Revising SQL queries

Handouts on Join, Select, and Projection in SQL

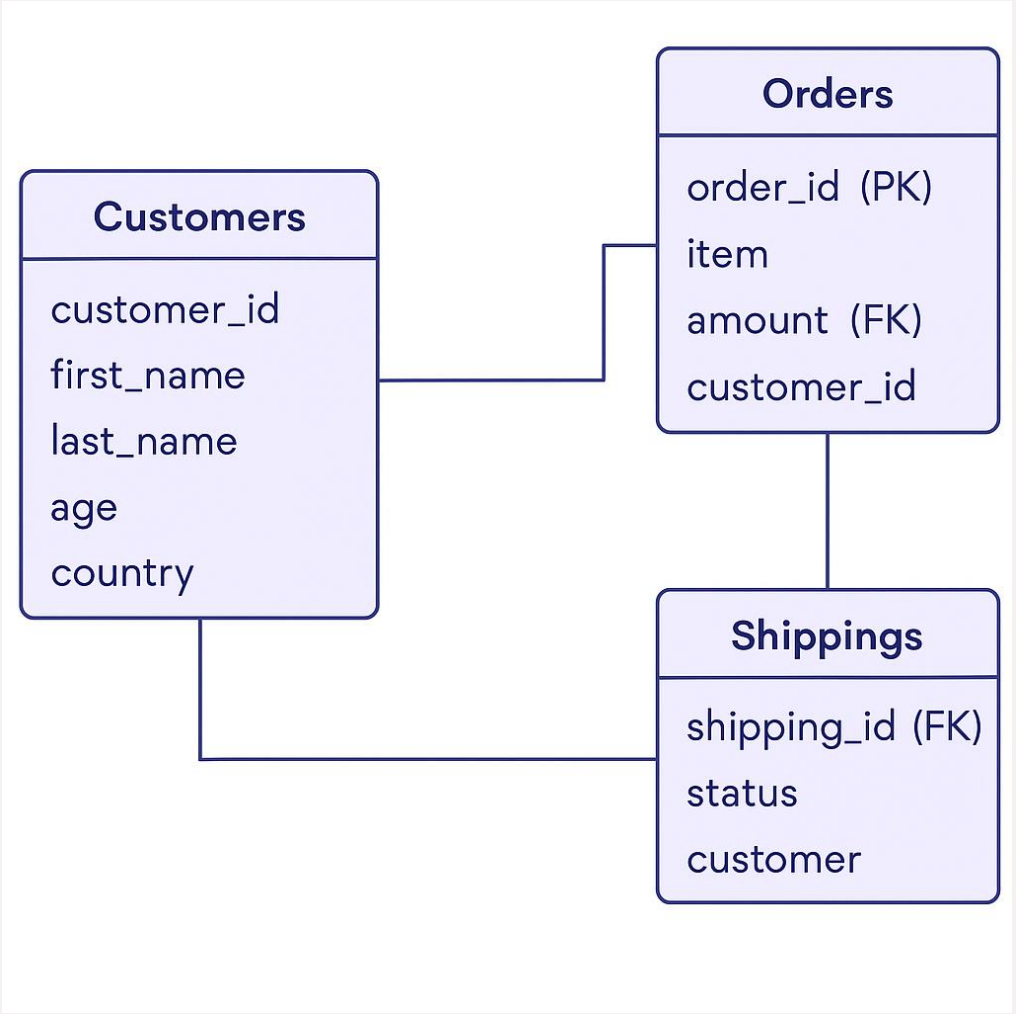
Data Mining and Text Analytics

Postgraduate Programme in Artificial Intelligence for Business and Society

Professor Alessandro Bruno

SQL – Structured Query Language

- Let's revise some SQL functions and the results when applied on an SQL DB.
- Figure out a relational DB represented by the ER diagram on the righthand side.



Processing Data with SQL

Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

Orders

order_id	item	amount	customer_id
1	Keyboard	400	4
2	Mouse	300	4
3	Monitor	12000	3
4	Keyboard	400	1
5	Mousepad	250	2

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Creating a TABLE with SQL

- CREATE TABLE Customers (
  customer_id INT PRIMARY KEY,
  first_name VARCHAR(50),
  last_name VARCHAR(50),
  age INT,
  country VARCHAR(50));

-- Insert sample data

INSERT INTO Customers (customer_id, first_name, last_name, age, country)

VALUES

(1, 'John', 'Doe', 31, 'USA'),

(2, 'Robert', 'Luna', 22, 'USA'),

(3, 'David', 'Robinson', 22, 'UK'),

(4, 'John', 'Reinhardt', 25, 'UK'),

(5, 'Betty', 'Doe', 28, 'UAE');

Running SQL

- Let's run the SQL Compiler at the following links:
- https://www.programiz.com/sql/online-compiler
- https://sqliteonline.com/
- **SELECT - Example:**

SELECT first_name, last_name FROM Customers WHERE country = "USA" ORDER BY age DESC;

SELECT

first_name	last_name
John	Doe
Robert	Luna

JOIN

- SELECT c.first_name, o.item, o.amount
- FROM Customers c
- JOIN Orders o ON c.customer_id = o.customer_id;

This is an **INNER JOIN** (default type if you just say `JOIN`).

It means:
"Only return rows where a customer in the **Customers** table has a matching `customer_id` in the **Orders** table."

JOIN (INNER)

first_name	item	amount
John	Keyboard	400
John	Mouse	300
David	Monitor	12000
John	Keyboard	400
Robert	Mousepad	250

TYPES of JOIN

- **`INNER JOIN`** → only matching rows
- **`LEFT JOIN`** → all rows from left table, matching ones from right
- **`RIGHT JOIN`** → all rows from right table
- **`FULL JOIN`** → all rows from both sides

LEFT JOIN

```
SELECT c.first_name, c.last_name, o.item, o.amount
FROM Customers c
LEFT JOIN Orders o ON c.customer_id = o.customer_id;
```

Meaning: Return **all customers**, even those who have **no orders**. If they haven't ordered anything, the `item` and `amount` will be **NULL**.

Results - LEFT JOIN

first_name	last_name	item	amount
John	Doe	Keyboard	400
Robert	Luna	Mousepad	250
David	Robinson	Monitor	12000
John	Reinhardt	Keyboard	400
John	Reinhardt	Mouse	300
Betty	Doe		

RIGHT JOIN

```
SELECT c.first_name, c.last_name, o.item, o.amount
FROM Customers c
RIGHT JOIN Orders o ON c.customer_id = o.customer_id;
```

Meaning:

- Return all orders, even if there's no matching customer.
- If the customer doesn't exist in the Customers table, customer fields will be NULL.

FULL JOIN

```
SELECT c.first_name, c.last_name, o.item, o.amount
FROM Customers c
FULL JOIN Orders o ON c.customer_id = o.customer_id;
```

Meaning:

Return all customers and all orders, matching when possible.

If a customer has no orders → order info is NULL.

If an order has no customer → customer info is NULL.

PROJECTION

- A **Projection** selects only certain columns (attributes) from a table.
  Example: Display only customer names and countries.

```
SELECT first_name, last_name, country
FROM Customers;
```

PROJECTION

- Check out the result down below

first_name	last_name	country
John	Doe	USA
Robert	Luna	USA
David	Robinson	UK
John	Reinhardt	UK
Betty	Doe	UAE