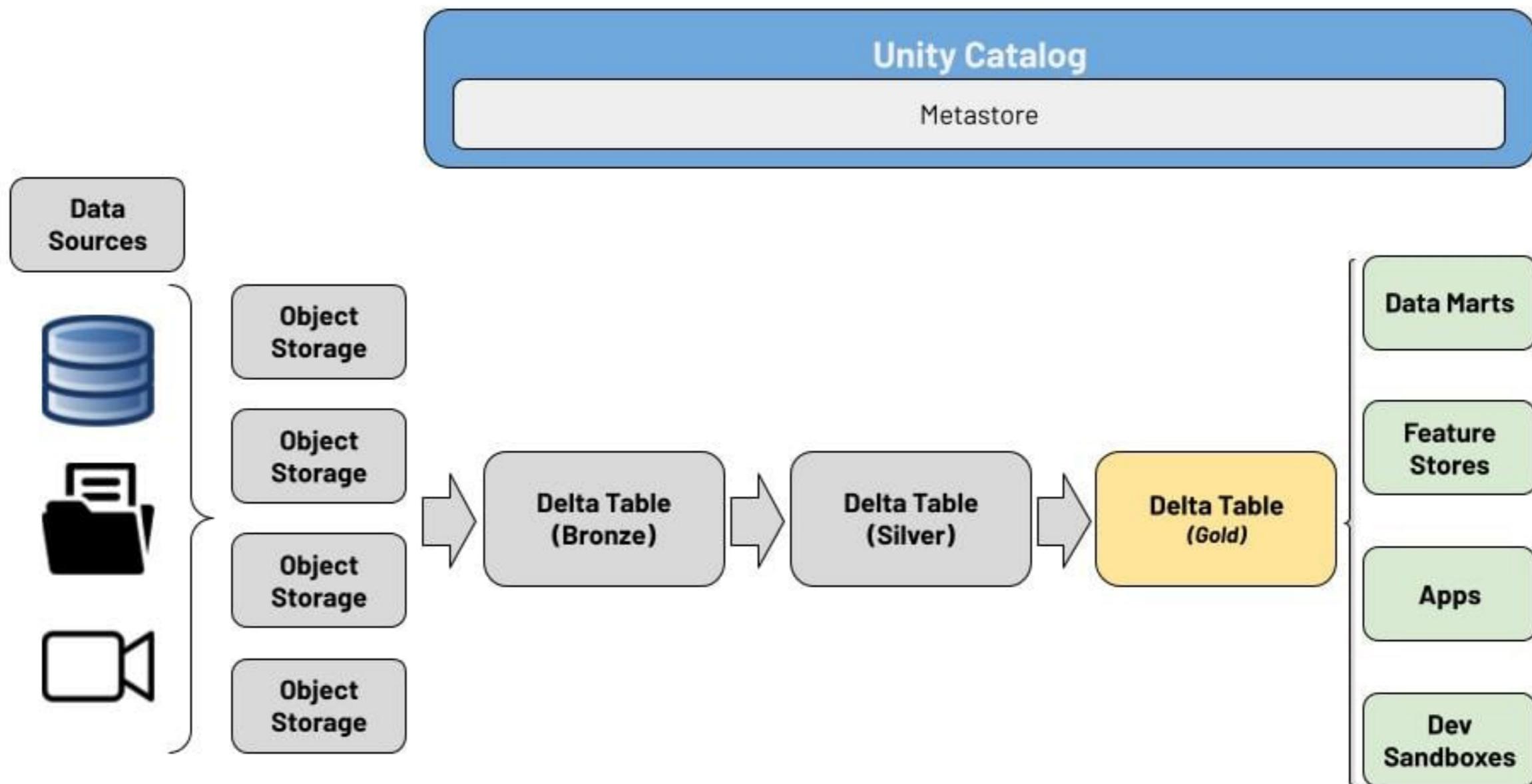
INTRODUCTION TO DATABRICKS SQL



Motivation



Motivation



SQL query basics

- Based on ANSI SQL
- Common patterns and functions to other SQL syntaxes
 - SELECT ... FROM ... syntaxes
 - Built-in and custom functions
 - Query data tables in Unity Catalog or in other database systems

SELECT

id,
name,
product,
store_id,
sales,
unit_price,

FROM

sales_data

WHERE

sales > 10 **AND**
product **IN** ('widget', 'thingy')

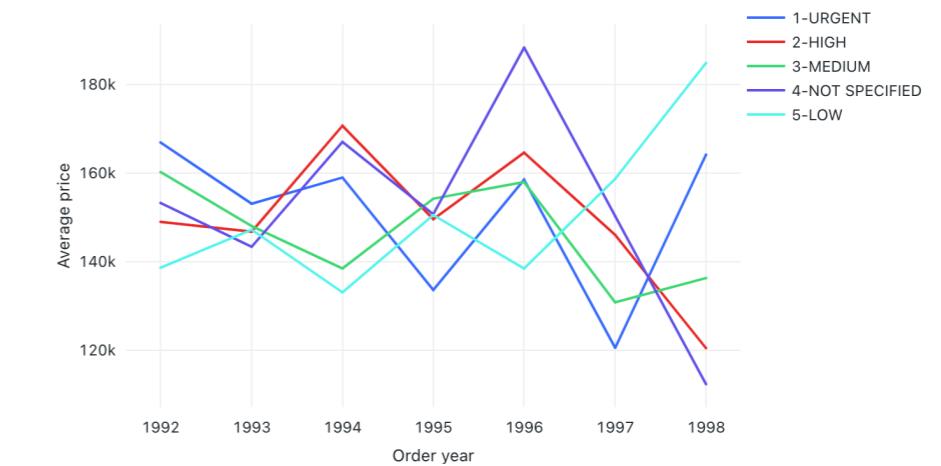
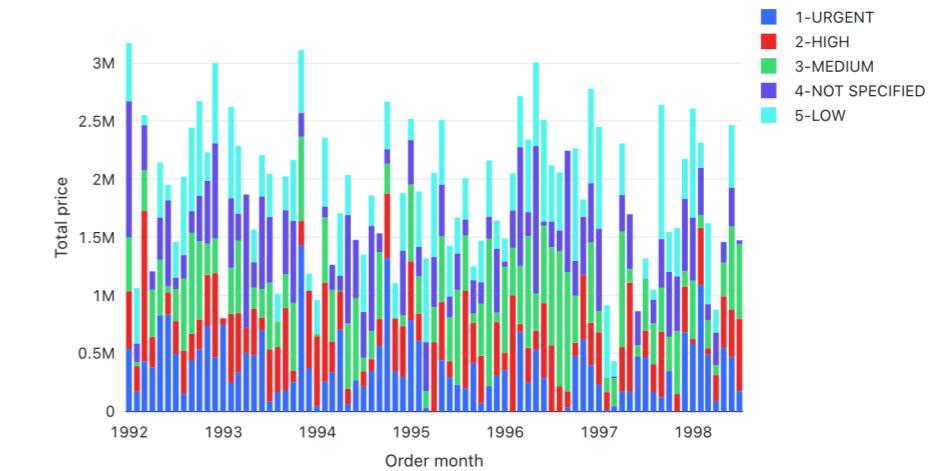
Common functions

- Databricks SQL functions mirror some of the most common operations in SQL, Python, and Spark
 - ROUND() and FORMAT_NUMBER()
 - CONCAT(), LEFT(), and RIGHT()
 - DATE(), DATE_ADD(), and DATE_DIFF()
 - CASE, IF(), and ISNULL()
 - FROM_CSV() and FROM_JSON()
- Create a custom User Defined Function (UDF)

```
SELECT  
    id,  
    initcap(name) as name,  
    right(product, 10) as productSKU,  
    store_id,  
    int(sales) as numSales,  
    round(unit_price, 2) as unit_price  
FROM  
    sales_data  
WHERE  
    sales > 10 AND  
    product IN ('widget', 'thingy')
```

Visualizations

- Visual representations of our query results
- Support for the most common visual types
 - Bar and line charts
 - Donut charts
 - Map visualizations
 - Pivot tables



Let's practice!

INTRODUCTION TO DATABRICKS SQL

Querying our coffee dataset

INTRODUCTION TO DATABRICKS SQL



Let's practice!

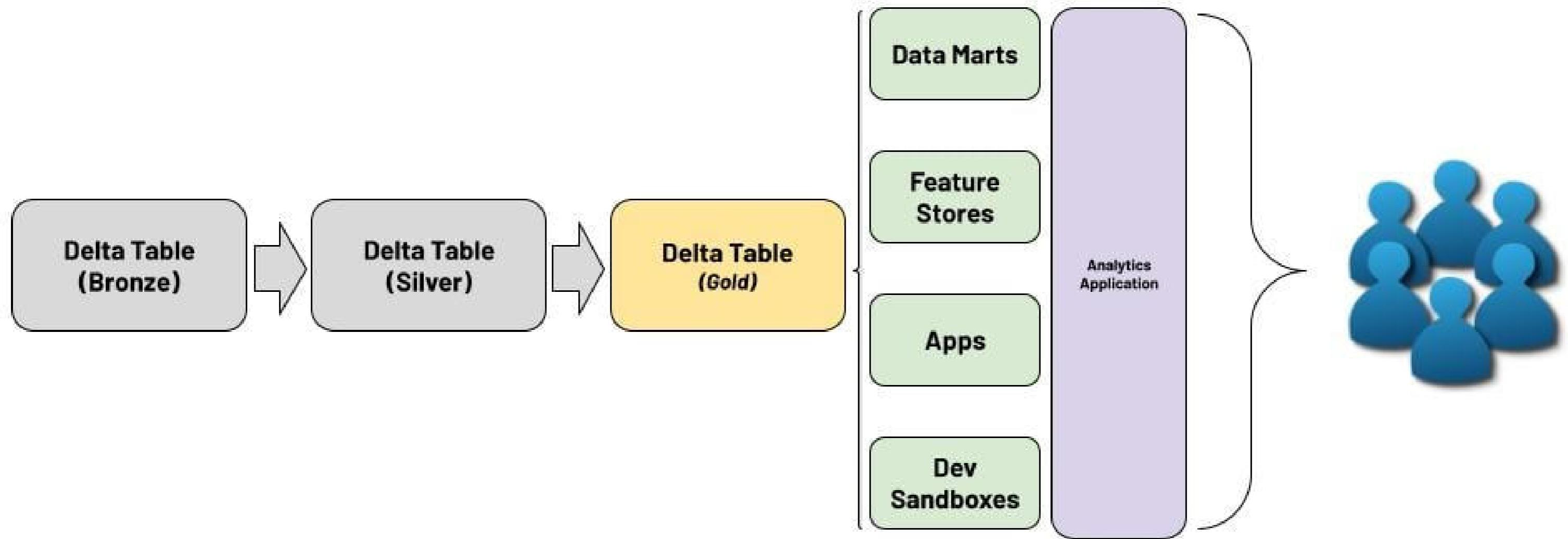
INTRODUCTION TO DATABRICKS SQL

Creating an analytics application

INTRODUCTION TO DATABRICKS SQL



What is an analytics application?



Filters and parameters

Filters

- Creates a sub-selection of the resulting rows based on the specified value or criteria
- Functions the same as a `WHERE` clause in a SQL query
 - Text-based, numerical, date ranges, dropdowns

Parameters

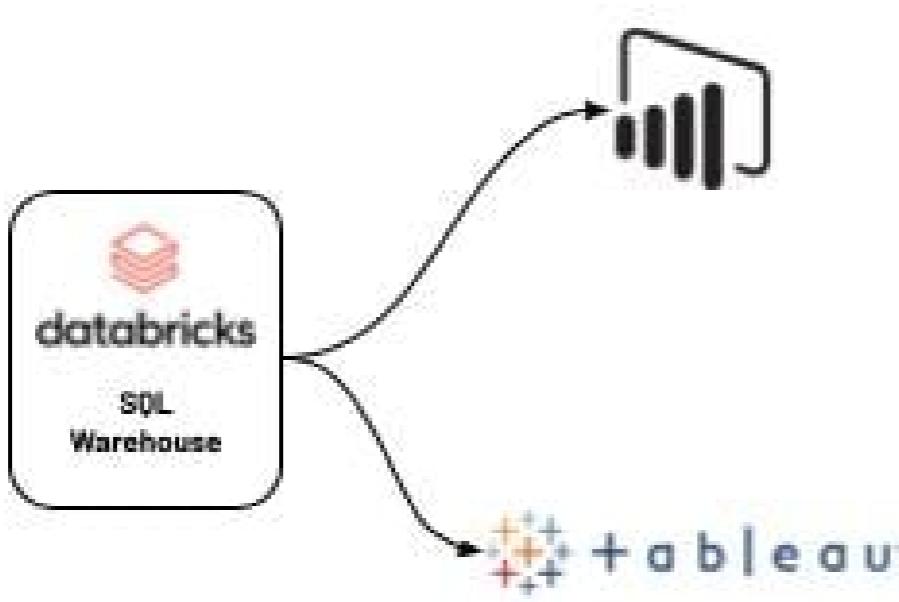
- Arbitrary text values that can be injected into a SQL query
- Often can be used as a filter
- Also able to inject other text values
 - Dynamically add fields to query
 - Change the behavior of a function

Dashboards in Databricks SQL

- Collection of visualizations from multiple queries and tables
- Allows for a more dynamic and user-friendly experience for understanding data
 - Leverage existing queries / visualizations
 - Create new queries and visualizations ad hoc
 - Bring in advanced logic with AI / LLMs
- Shareable through a URL



Partner Connect



- Leverage an extensive network of partner technologies with your lakehouse architecture
- Directly connect your favorite BI tool
 - Scalability and performance of compute with Databricks
 - User-friendly and powerful BI capabilities with Power BI, Tableau, etc.
- Keep users with the tools they are familiar with

Let's practice!

INTRODUCTION TO DATABRICKS SQL

A coffee data dashboard

INTRODUCTION TO DATABRICKS SQL



Let's practice!

INTRODUCTION TO DATABRICKS SQL

Managing local dashboards

DATA VISUALIZATION IN DATABRICKS



Cloning dashboards

Purpose of cloning:

- Create duplicates of existing dashboards for experimentation and modifications

Benefits:

- Allows safe exploration of changes without affecting the original dashboard
- Facilitates version control by keeping previous iterations intact



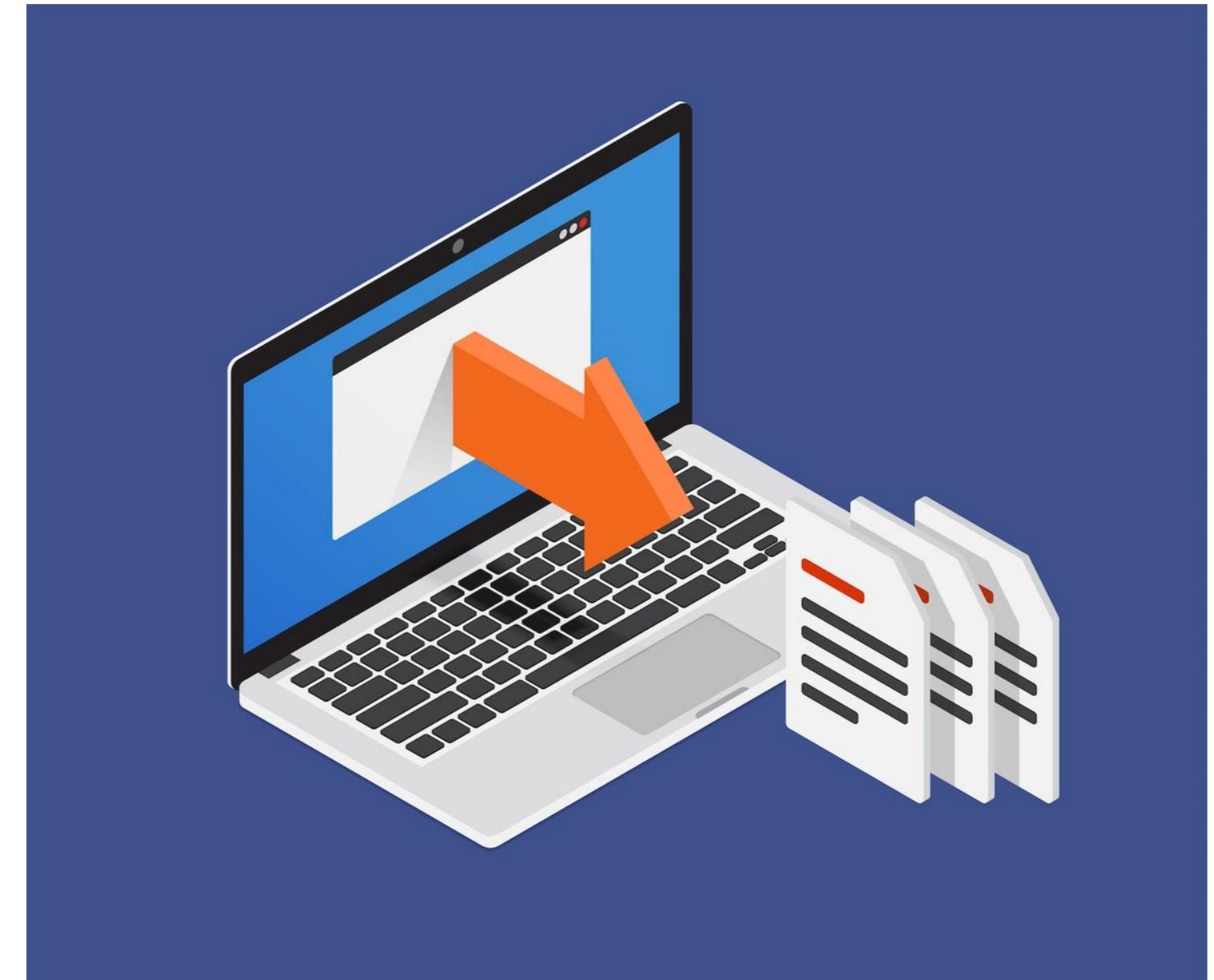
Exporting dashboards

Purpose of exporting:

- Save dashboards in various formats for external use and sharing

Benefits:

- Enables presentation of insights outside the primary workspace
- Allows for offline access and distribution of dashboard content



Deleting dashboards

Purpose of deleting:

- Remove unnecessary or outdated dashboards from the workspace

Benefits:

- Helps maintain a clean and organized dashboard environment
- Reduces clutter, making it easier to find relevant dashboards



Summary of local dashboard management

- Cloning helps in creating backups.
- Exporting allows for easy sharing.
- Deleting keeps your workspace organized.

Let's practice!

DATA VISUALIZATION IN DATABRICKS

Managing dashboards

DATA VISUALIZATION IN DATABRICKS



Let's practice!

DATA VISUALIZATION IN DATABRICKS

Sharing a dashboard

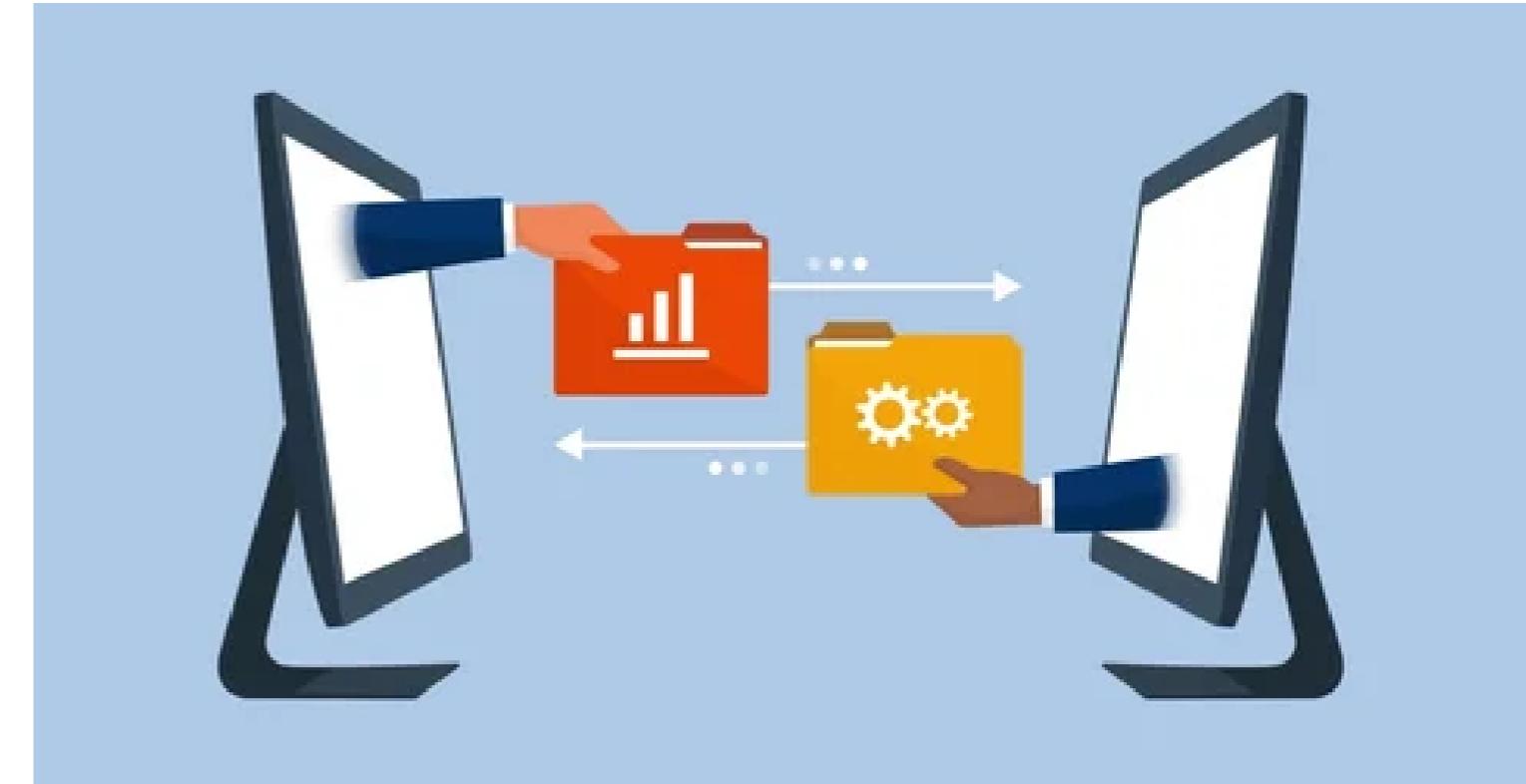
DATA VISUALIZATION IN DATABRICKS



Sharing dashboards within the workspace

Share with users and groups in your Databricks workspace

- **Permission levels:** View, Edit, Manage
- Access to both draft and published versions
- Inherit permissions from enclosing folders



Permissions levels

Ability	CAN VIEW/CAN RUN	CAN EDIT	CAN MANAGE
View dashboard and results	x	x	x
Interact with widgets	x	x	x
Refresh the dashboard	x	x	x
Edit dashboard		x	x
Clone dashboard	x	x	x
Publish dashboard snapshot		x	x
Modify permissions			x
Delete dashboard			x

Sharing dashboards with account members

- View-only access for users not in the workspace
- Granting access to account members with embedded credentials
- Administrator registration required for account members
- Options for sharing: specific users, groups, or everyone in the account

Managing alerts and notifications

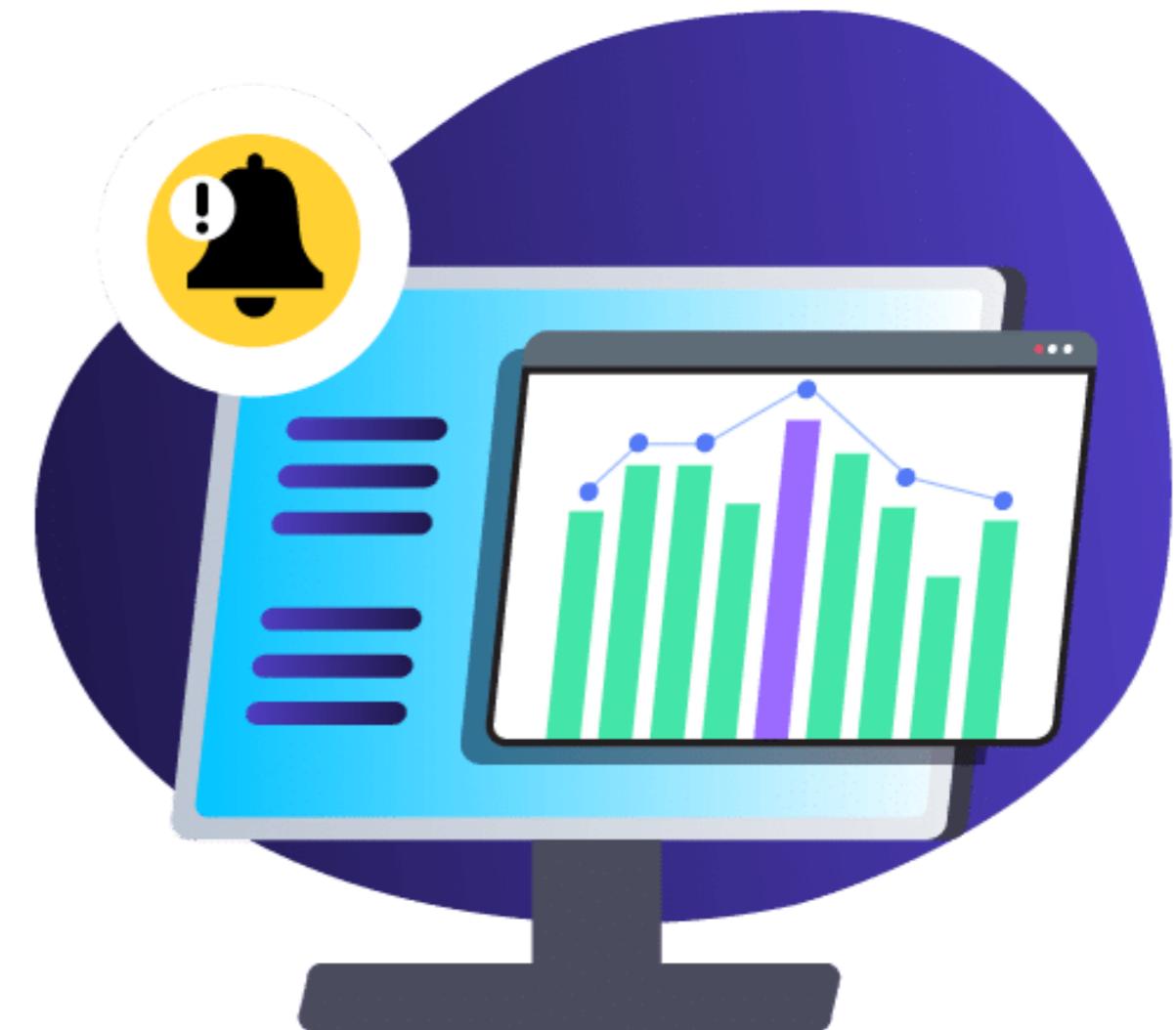
Alerts: Keep users informed of important changes in data.

Databricks SQL alerts

- Periodically run queries to evaluate defined conditions.
- Send notifications if a condition is met.

Setting up alerts: Define conditions for alerts based on metrics or thresholds.

Custom notifications: Tailor alerts to specific users or groups.



Summary of effective dashboard sharing

Sharing dashboards:

- Fosters team collaboration and informed decision-making.
- Share with users/groups in your Databricks workspace.
- Assign permissions: View, Edit, and Manage.

Managing alerts:

- Set alerts for significant data changes.
- Customize notifications for key business metrics.

Let's practice!

DATA VISUALIZATION IN DATABRICKS

Sharing dashboards in Databricks

DATA VISUALIZATION IN DATABRICKS



Let's practice!

DATA VISUALIZATION IN DATABRICKS

Recap

DATA VISUALIZATION IN DATABRICKS



Wrap-up

Chapter 1: Visualizations Basics

- Learn fundamental concepts of data visualization
- Create various chart types (bar, line, combo) and maps (choropleth, marker)

Chapter 3: Dashboard Creation

- Build dashboards by combining visualizations
- Manage dashboard updates and refresh schedules

Chapter 2: Formatting & Storytelling

- Format visual elements like colors, axes, and labels
- Explore data storytelling through customizable tables and visualizations

Chapter 4: Dashboard Management

- Clone, export, and delete dashboards
- Share dashboards with specific permissions and manage alerts

Thank you!

DATA VISUALIZATION IN DATABRICKS