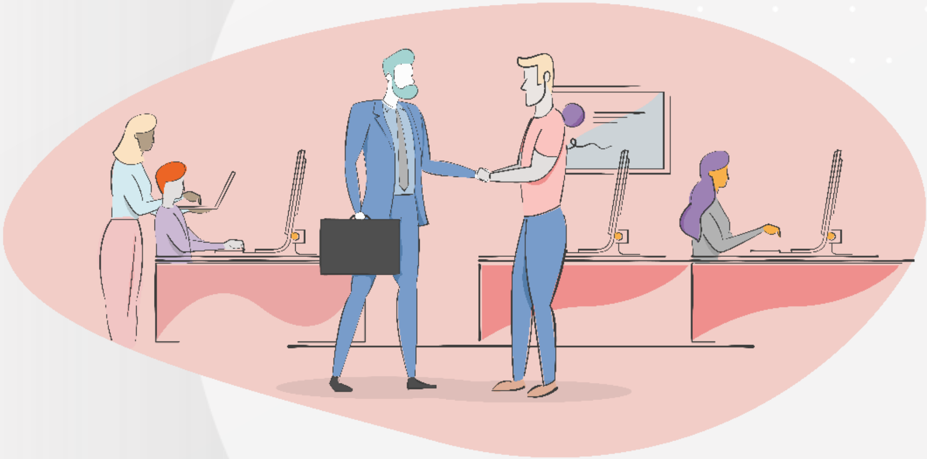
Schemas and Integration



L5 Data Engineer Higher Apprenticeship
Module 2 / 12 (**“Databases and Data Lakes”**)
Topic 2 / 5

rev. 1 (2024)

Webinar agenda

This webinar will cover the following:

- Data profiling
- Database archiving
- Query profiling
- Query optimisation
- Recovery
- Indexing
- Security
- Outages

Building Careers Through Education

Everybody Matters. Stronger Together. Student, Learner and Client Centric. Trust and Respect. Embrace Change.

BPP

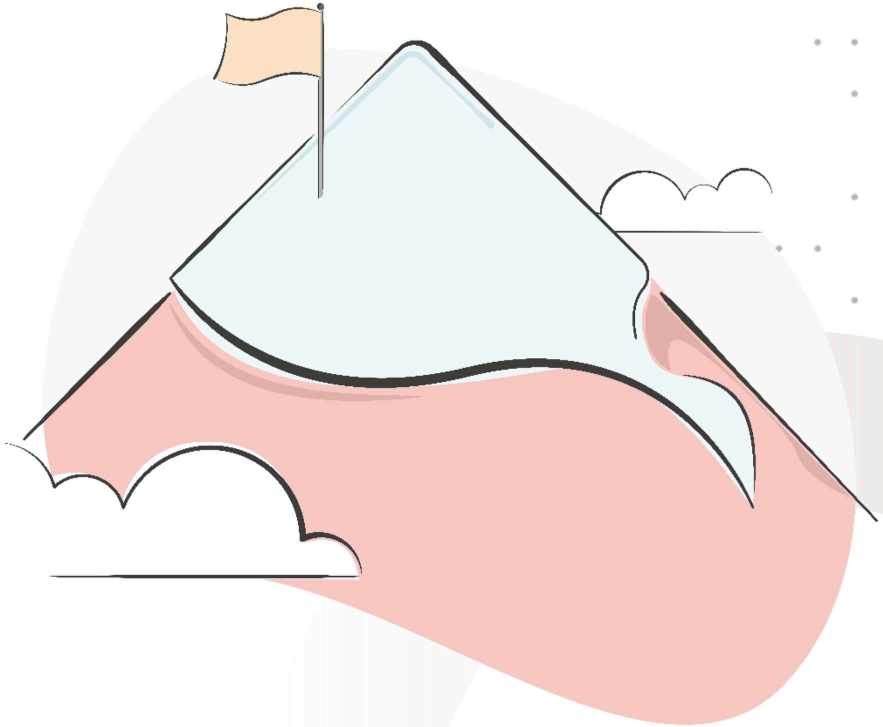
Learning objectives

Building Careers Through Education

Everybody Matters. Stronger Together. Student, Learner and Client Centric. Trust and Respect. Embrace Change.

By the end of today’s webinar, you will be able to:

- **Demonstrate** familiarity with industry database tools and best practices for designing and setting up databases
- **Explain** fundamental SQL concepts, including the impact of changes in database systems on diverse data consumers.
- **Demonstrate** the application of schemas, metadata, and data modeling to ensure data reliability and sustainability, and to mitigate data risks.
- **Evaluate** user and business needs to ensure the accuracy, completeness, consistency, timeliness, and accessibility of upstream data for downstream consumers, taking proactive responsibility to solve issues.



BPP

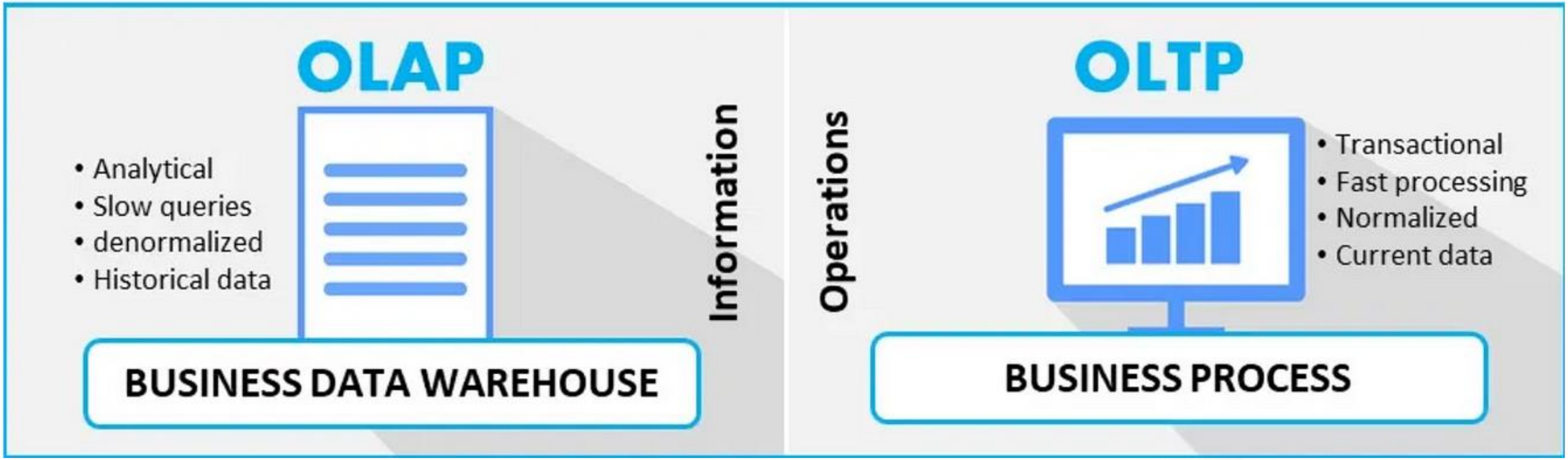
Recap

Recap of Pre-Learning

Overview of:

- relational data model
- types of keys
- relationships
- schemas

OLAP Vs OLTP



OLAP

- Analytical
- Slow queries
- denormalized
- Historical data

Information

BUSINESS DATA WAREHOUSE

OLTP

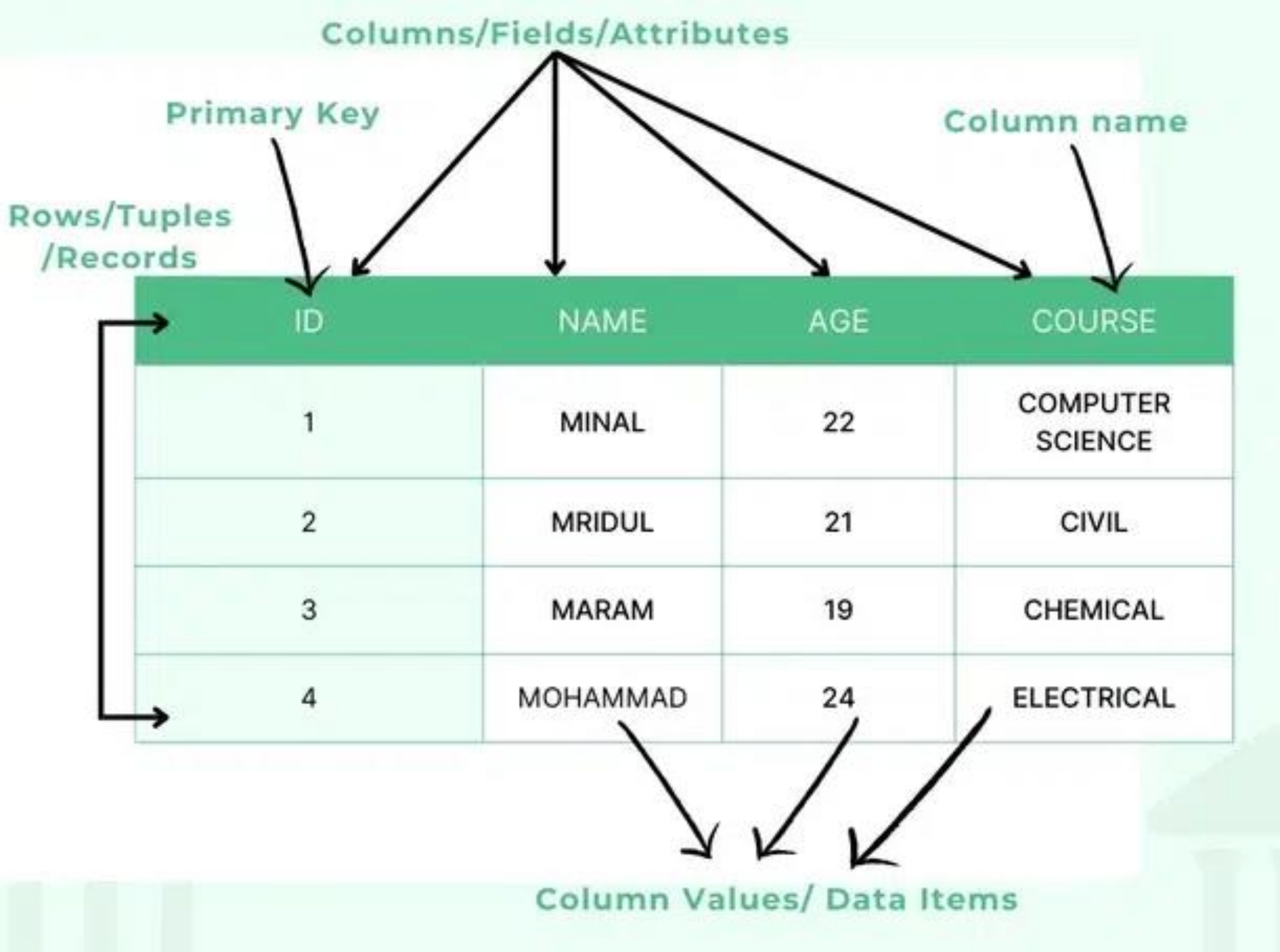
Operations

- Transactional
- Fast processing
- Normalized
- Current data

BUSINESS PROCESS

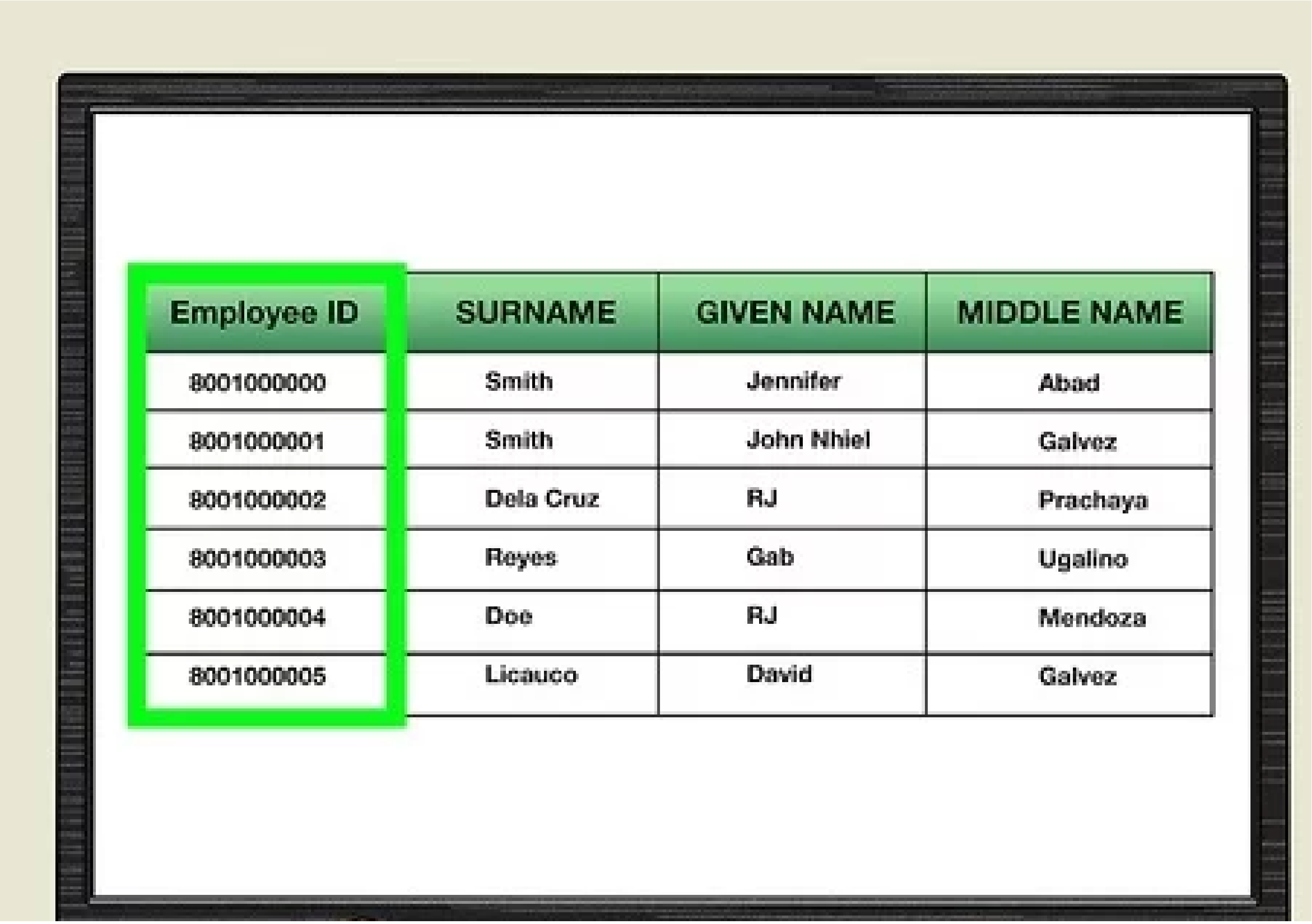
	OLTP	OLAP
Access Patterns	The access pattern of an OLTP system is characterized by a high volume of small, frequent transactions that require fast response times and concurrent access by multiple users.	The access pattern of an OLAP system is characterized by fewer, larger, and more complex queries that require longer response times but provide greater analytical capabilities.
Data Model	OLTP systems typically use a normalized data model, where data is organized into multiple tables and relationships. Normalization reduces redundancy and ensures data consistency.	OLAP data models tend to be more denormalized. This should reduce the number of joins required and generally make it easier for an analyst to understand how to write their query.
Size	OLTPs tend to be smaller in terms of memory since they might only hold the current data and not historical changes.	OLAPs will be larger as they will store historical data as well as data from multiple systems.
Performance Needs	OLTPs need to have fast response times. Otherwise, end-users would be concerned that their tweet didn't go through	OLAP systems can get away with being a little slower. But if your dashboard is taking 10 minutes, DM me.

Recap: what is a table



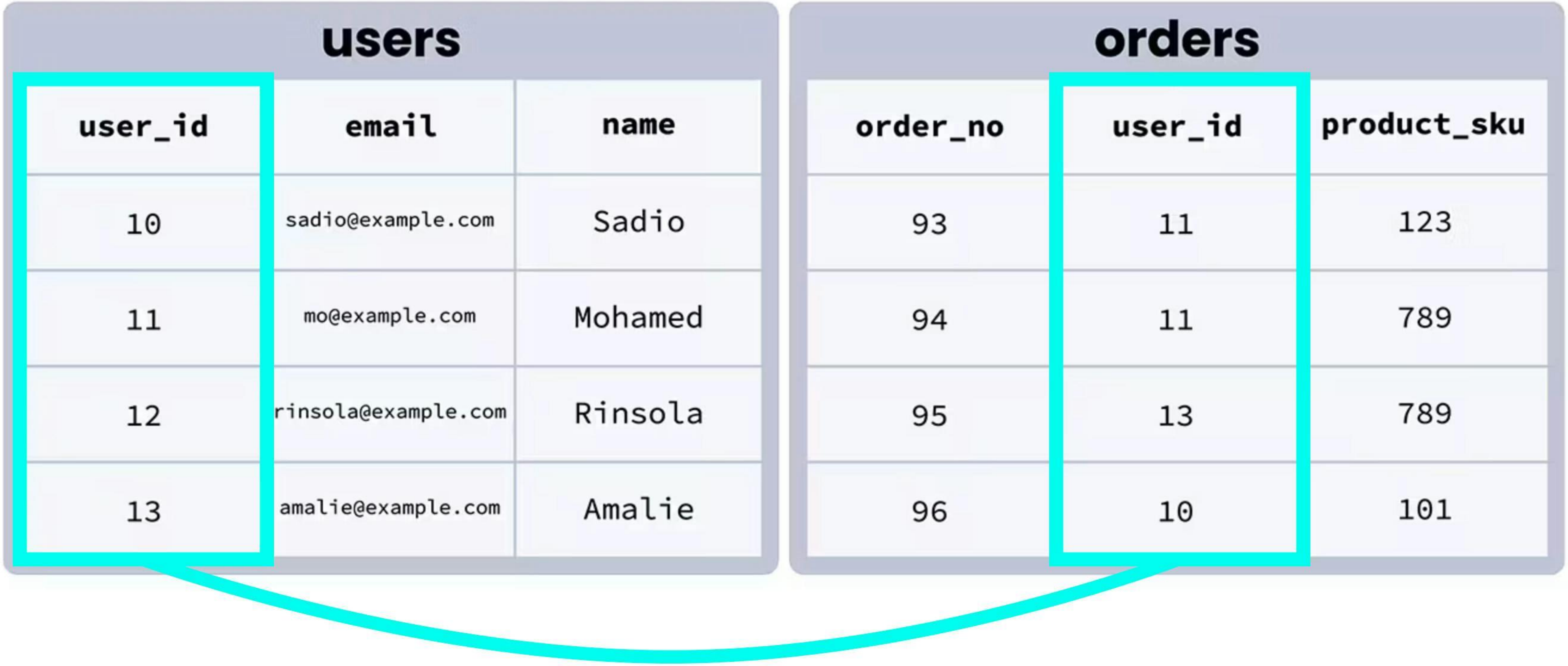
ID	NAME	AGE	COURSE
1	MINAL	22	COMPUTER SCIENCE
2	MRIDUL	21	CIVIL
3	MARAM	19	CHEMICAL
4	MOHAMMAD	24	ELECTRICAL

Recap: Primary key



Employee ID	SURNAME	GIVEN NAME	MIDDLE NAME
8001000000	Smith	Jennifer	Abad
8001000001	Smith	John Nhiel	Galvez
8001000002	Dela Cruz	RJ	Prachaya
8001000003	Reyes	Gab	Ugalino
8001000004	Doe	RJ	Mendoza
8001000005	Licauco	David	Galvez

Recap: Foreign keys



users

user_id	email	name
10	sadio@example.com	Sadio
11	mo@example.com	Mohamed
12	rinsola@example.com	Rinsola
13	amalie@example.com	Amalie

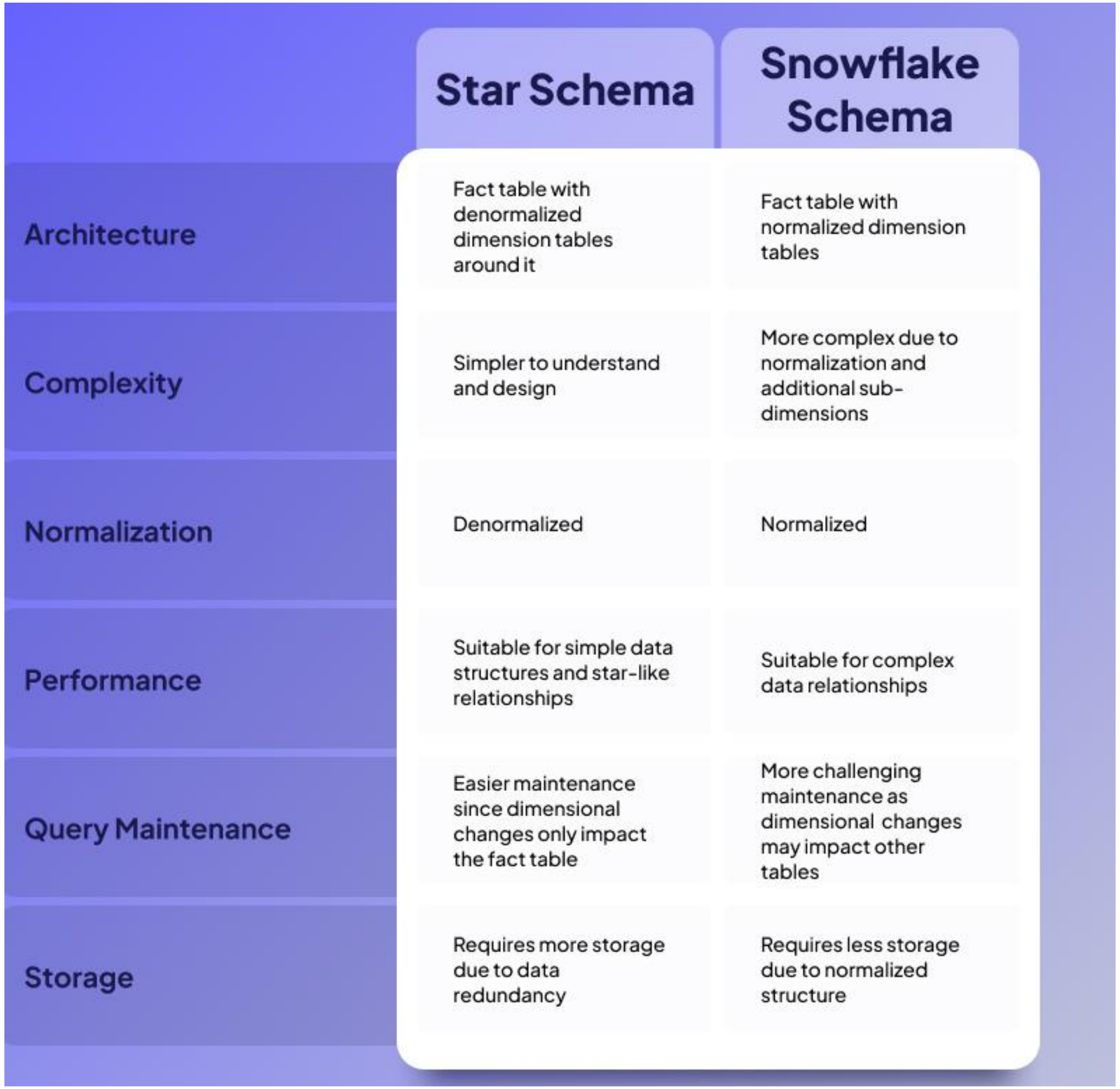
orders

order_no	user_id	product_sku
93	11	123
94	11	789
95	13	789
96	10	101

A row can only be added or updated in the **orders** table if the value in **orders.user_id** matches an existing user ID in the **users** table.

This type of **database rule** is called a **foreign key constraint**.

Star and Snowflake Schemas

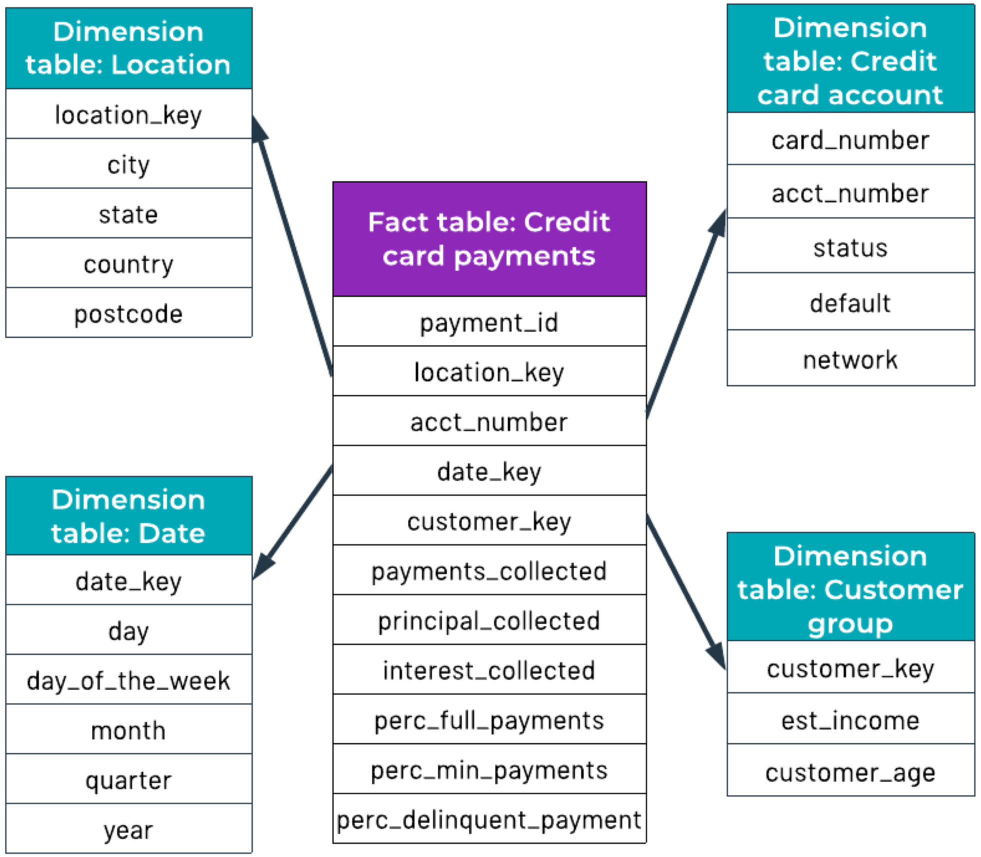


	Star Schema	Snowflake Schema
Architecture	Fact table with denormalized dimension tables around it	Fact table with normalized dimension tables
Complexity	Simpler to understand and design	More complex due to normalization and additional sub-dimensions
Normalization	Denormalized	Normalized
Performance	Suitable for simple data structures and star-like relationships	Suitable for complex data relationships
Query Maintenance	Easier maintenance since dimensional changes only impact the fact table	More challenging maintenance as dimensional changes may impact other tables
Storage	Requires more storage due to data redundancy	Requires less storage due to normalized structure

Star Schema: A database structure in which a central fact table is surrounded by dimension tables, resembling a star. The fact table contains measurable quantities and foreign keys from dimension tables that describe the data's context.

Snowflake Schema: An extension of the star schema where dimension tables are normalized into multiple related tables, resulting in a more complex database structure that resembles a snowflake with branches radiating from a central node.

Star and Snowflake Schemas Example



Dimension table: Location
location_key
city
state
country
postcode

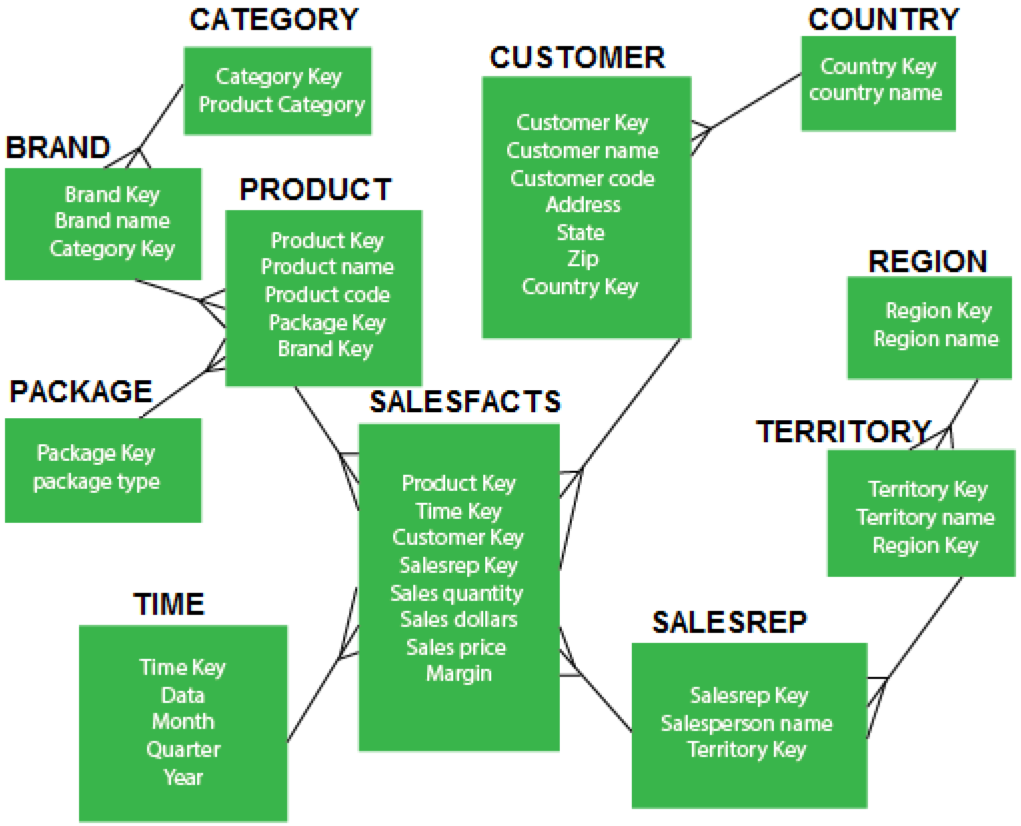
Dimension table: Date
date_key
day
day_of_the_week
month
quarter
year

Fact table: Credit card payments
payment_id
location_key
acct_number
date_key
customer_key
payments_collected
principal_collected
interest_collected
perc_full_payments
perc_min_payments
perc_delinquent_payment

Dimension table: Credit card account
card_number
acct_number
status
default
network

Dimension table: Customer group
customer_key
est_income
customer_age

Star schema



CATEGORY
Category Key
Product Category

BRAND
Brand Key
Brand name
Category Key

PRODUCT
Product Key
Product name
Product code
Package Key
Brand Key

PACKAGE
Package Key
package type

CUSTOMER
Customer Key
Customer name
Customer code
Address
State
Zip
Country Key

COUNTRY
Country Key
country name

REGION
Region Key
Region name

TERRITORY
Territory Key
Territory name
Region Key

SALESFACTS
Product Key
Time Key
Customer Key
Salesrep Key
Sales quantity
Sales dollars
Sales price
Margin

TIME
Time Key
Data
Month
Quarter
Year

SALESREP
Salesrep Key
Salesperson name
Territory Key

Snowflake Schema

Star and Snowflake Schemas Considerations

Star Schema

Widely used in data warehousing for fast data retrieval in Business Intelligence applications due to less join complexity.

Snowflake Schema

Suitable for environments where data integrity and detailed dimensional analysis are crucial.

Question to reflect on:

Why might an organisation prefer the simplicity of a star schema for reporting and querying purposes?

Things to think about:

Consider the trade-offs between the simplicity and performance of the star schema versus the normalization benefits of the snowflake schema, which might complicate data retrieval processes.

Decision Factors for Schema Selection

Query Performance

Star schemas typically offer faster query performance for large datasets due to fewer joins.

Data Integrity and Normalization

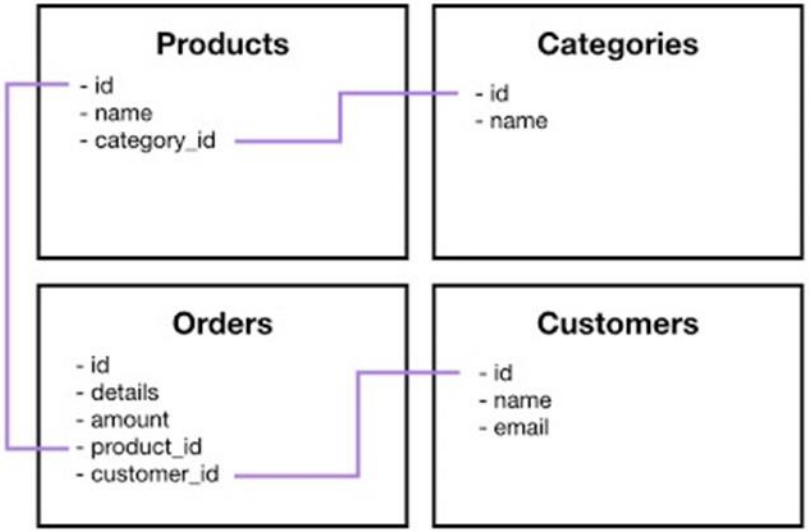
Snowflake schemas provide higher data integrity through normalization, which can simplify maintenance and reduce errors.

But what is normalisation?

Normalised vs Denormalised Data

Normalized

A schema design to store **non-redundant** and **consistent data**



Products
- id
- name
- category_id

Categories
- id
- name

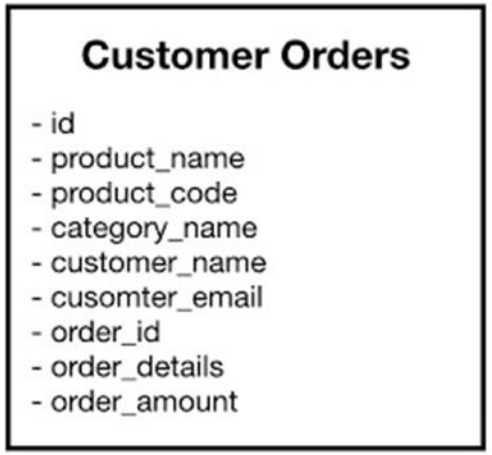
Orders
- id
- details
- amount
- product_id
- customer_id

Customers
- id
- name
- email

- Data Integrity is maintained
- Little to no redundant data
- Many tables
- Optimizes for storage of data

Denormalized

A schema that **combines data** so that **accessing data (querying) is fast**

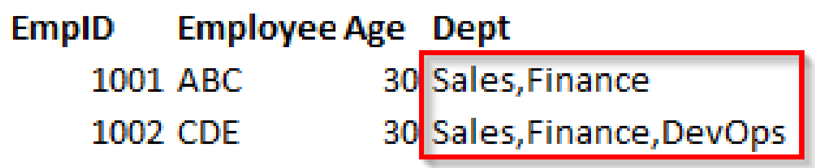


Customer Orders
- id
- product_name
- product_code
- category_name
- customer_name
- cusomter_email
- order_id
- order_details
- order_amount

- Data Integrity is not maintained
- Redundant data is common
- Fewer tables

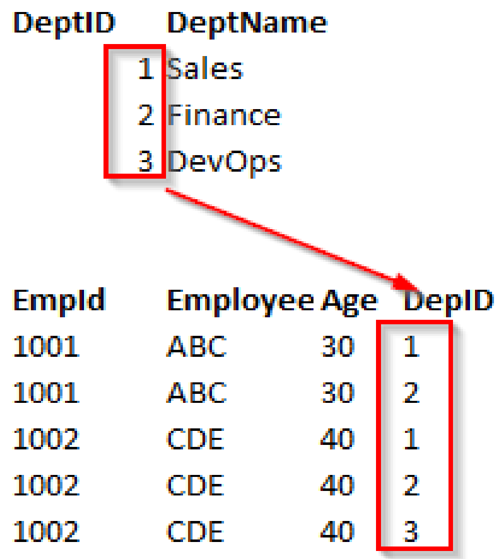
Normalised vs Denormalised Data

Consider the following **EMPLOYEE** table:



EmpID	Employee	Age	Dept
1001	ABC	30	Sales,Finance
1002	CDE	30	Sales,Finance,DevOps

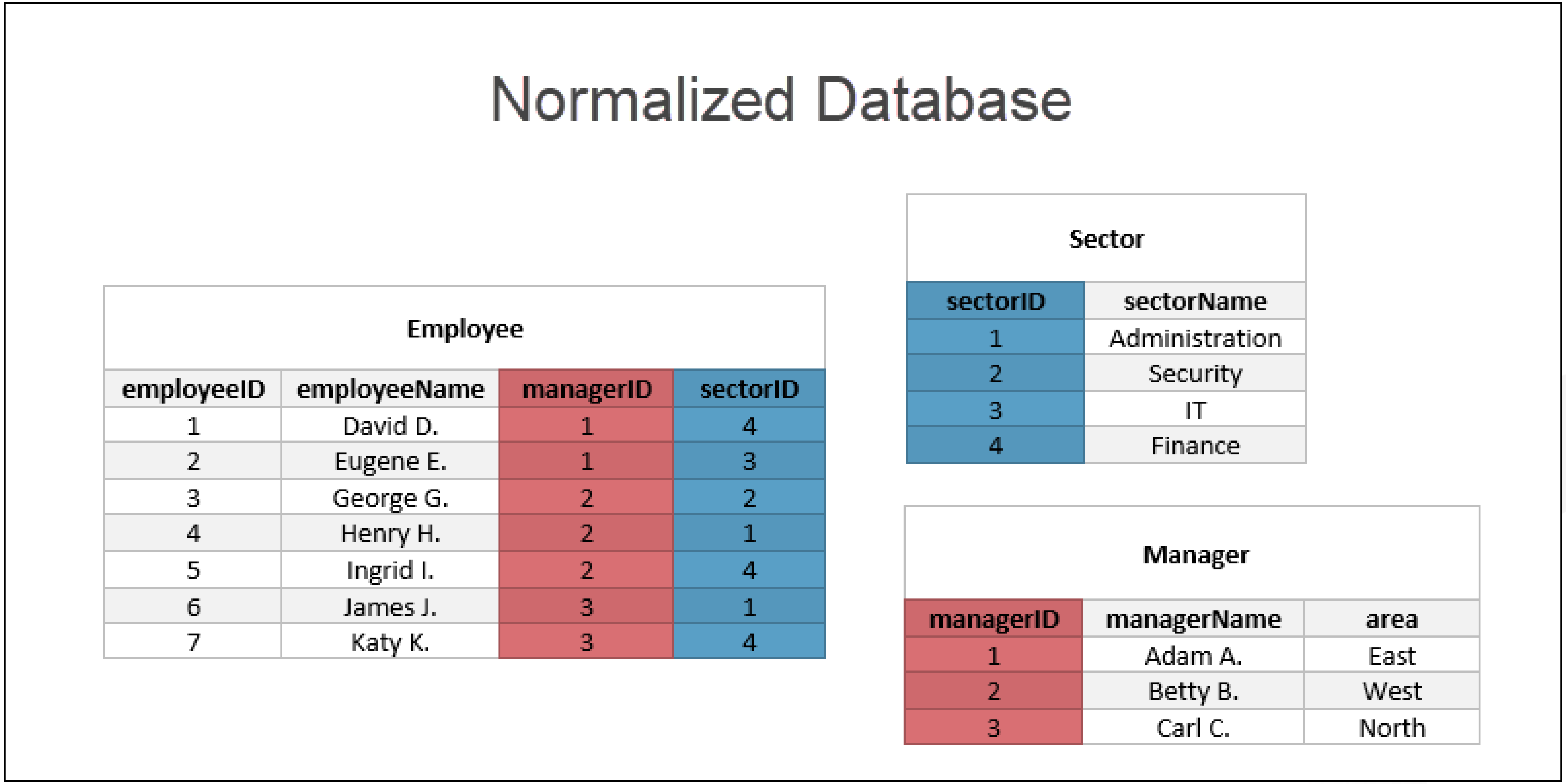
Now, after normalisation, the normalised tables **DEPT** and **EMPLOYEE** look like below:



DeptID	DeptName
1	Sales
2	Finance
3	DevOps

EmpId	Employee	Age	DepID
1001	ABC	30	1
1001	ABC	30	2
1002	CDE	40	1
1002	CDE	40	2
1002	CDE	40	3

A normalised database example



Normalized Database

Employee

employeeID	employeeName	managerID	sectorID
1	David D.	1	4
2	Eugene E.	1	3
3	George G.	2	2
4	Henry H.	2	1
5	Ingrid I.	2	4
6	James J.	3	1
7	Katy K.	3	4

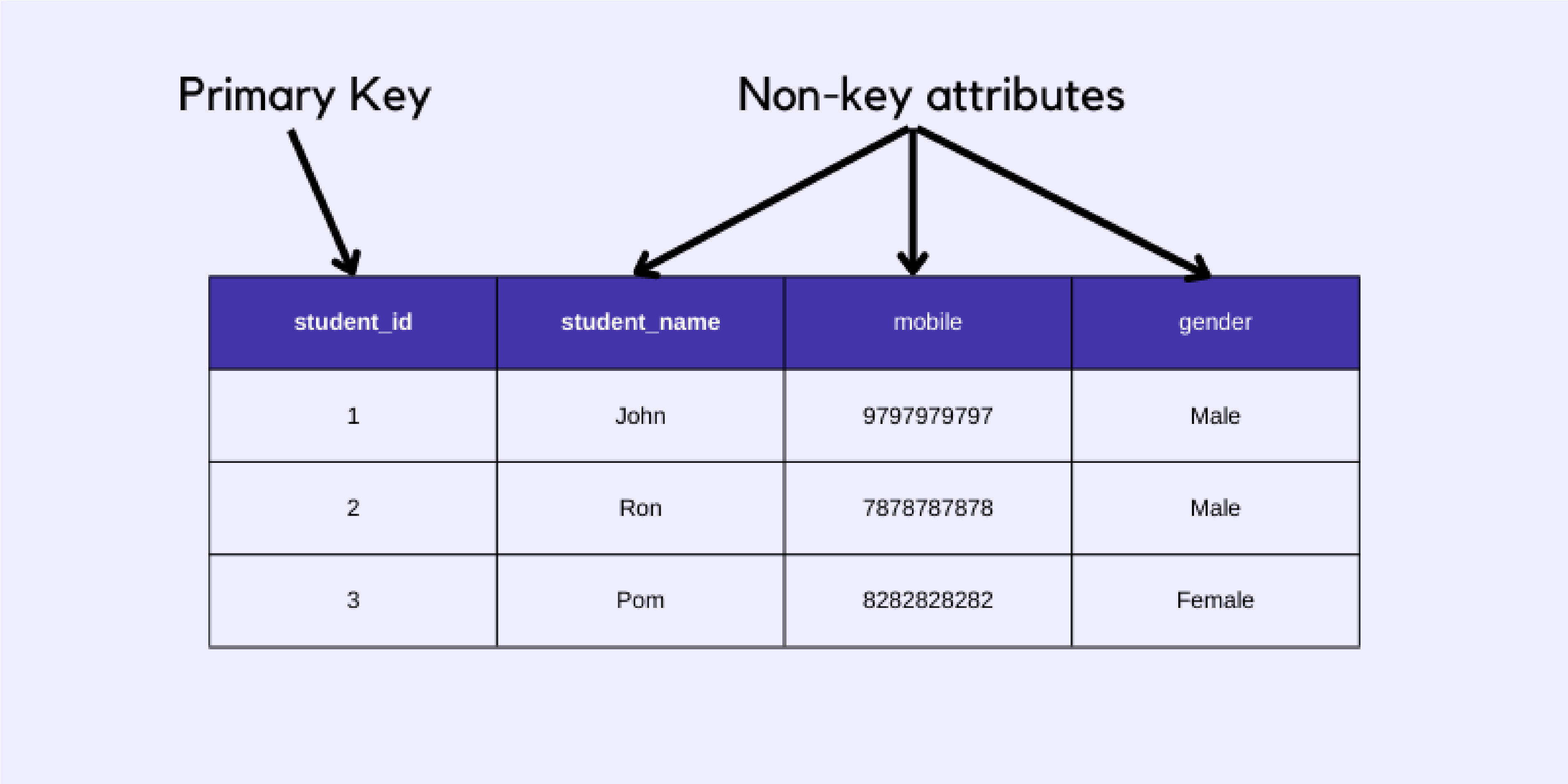
Sector

sectorID	sectorName
1	Administration
2	Security
3	IT
4	Finance

Manager

managerID	managerName	area
1	Adam A.	East
2	Betty B.	West
3	Carl C.	North

Non-key attributes



student_id	student_name	mobile	gender
1	John	9797979797	Male
2	Ron	7878787878	Male
3	Pom	8282828282	Female

Research exercise

In groups, research the following:

- 1st Normal Form
- 2nd Normal Form
- 3rd Normal Form
- Denormalized data

Create a short presentation explaining the differences between the normal forms and denormalized data, their pros and cons, and which one is the best one in your opinion?

Useful resource:
What is Database Normalization in SQL Server? (sqlshack.com)

Building Careers Through Education

Everybody Matters.

Stronger Together.

Student, Learner and Client Centric.

Trust and Respect.

Embrace Change.

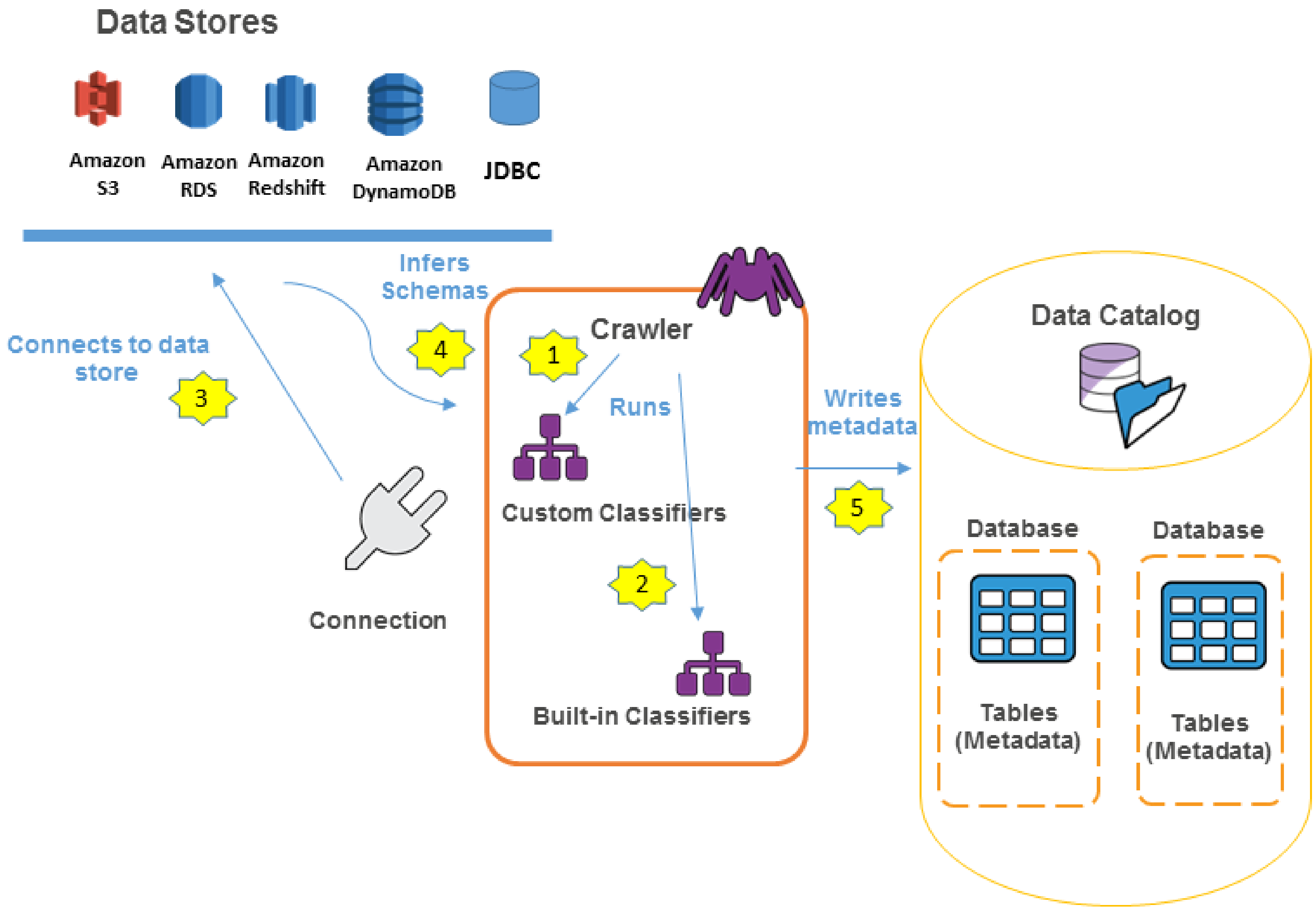
Break

What are metadata repositories/catalogs?

- Companies use a metadata repository to store and share information about data or metadata. Metadata repositories, once thought limited to databases or diagrams, have evolved sophisticated Data Architectures, driving businesses to transform the marketplace digitally.

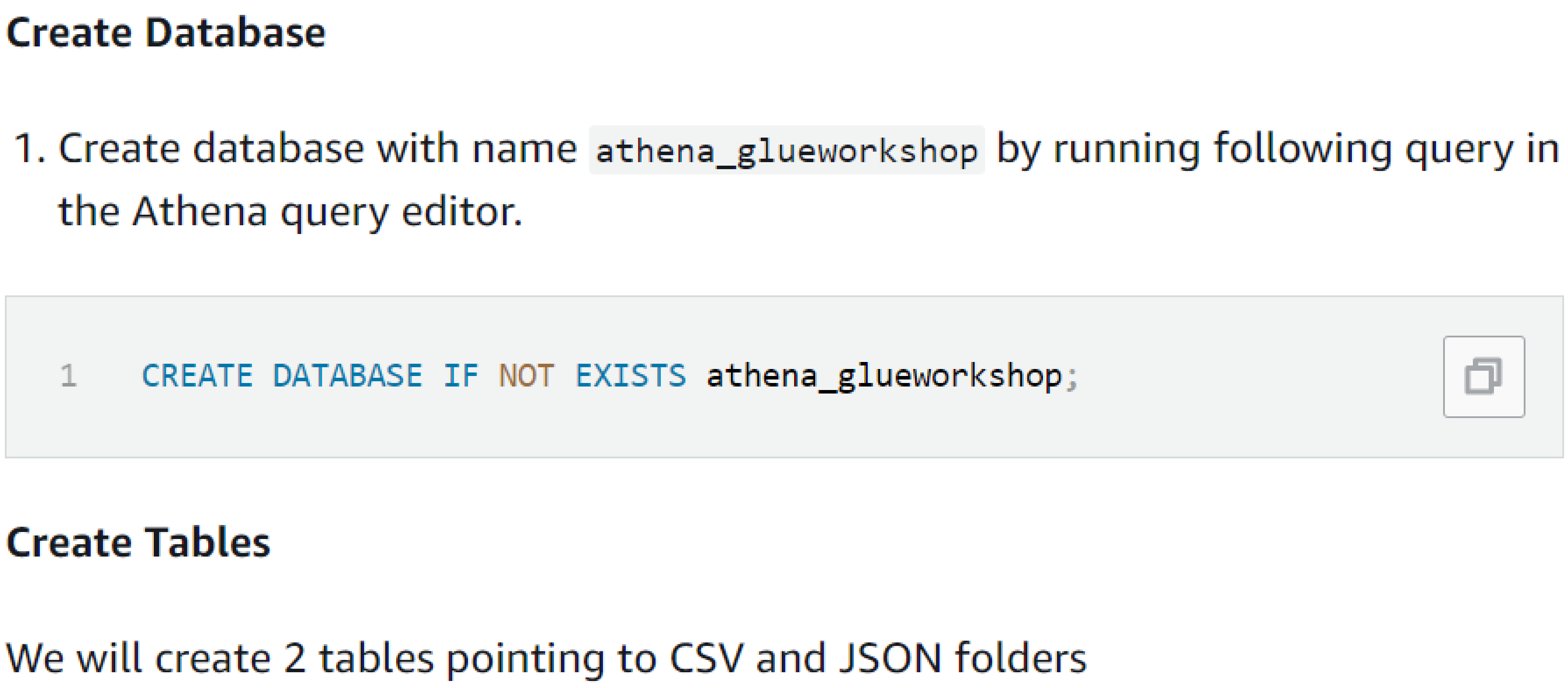
- Take the New South Wales (NSW) government’s Spatial Digital Twin, which went live in February 2020. NSW, an Australian state containing Sydney, envisioned a more efficient and better state infrastructure, including “major hospital upgrades.”
  - this digital twin, relies on a metadata repository to make tons of data faster to search and understand and to pull in even more data sets. From that metadata repository, Australians can digitally plan and build structures in real-time.

Metadata catalog example – AWS Glue



AWS Glue

[AWS Glue Immersion day (workshops.aws)]



Create Database

1. Create database with name `athena_glueworkshop` by running following query in the Athena query editor.

```
CREATE DATABASE IF NOT EXISTS athena_glueworkshop;
```

Create Tables

We will create 2 tables pointing to CSV and JSON folders

Exercises in scope (AWS)

- How to Start?
  - Self Paced Labs
  - Workshop Studio

Lab 01: Working with Glue Data Catalog:

Using AWS Console

Using AWS CLI

Optional: Using Athena

Stretch and extend exercises

- Lab 11: Working with Glue Databrew
- Glue Databrew Dataset
- Glue DataBrew Project
- Manage Glue DataBrew Recipe
- Run Glue DataBrew Job

Datamarts

Datamarts

A focused database designed to facilitate specific, department-level data analysis and reporting. Unlike data warehouses, which store comprehensive datasets, datamarts contain only relevant data.

Key Concepts

Star Schema

Smaller, more focused than entire data warehouses; designed to improve response time and data relevance for specific business units.

Snowflake Schema

Typically segregated by department, function, or subject, enhancing performance and end-user productivity.

Apache Iceberg

An open table format for huge analytic datasets, which allows for efficient schema evolution without downtime or performance penalties.

Key Concepts

Schema Evolution

The ability of a database schema to incrementally change without significantly disrupting existing data and queries.

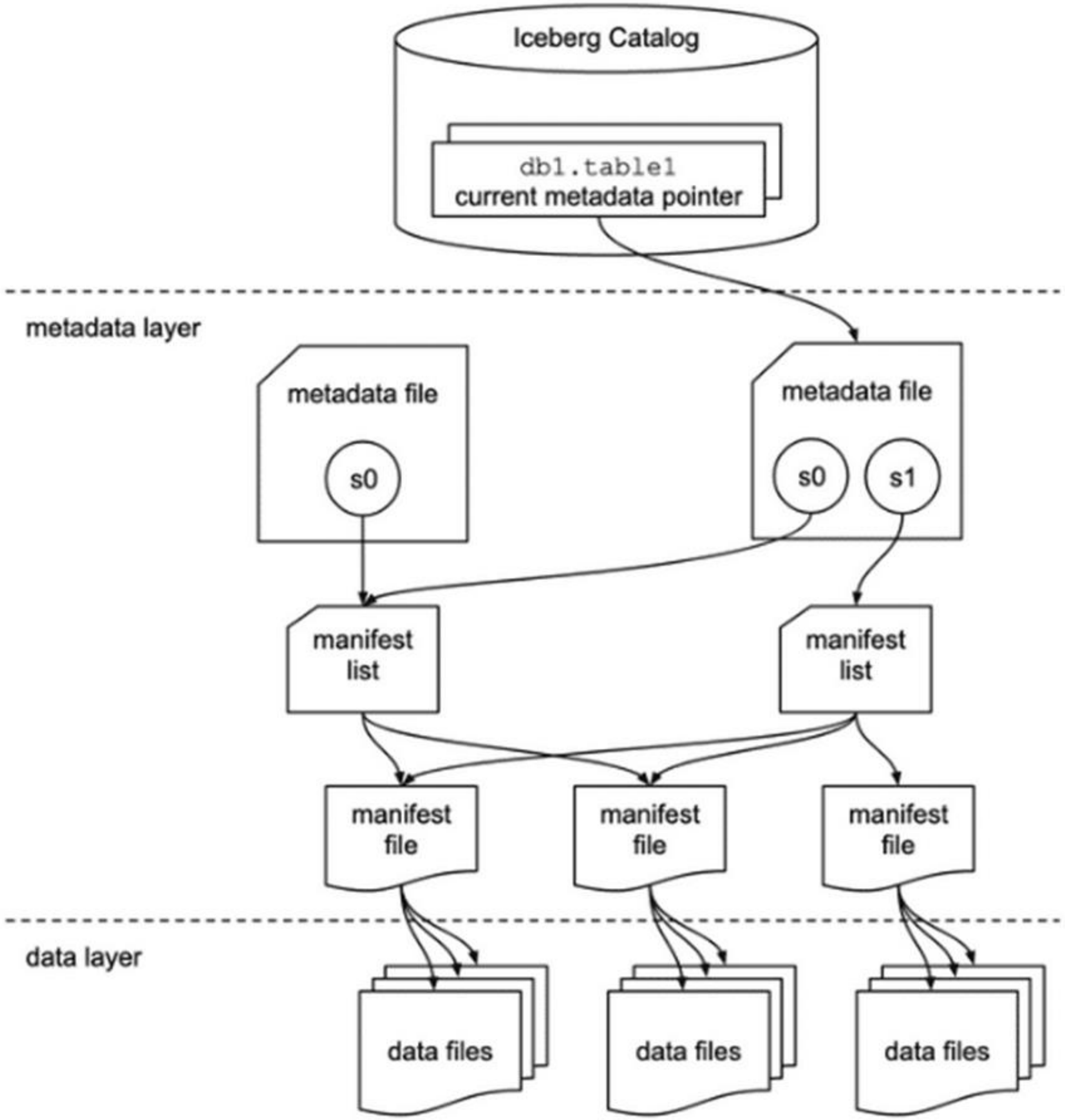
Flexibility and Scalability

Supports additions, deletions, and alterations of table schema on the fly, making it suitable for modern data lake requirements.

Use Cases

E-commerce Platforms: Manage and evolve product catalogues in their data lakes where schemas need to frequently adapt to changes in product attributes.

Apache Iceberg



Apache Iceberg

Practical Example

A streaming service uses Apache Iceberg to manage user data across multiple regions, seamlessly adapting schemas as new data types and sources are introduced.

Things to Think About

How does Apache Iceberg enhance data management strategies, particularly in environments requiring frequent updates to the data schema?

Building Careers Through Education

Everybody Matters.

Stronger Together.

Student, Learner and Client Centric.

Trust and Respect.

Embrace Change.

Break

BPP

SQL Practice with your Tutor

- **Foundational Language for Data Management**: SQL (Structured Query Language) is the standard language for relational database management and data manipulation. It allows users to create, retrieve, update, and delete database records efficiently.

- **Ubiquitous and Standardised**: SQL is supported by virtually all relational database systems like MySQL, PostgreSQL, Oracle, and SQL Server, making it a critical skill for data professionals.

- **Enhanced Data Retrieval**: SQL provides powerful but straightforward means to retrieve data from databases through SELECT queries, enabling complex analytics and reporting.

SQL Practice with your Tutor

Building Careers Through Education

Everybody Matters.
Stronger Together.
Student, Learner and Client Centric.
Trust and Respect.
Embrace Change.

- **Data Manipulation and Administration**: Beyond data retrieval, SQL is instrumental in structuring and managing large quantities of data, supporting operations like inserting new data, updating existing data, and performing transactional processes.

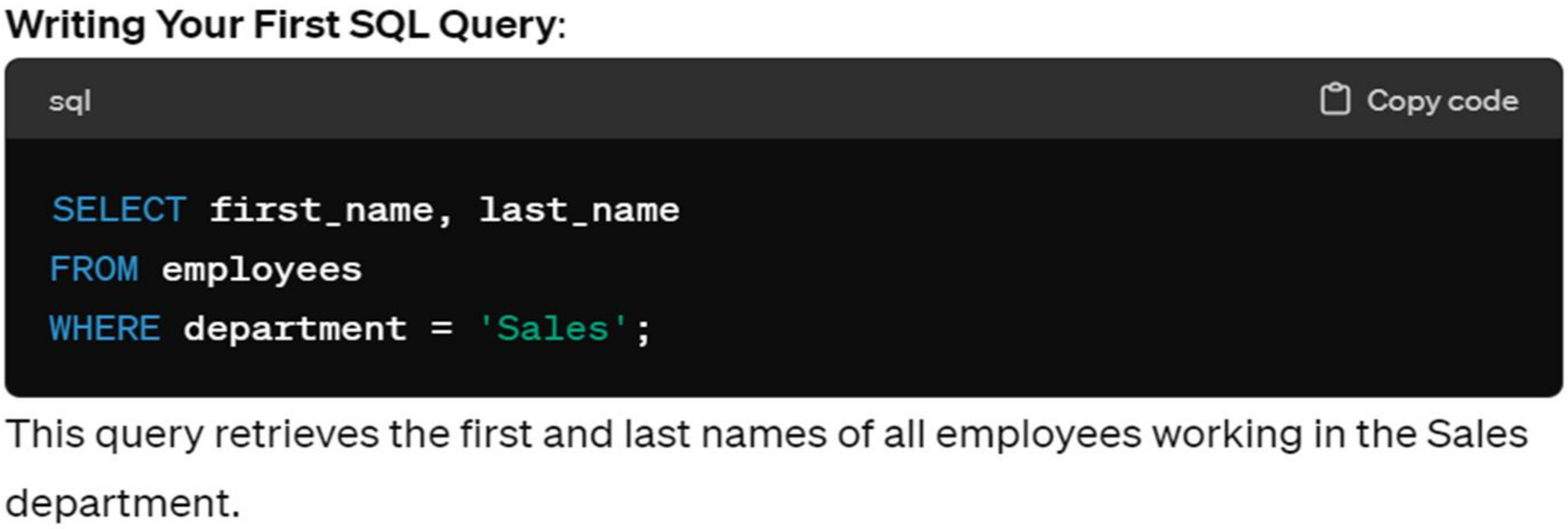
- **Integration with Other Technologies**: SQL databases easily integrate with numerous reporting and analytics tools, making SQL a pivotal part of data-driven decision-making processes in businesses.

- **Advanced Data Analysis and Business Intelligence**: SQL will be used to extract and analyze data, forming the basis for decision-making in business intelligence and data analytics topics.

BPP

Introduction to SQL

Basic Structure of SQL Queries: The fundamental structure of a SQL query includes a SELECT clause to specify the columns, a FROM clause to designate the tables, and an optional WHERE clause to filter records.



Writing Your First SQL Query:

```sql
SELECT first_name, last_name
FROM employees
WHERE department = 'Sales';
```

This query retrieves the first and last names of all employees working in the Sales department.

Practice with tutor

Building Careers Through Education
Everybody Matters.
Stronger Together.
Student, Learner and Client Centric.
Trust and Respect.
Embrace Change.

- SQL Tutorial (w3schools.com)
- SQLBolt - Learn SQL - SQL Lesson 1: SELECT queries 101

Focus on open-source vs proprietary

The Differences between Proprietary and Open Source Software

Open Software

(Linux Ubuntu, OpenOffice.org Write, GIMP)

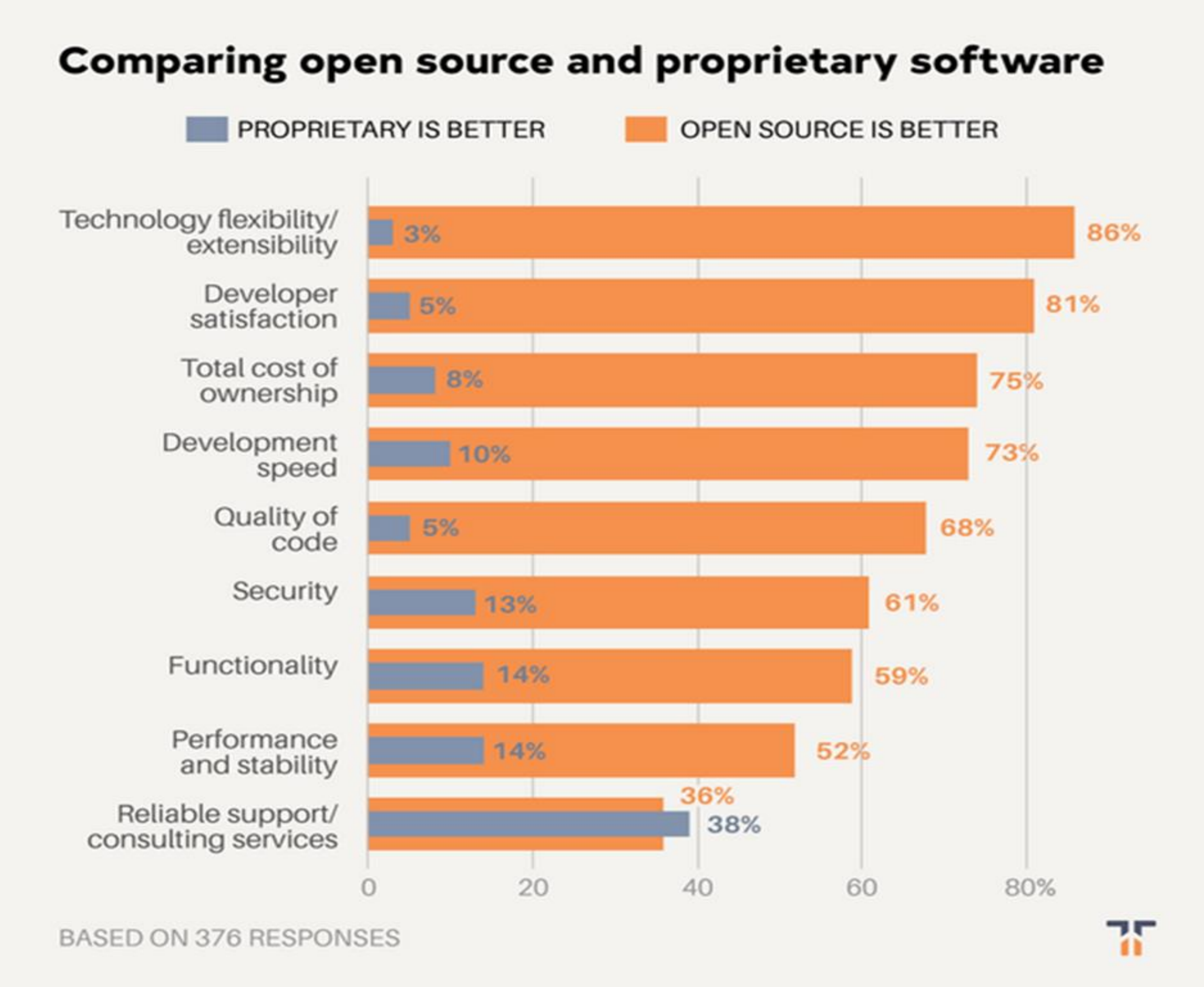
- Purchased with its source code
- User can get open software for free Of charge
- Users can modify the software
- Users can install software freely into any computer
- No one is responsible to the software
- Full support from vendor if anything happened to the software

Proprietary Software

(Windows Vista. Microsoft Word 2007. Adobe Photoshop CS3)

- Purchased without its source code
- User must pay to get the proprietary software
- Users Cannot modify the software
- User must have a license from vendor before install into computer
- Full support from vendor if anything happened to the software

Focus on open-source vs proprietary



Building Careers Through Education

Everybody Matters.

Stronger Together.

Student, Learner and Client Centric.

Trust and Respect.

Embrace Change.

[3 charts that show how open source developers think | Opensource.com]

BPP

Community support

Building Careers Through Education
Everybody Matters.
Stronger Together.
Student, Learner and Client Centric.
Trust and Respect.
Embrace Change.

- Open Source: Benefits from a broad community for troubleshooting, with varied levels of volunteer or commercial support available.
- Proprietary Software: Typically offers structured support via paid contracts, providing guaranteed assistance levels.

Discussion Points:

- Consider how the need for control vs. support may influence the choice between open source and proprietary systems.
- Reflect on how these factors impact your organisation's long-term tech strategy.

Overview of post-webinar e-learning

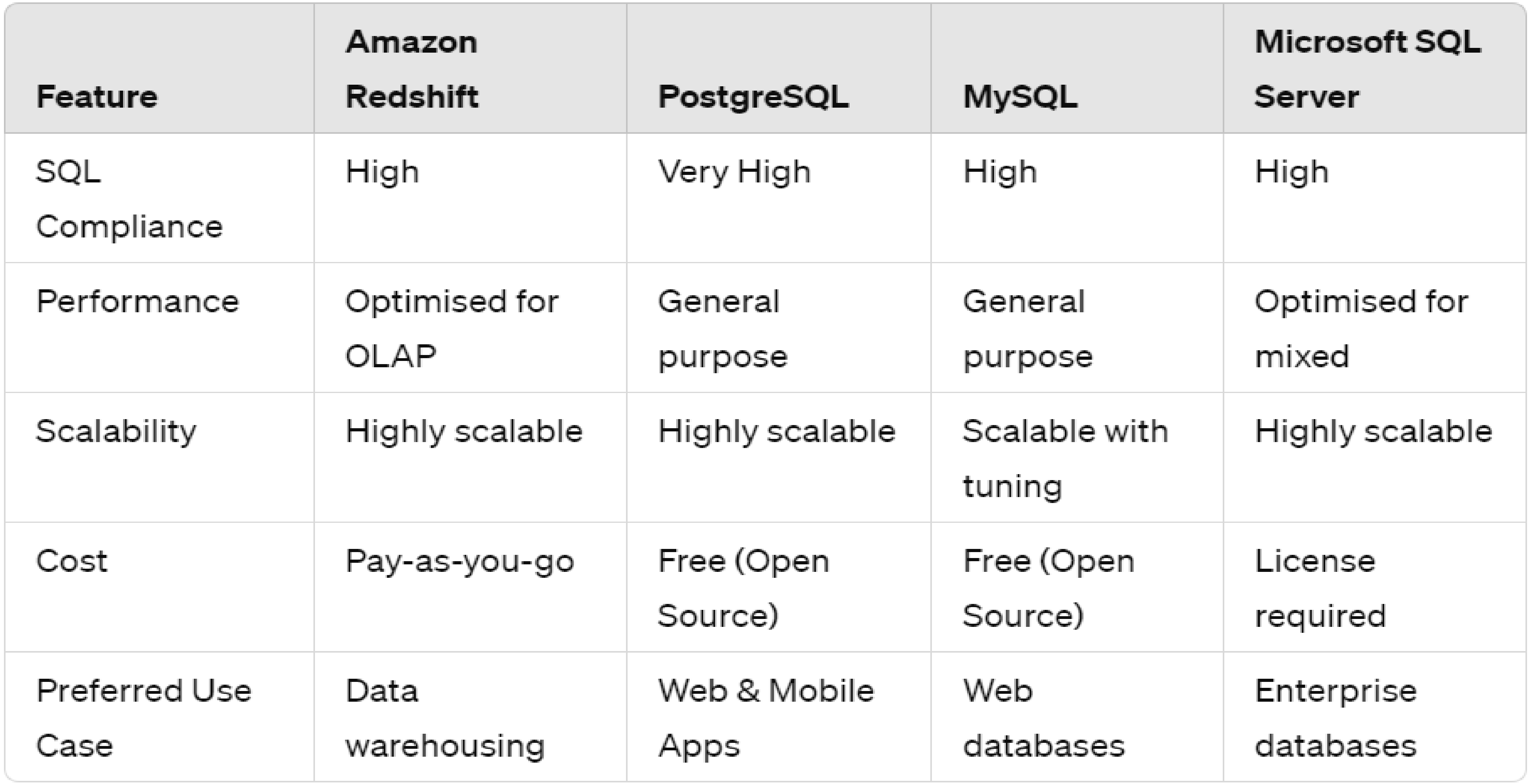
Extend and stretch activities on the Hub:

- **Introduction to Snowflake** (software, do not confuse with snowflake schema ☺)
- Discuss Snowflake's capabilities in cloud data warehousing, highlighting its architecture that supports seamless data scalability and integration.
- **Steps for Integrating a New Data Source**:
  - Identify the data source and ensure compatibility with Snowflake's supported formats and data ingestion methods.
  - Establish a connection to the data source using Snowflake’s connectors or via standard API integrations.

Research for your learning journal

Building Careers Through Education

Explore how each database handles SQL standards differently and the implications for application development and database administration. Consider the trade-offs in performance, scalability, and cost when choosing a database system.



Feature	Amazon Redshift	PostgreSQL	MySQL	Microsoft SQL Server
SQL Compliance	High	Very High	High	High
Performance	Optimised for OLAP	General purpose	General purpose	Optimised for mixed
Scalability	Highly scalable	Highly scalable	Scalable with tuning	Highly scalable
Cost	Pay-as-you-go	Free (Open Source)	Free (Open Source)	License required
Preferred Use Case	Data warehousing	Web & Mobile Apps	Web databases	Enterprise databases

BPP

Thank you

Do you have any questions, comments, or feedback?