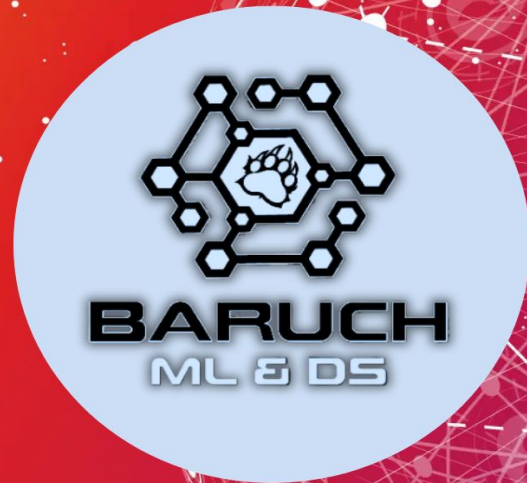


# Introduction to Databases and



# What is A Database

A database is an organized collection of data that can be easily accessed, managed and updated.

Relational databases are structured systems that store data in tables with rows and columns, allowing easy access and relationships between different data sets.



Data is organized into tables (relations)



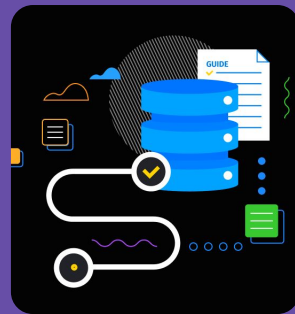
Each table contains rows (records) and columns (fields)



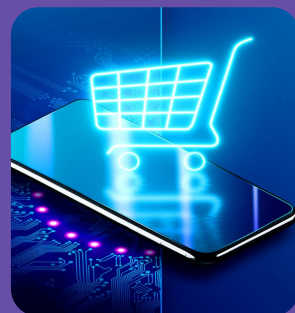
Primary key uniquely identifies each row



Ensures data integrity and supports complex queries



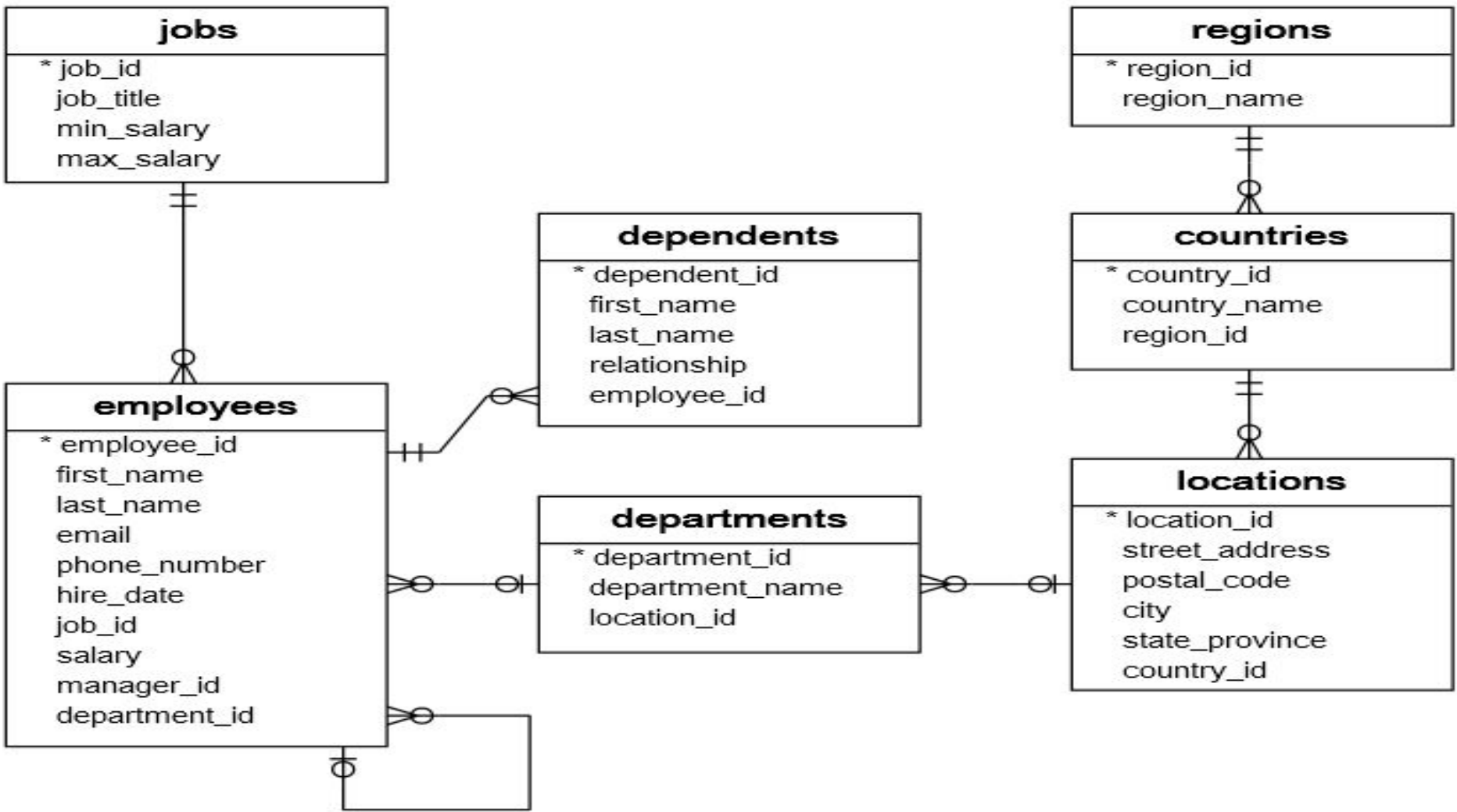
Foreign key links rows across tables



Commonly used in banking, websites, healthcare, and commerce

# Database Schema Diagram

- Provide a **blueprint** for how data is stored and related
  - Show how **tables** are organized and connected
  - Help in **designing, understanding, and maintaining** databases
  - Often used by developers, analysts, and database administrators
- **Tables:** Represented as boxes; each box is a database table
  - **Columns:** Listed inside each table; define the data fields
  - **Primary Keys:** Uniquely identify each row in a table (usually underlined or marked with "PK")
  - **Foreign Keys:** Connect tables by referencing primary keys in other tables (marked as "FK")
  - **Relationships:** Lines or arrows between tables showing how data is linked
  - **Cardinality:** Indicates how many rows in one table relate to rows in another (e.g., one-to-many)



# History of SQL

- **1970:** IBM's **Edgar F. Codd** proposed the **relational database model**.
- **Early 1970s:** IBM developed **System R**, the first prototype of a relational database.
- **1974:** IBM created **SEQUEL** (later renamed **SQL**) as a language to query relational data.
- **1981:** IBM released **SQL/DS**, an early commercial SQL-based product.
- **1983:** IBM launched **DB2**, a full relational database system using SQL.
- **1986:** SQL became an **ANSI standard**.



1889



1890



1910



1924

IBM IBM

1946

1956



1967



1972



2018

# Role of SQL

- **SQL (Structured Query Language)** is used to communicate with **relational databases**.
- Allows users to **create**, **read**, **update**, and **delete** data — known as **CRUD operations**.
- Widely used in **Data Science and Analytics** to **explore**, **filter**, **aggregate**, and **visualize** large datasets.
- Ensures **consistency**, **security**, and **controlled access** to data.

# Basic SQL Functions

## SELECT

**Use:** Specifies which columns to retrieve from a table.

**Parameters:** Column names, \* for all columns, or expressions (e.g., `COUNT(*)`, `name`, `salary * 1.1`).

## GROUP BY

**Use:** Groups rows that share the same values in specified columns.

**Parameters:** Column name(s) to group by (e.g., `department_id`, `region`).

## FROM

**Use:** Indicates the table(s) to query data from.

**Parameters:** Table name(s), with optional aliases.

## HAVING

**Use:** Filters grouped records; works like `WHERE` but for aggregated results.

**Parameters:** Conditions on aggregate functions (e.g., `HAVING COUNT(*) > 5`).

## WHERE

**Use:** Filters rows based on a condition.

**Parameters:** Logical expressions (e.g., `salary > 50000`, `department = 'Sales'`).

## ORDER BY

**Use:** Sorts the result set by one or more columns.

**Parameters:** Column names or expressions, with optional `ASC` (default) or `DESC` (e.g., `ORDER BY salary DESC`).

**Aggregates (MAX, MIN, COUNT, SUM, AVG)** **Use:** Perform calculations on values to return a result.

**Use:** Restricts the number of rows returned. (e.g., `LIMIT 10`)