Practicum 02_08

Databases

Authors:	Noah Catalán Rosell Diego Quiroga Bausa

Index

SQL Tutorial

In this section, we summarize the key concepts and techniques learned from the W3Schools SQL tutorial, focusing on advanced query functionalities and their practical applications:

SQL Joins

Joins are used to combine rows from two or more tables based on a related column. They enable us to retrieve data that is spread across multiple tables.

Inner Join

An Inner Join retrieves records that have matching values in both tables. It is the most commonly used type of join when we want data only where a match exists.

Left Join

A Left Join (or Left Outer Join) returns all records from the left table and the matched records from the right table. If there is no match, NULL values are returned for columns from the right table.

Right Join

A Right Join (or Right Outer Join) is similar to a Left Join, but it returns all records from the right table and the matched records from the left table, filling unmatched rows with NULL values.

Full Join

A Full Join (or Full Outer Join) retrieves all records from both tables, matching them where possible. If there is no match, it fills missing columns with NULL values.

Self Join

A Self Join is a join where a table is joined with itself. This is useful for comparing rows within the same table, often using aliases to differentiate table instances.

Union

The UNION operator is used to combine the results of two or more SELECT statements into a single result set. Duplicate rows are removed by default.

Group By

The GROUP BY statement groups rows with the same values into summary rows. It is often used with aggregate functions like COUNT, SUM, AVG, MAX, or MIN.

Having

The HAVING clause allows us to filter records after grouping. It is similar to WHERE but operates on aggregated data.

Exists

The EXISTS operator is used to check if a subquery returns any rows. It is often employed to verify the existence of data that satisfies specific conditions.

Any

The ANY operator compares a value to a set of values returned by a subquery. It evaluates to true if any of the subquery results meet the condition.

All

The ALL operator compares a value to all values in a set returned by a subquery. It evaluates to true only if all values satisfy the condition.

Select Into

The SELECT INTO statement copies data from one table into a new table. It is useful for creating backup tables or archiving data.

Insert Into Select

The INSERT INTO SELECT statement copies data from one table to another existing table. It is useful for populating tables with data from other sources.

Case

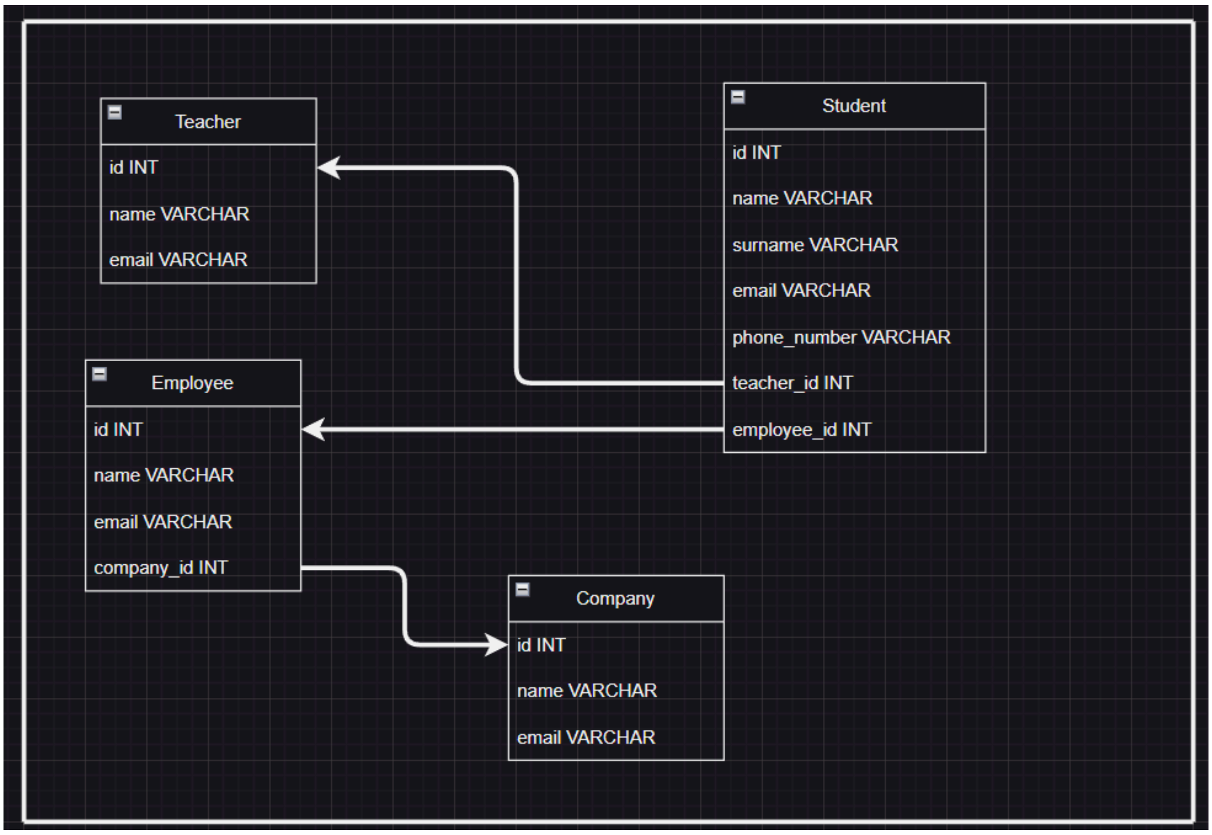
The CASE statement provides a way to add conditional logic to queries. It allows us to return different values based on specified conditions.

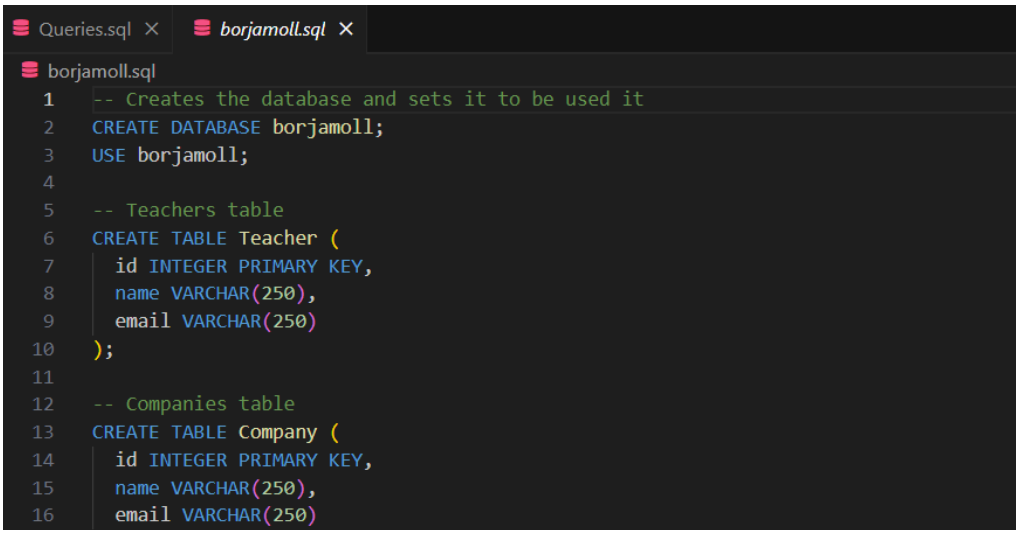
Null Functions

NULL functions, such as ISNULL or COALESCE, handle NULL values by replacing them with specified alternatives or determining if a value is NULL.

Database creation

For this assignment, we will use the *borjamoll* database, which was created in a previous assignment. This database includes the necessary tables and data to execute and test the queries required for this task.



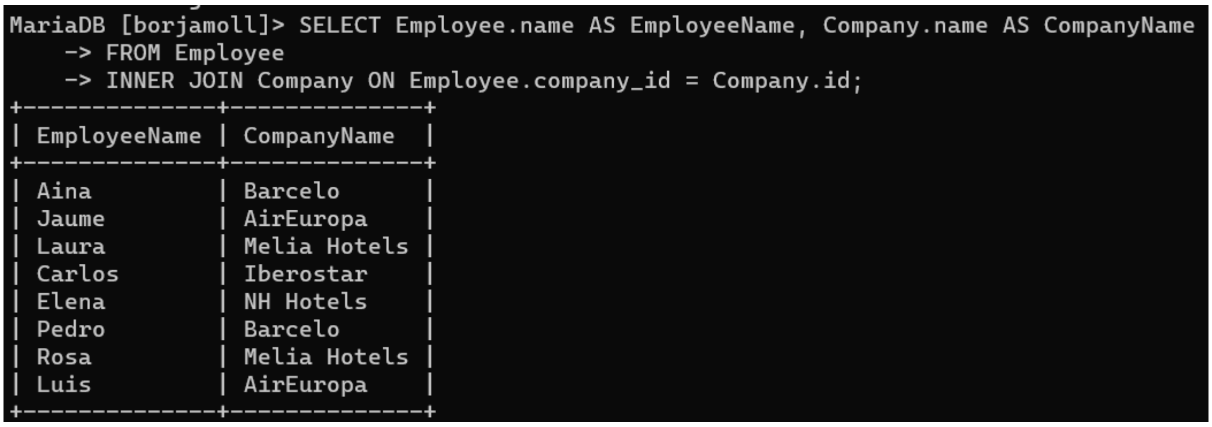


```sql
-- Creates the database and sets it to be used it
CREATE DATABASE borjamoll;
USE borjamoll;

-- Teachers table
CREATE TABLE Teacher (
  id INTEGER PRIMARY KEY,
  name VARCHAR(250),
  email VARCHAR(250)
);

-- Companies table
CREATE TABLE Company (
  id INTEGER PRIMARY KEY,
  name VARCHAR(250),
  email VARCHAR(250)
```

Database queries

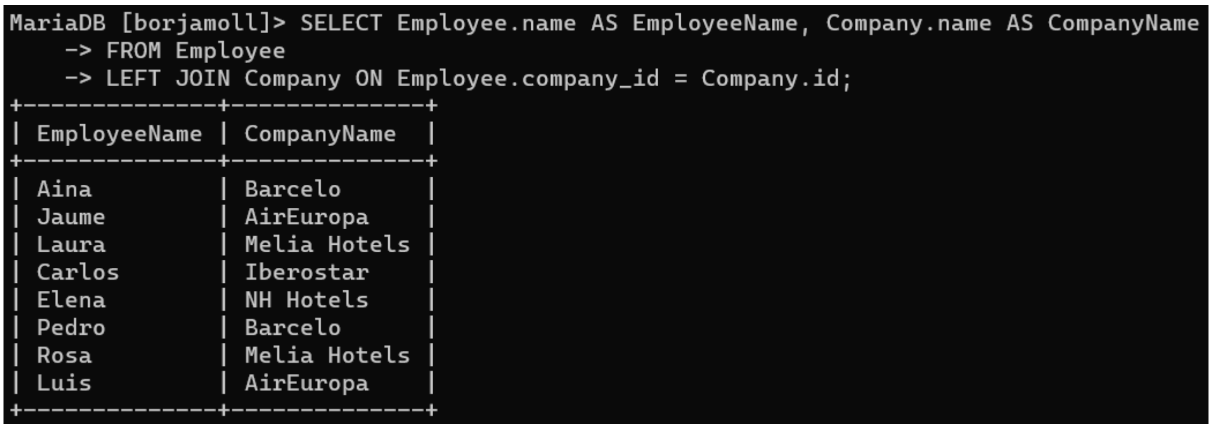
We will submit an SQL file containing all the queries with its appropriate comments. Also we will include a few screenshots of the output of those queries. We used Beekeeper Studio, phpMyAdmin, PostgreSQL and MariaDB.

Inner Join



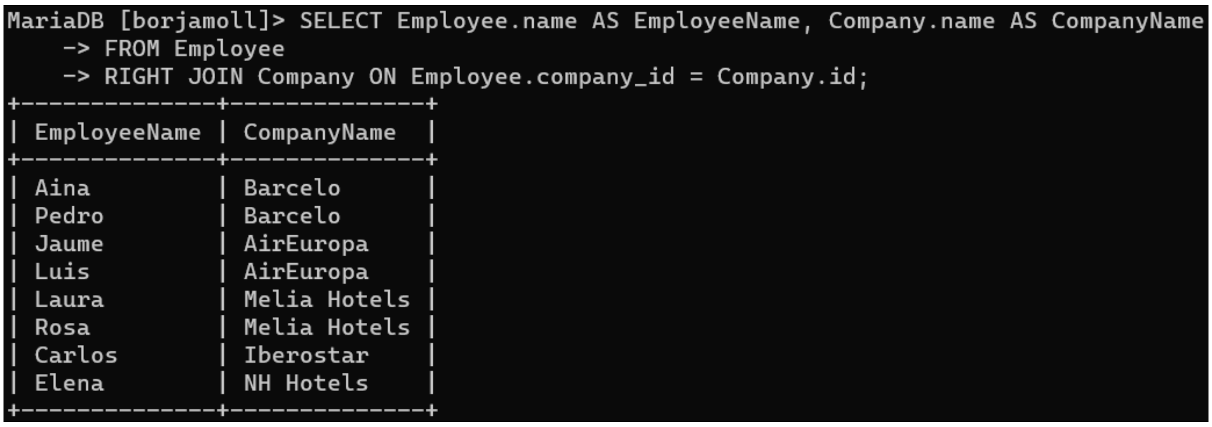
```
MariaDB [borjamoll]> SELECT Employee.name AS EmployeeName, Company.name AS CompanyName
    -> FROM Employee
    -> INNER JOIN Company ON Employee.company_id = Company.id;
+--------------+--------------+
| EmployeeName | CompanyName  |
+--------------+--------------+
| Aina         | Barcelo      |
| Jaume        | AirEuropa    |
| Laura        | Melia Hotels |
| Carlos       | Iberostar    |
| Elena        | NH Hotels    |
| Pedro        | Barcelo      |
| Rosa         | Melia Hotels |
| Luis         | AirEuropa    |
+--------------+--------------+
```

Left Join



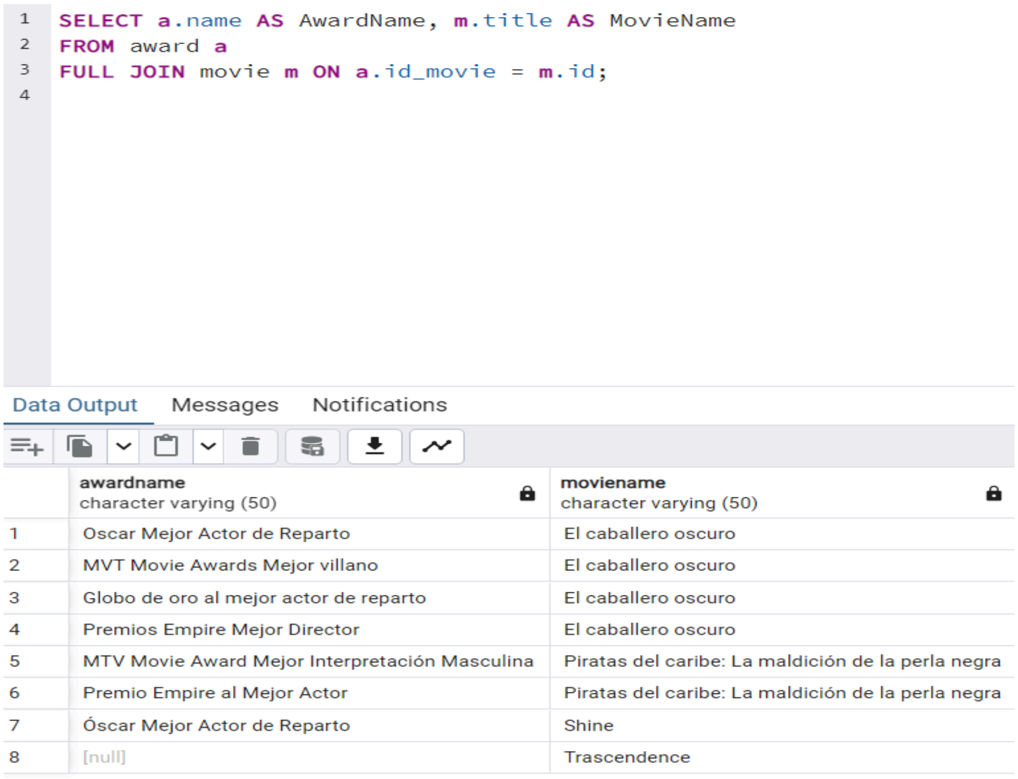
```
MariaDB [borjamoll]> SELECT Employee.name AS EmployeeName, Company.name AS CompanyName
    -> FROM Employee
    -> LEFT JOIN Company ON Employee.company_id = Company.id;
+--------------+--------------+
| EmployeeName | CompanyName  |
+--------------+--------------+
| Aina         | Barcelo      |
| Jaume        | AirEuropa    |
| Laura        | Melia Hotels |
| Carlos       | Iberostar    |
| Elena        | NH Hotels    |
| Pedro        | Barcelo      |
| Rosa         | Melia Hotels |
| Luis         | AirEuropa    |
+--------------+--------------+
```

Right Join



```
MariaDB [borjamoll]> SELECT Employee.name AS EmployeeName, Company.name AS CompanyName
    -> FROM Employee
    -> RIGHT JOIN Company ON Employee.company_id = Company.id;
+--------------+--------------+
| EmployeeName | CompanyName  |
+--------------+--------------+
| Aina         | Barcelo      |
| Pedro        | Barcelo      |
| Jaume        | AirEuropa    |
| Luis         | AirEuropa    |
| Laura        | Melia Hotels |
| Rosa         | Melia Hotels |
| Carlos       | Iberostar    |
| Elena        | NH Hotels    |
+--------------+--------------+
```

Full Join

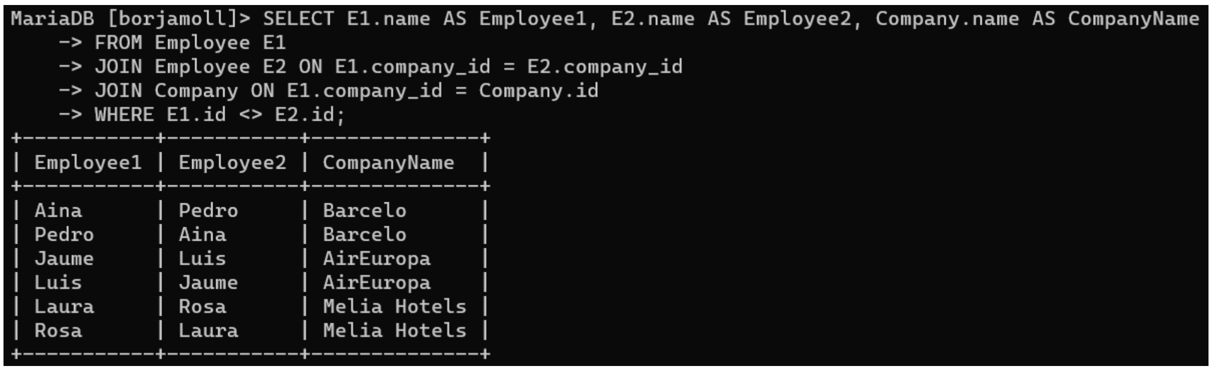


```sql
SELECT a.name AS AwardName, m.title AS MovieName
FROM award a
FULL JOIN movie m ON a.id_movie = m.id;
```

Data Output Messages Notifications

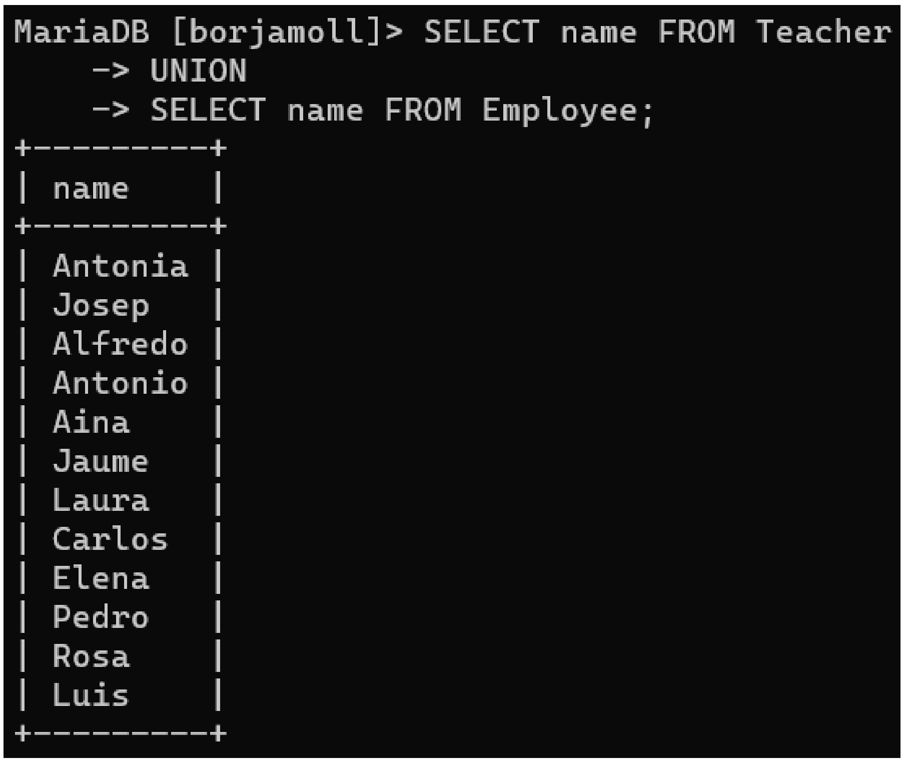
	awardname character varying (50)	moviename character varying (50)
1	Oscar Mejor Actor de Reparto	El caballero oscuro
2	MVT Movie Awards Mejor villano	El caballero oscuro
3	Globo de oro al mejor actor de reparto	El caballero oscuro
4	Premios Empire Mejor Director	El caballero oscuro
5	MTV Movie Award Mejor Interpretación Masculina	Piratas del caribe: La maldición de la perla negra
6	Premio Empire al Mejor Actor	Piratas del caribe: La maldición de la perla negra
7	Óscar Mejor Actor de Reparto	Shine
8	[null]	Trascendence

Self Join



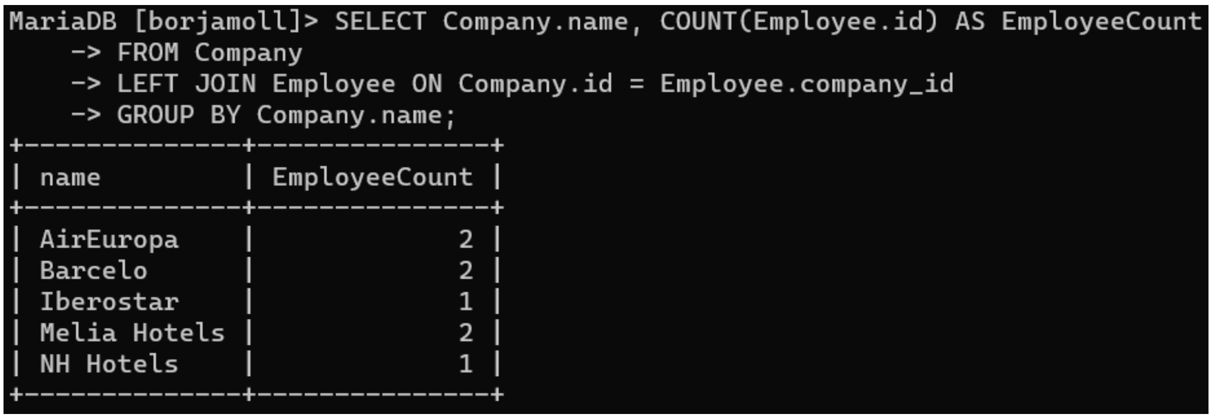
```
MariaDB [borjamoll]> SELECT E1.name AS Employee1, E2.name AS Employee2, Company.name AS CompanyName
    -> FROM Employee E1
    -> JOIN Employee E2 ON E1.company_id = E2.company_id
    -> JOIN Company ON E1.company_id = Company.id
    -> WHERE E1.id <> E2.id;
+-----------+-----------+--------------+
| Employee1 | Employee2 | CompanyName  |
+-----------+-----------+--------------+
| Aina      | Pedro     | Barcelo      |
| Pedro     | Aina      | Barcelo      |
| Jaume     | Luis      | AirEuropa    |
| Luis      | Jaume     | AirEuropa    |
| Laura     | Rosa      | Melia Hotels |
| Rosa      | Laura     | Melia Hotels |
+-----------+-----------+--------------+
```

Union



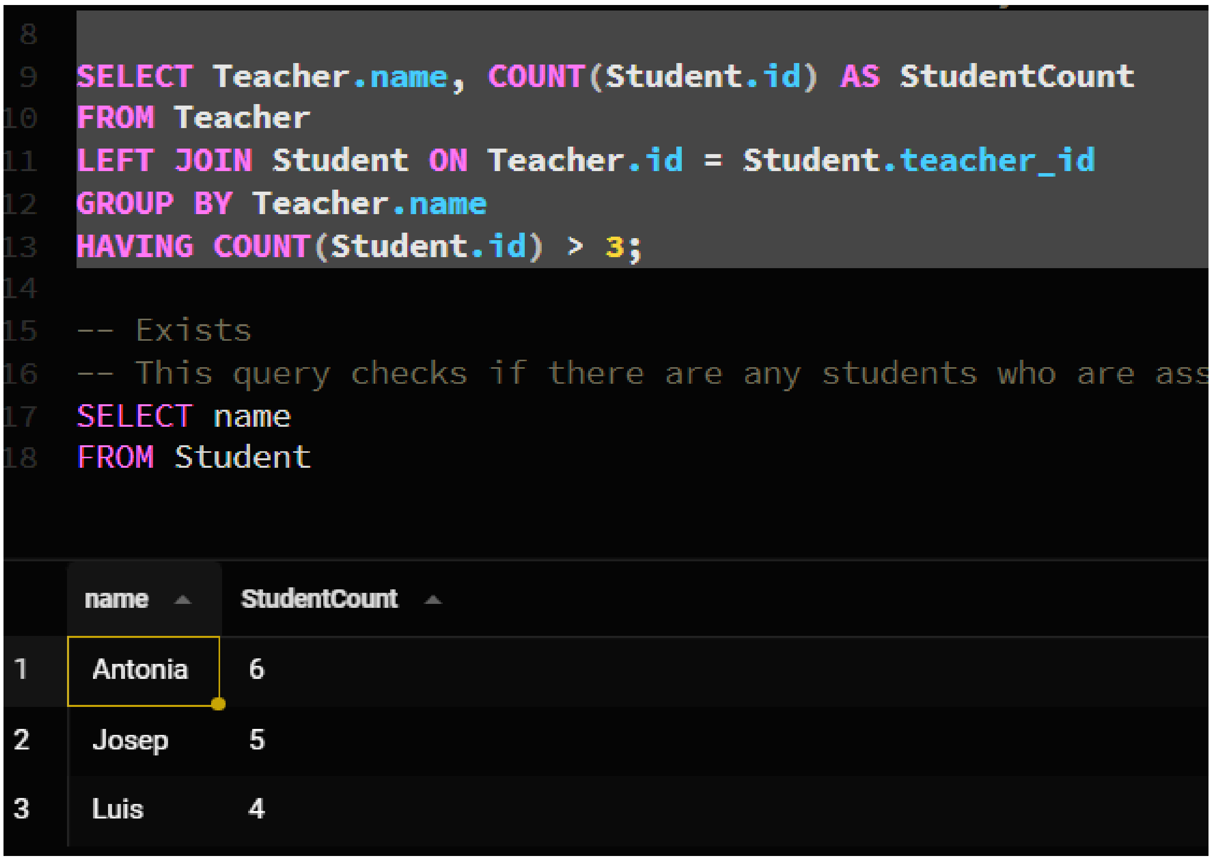
```
MariaDB [borjamoll]> SELECT name FROM Teacher
    -> UNION
    -> SELECT name FROM Employee;
+---------+
| name    |
+---------+
| Antonia |
| Josep   |
| Alfredo |
| Antonio |
| Aina    |
| Jaume   |
| Laura   |
| Carlos  |
| Elena   |
| Pedro   |
| Rosa    |
| Luis    |
+---------+
```

Group By



```
MariaDB [borjamoll]> SELECT Company.name, COUNT(Employee.id) AS EmployeeCount
    -> FROM Company
    -> LEFT JOIN Employee ON Company.id = Employee.company_id
    -> GROUP BY Company.name;
+--------------+---------------+
| name         | EmployeeCount |
+--------------+---------------+
| AirEuropa    |             2 |
| Barcelo      |             2 |
| Iberostar    |             1 |
| Melia Hotels |             2 |
| NH Hotels    |             1 |
+--------------+---------------+
```

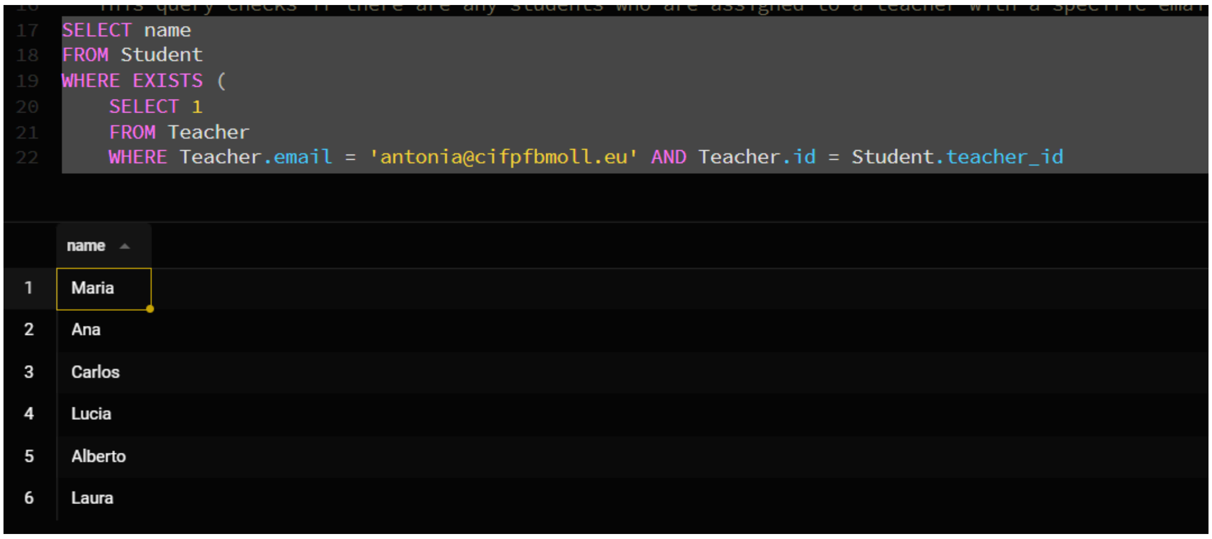
Having



```sql
SELECT Teacher.name, COUNT(Student.id) AS StudentCount
FROM Teacher
LEFT JOIN Student ON Teacher.id = Student.teacher_id
GROUP BY Teacher.name
HAVING COUNT(Student.id) > 3;

-- Exists
-- This query checks if there are any students who are ass
SELECT name
FROM Student
```

```
   name      StudentCount
1  Antonia   6
2  Josep     5
3  Luis      4
```

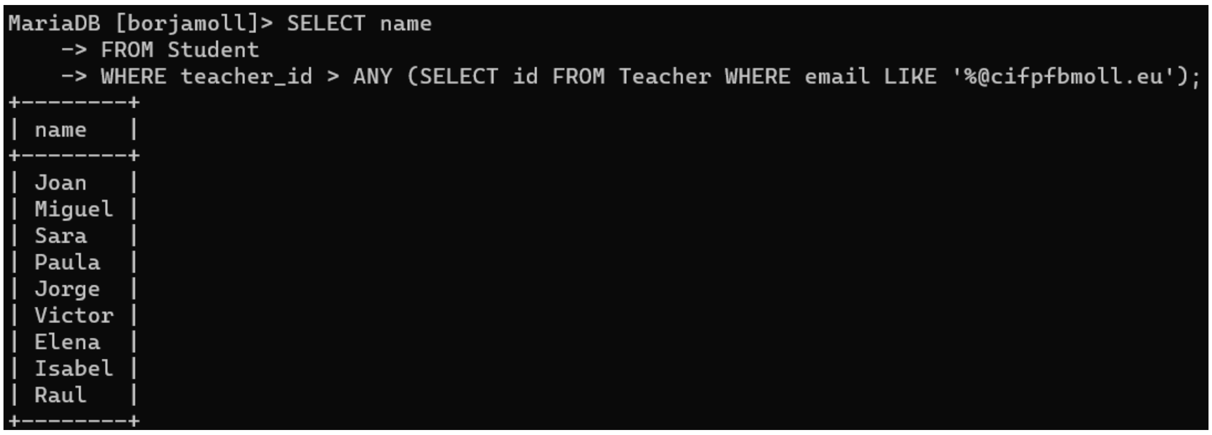
Exists



```sql
SELECT name
FROM Student
WHERE EXISTS (
    SELECT 1
    FROM Teacher
    WHERE Teacher.email = 'antonia@cifpfbmoll.eu' AND Teacher.id = Student.teacher_id
```

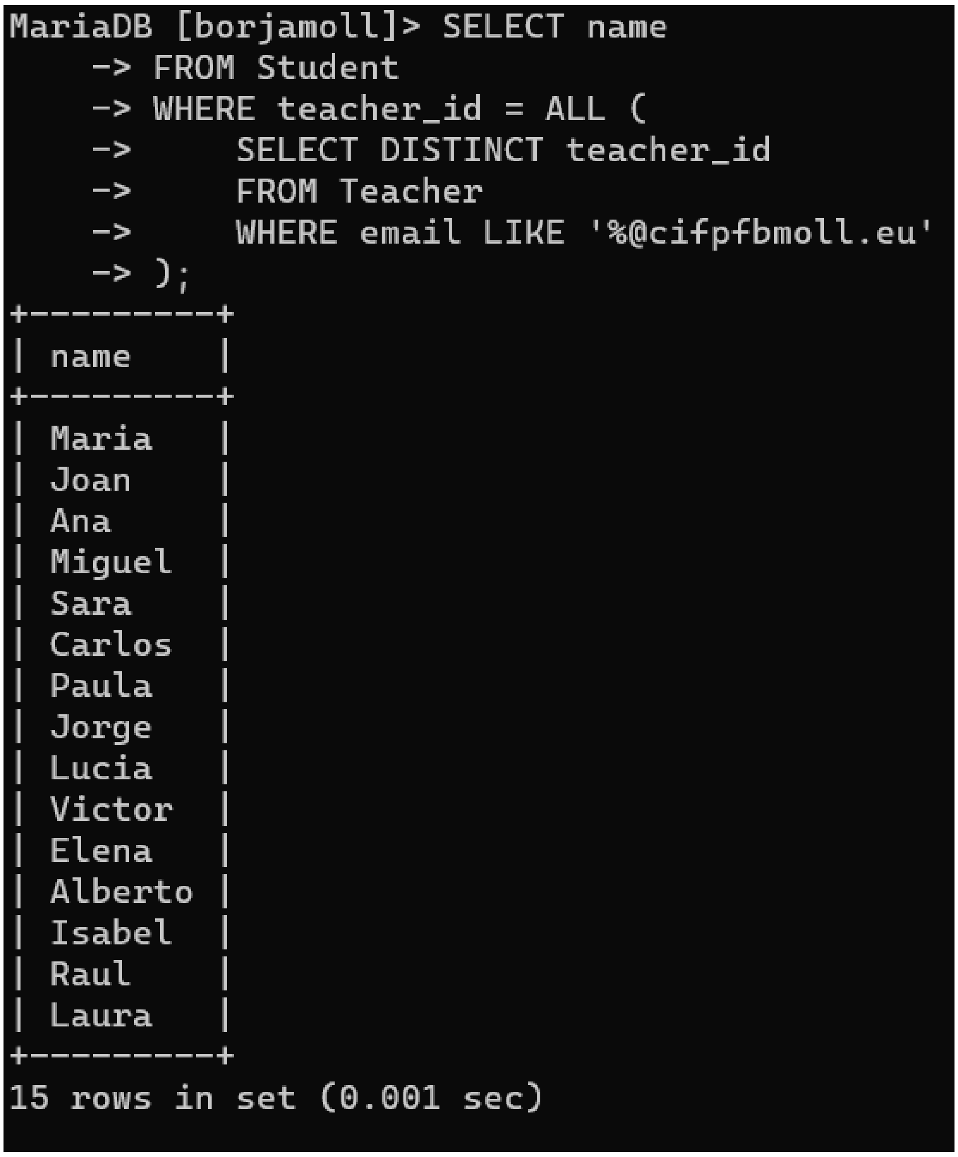
```
   name
1  Maria
2  Ana
3  Carlos
4  Lucia
5  Alberto
6  Laura
```

Any



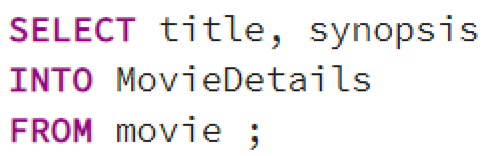
```
MariaDB [borjamoll]> SELECT name
    -> FROM Student
    -> WHERE teacher_id > ANY (SELECT id FROM Teacher WHERE email LIKE '%@cifpfbmoll.eu');
+--------+
| name   |
+--------+
| Joan   |
| Miguel |
| Sara   |
| Paula  |
| Jorge  |
| Victor |
| Elena  |
| Isabel |
| Raul   |
+--------+
```

All

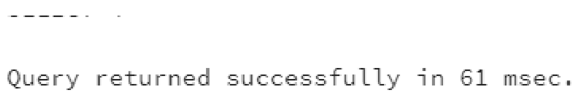


```
MariaDB [borjamoll]> SELECT name
    -> FROM Student
    -> WHERE teacher_id = ALL (
    ->     SELECT DISTINCT teacher_id
    ->     FROM Teacher
    ->     WHERE email LIKE '%@cifpfbmoll.eu'
    -> );
+---------+
| name    |
+---------+
| Maria   |
| Joan    |
| Ana     |
| Miguel  |
| Sara    |
| Carlos  |
| Paula   |
| Jorge   |
| Lucia   |
| Victor  |
| Elena   |
| Alberto |
| Isabel  |
| Raul    |
| Laura   |
+---------+
15 rows in set (0.001 sec)
```

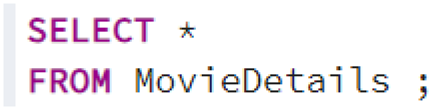
Select into



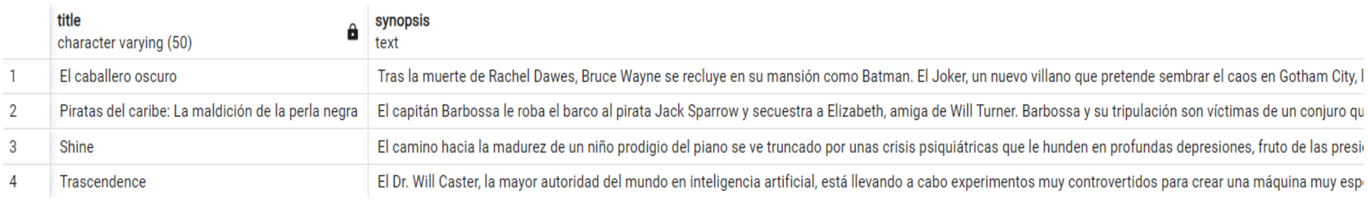
```
SELECT title, synopsis
INTO MovieDetails
FROM movie ;
```



```
Query returned successfully in 61 msec.
```

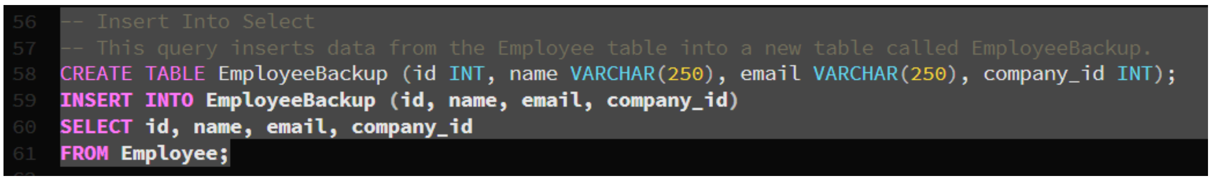


```
SELECT *
FROM MovieDetails ;
```

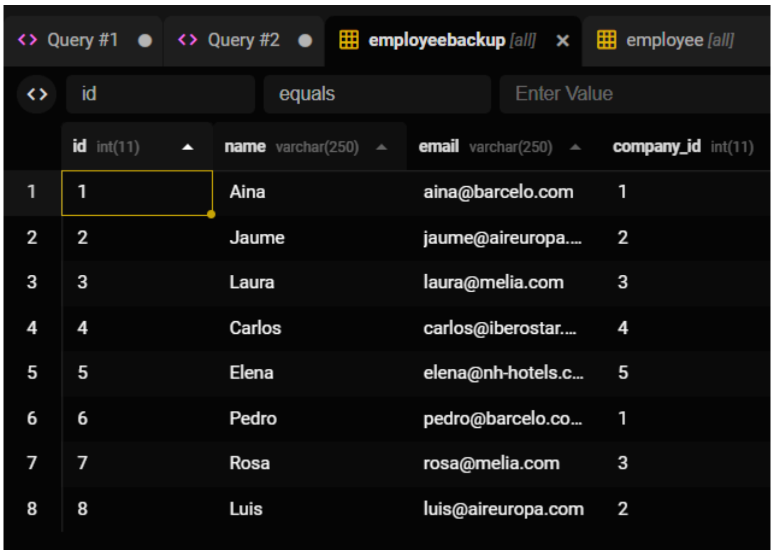


	title character varying (50)	synopsis text
1	El caballero oscuro	Tras la muerte de Rachel Dawes, Bruce Wayne se recluye en su mansión como Batman. El Joker, un nuevo villano que pretende sembrar el caos en Gotham City, l
2	Piratas del caribe: La maldición de la perla negra	El capitán Barbossa le roba el barco al pirata Jack Sparrow y secuestra a Elizabeth, amiga de Will Turner. Barbossa y su tripulación son víctimas de un conjuro qu
3	Shine	El camino hacia la madurez de un niño prodigio del piano se ve truncado por unas crisis psiquiátricas que le hunden en profundas depresiones, fruto de las presi
4	Trascendence	El Dr. Will Caster, la mayor autoridad del mundo en inteligencia artificial, está llevando a cabo experimentos muy controvertidos para crear una máquina muy esp

Insert into Select

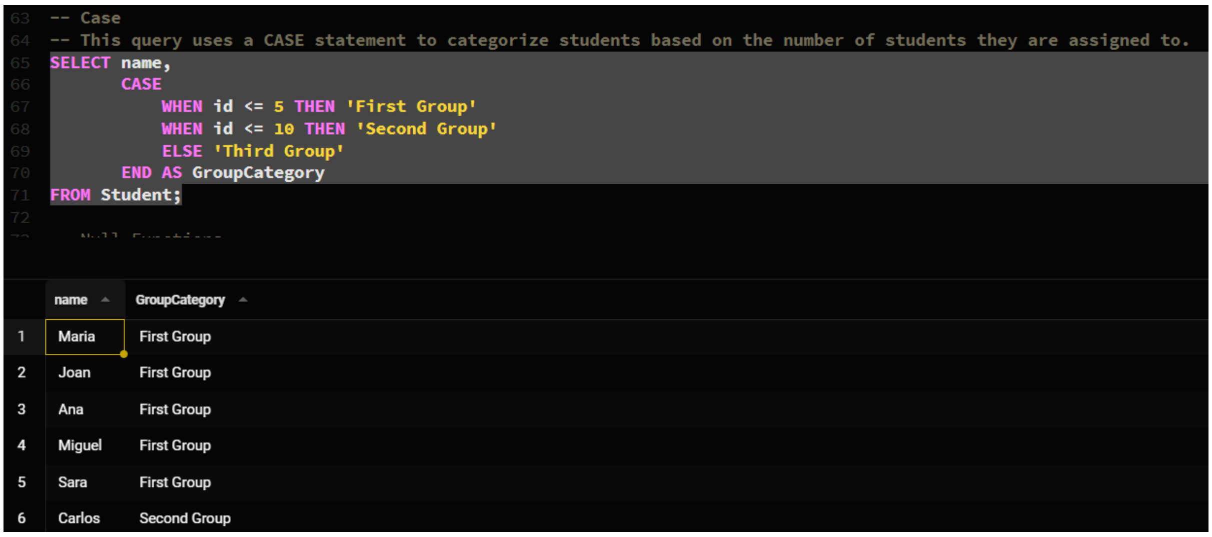


```
-- Insert Into Select
-- This query inserts data from the Employee table into a new table called EmployeeBackup.
CREATE TABLE EmployeeBackup (id INT, name VARCHAR(250), email VARCHAR(250), company_id INT);
INSERT INTO EmployeeBackup (id, name, email, company_id)
SELECT id, name, email, company_id
FROM Employee;
```



	id int(11)	name varchar(250)	email varchar(250)	company_id int(11)
1	1	Aina	aina@barcelo.com	1
2	2	Jaume	jaume@aireuropa....	2
3	3	Laura	laura@melia.com	3
4	4	Carlos	carlos@iberostar....	4
5	5	Elena	elena@nh-hotels.c...	5
6	6	Pedro	pedro@barcelo.co...	1
7	7	Rosa	rosa@melia.com	3
8	8	Luis	luis@aireuropa.com	2

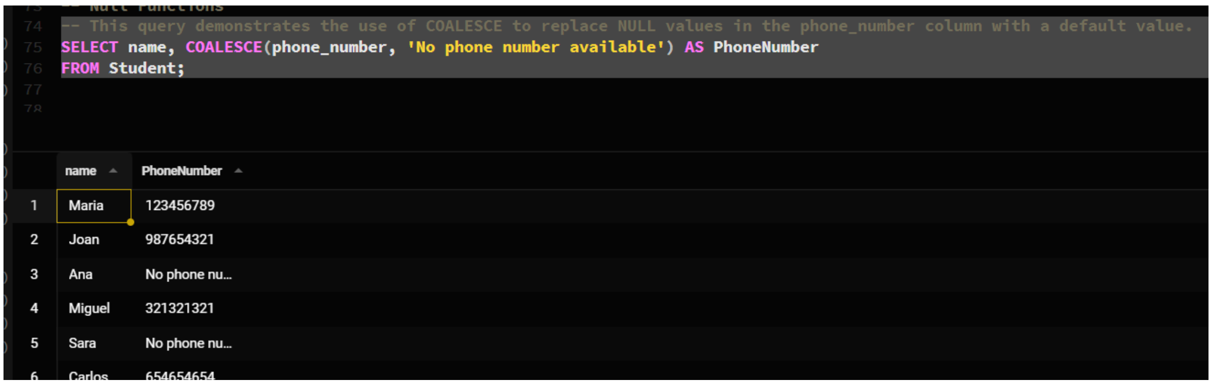
Case



```sql
-- Case
-- This query uses a CASE statement to categorize students based on the number of students they are assigned to.
SELECT name,
       CASE
           WHEN id <= 5 THEN 'First Group'
           WHEN id <= 10 THEN 'Second Group'
           ELSE 'Third Group'
       END AS GroupCategory
FROM Student;
```

	name	GroupCategory
1	Maria	First Group
2	Joan	First Group
3	Ana	First Group
4	Miguel	First Group
5	Sara	First Group
6	Carlos	Second Group

Null functions



```sql
-- Null Functions
-- This query demonstrates the use of COALESCE to replace NULL values in the phone_number column with a default value.
SELECT name, COALESCE(phone_number, 'No phone number available') AS PhoneNumber
FROM Student;
```

	name	PhoneNumber
1	Maria	123456789
2	Joan	987654321
3	Ana	No phone nu...
4	Miguel	321321321
5	Sara	No phone nu...
6	Carlos	654654654

Information sources

1. W3Schools SQL Tutorial
   https://www.w3schools.com/sql/

2. The Complete Guide to SQL GROUP BY
   https://www.sqltutorial.org/sql-group-by/

3. Khan Academy SQL Tutorial
   https://www.khanacademy.org/computing/computer-programming/sql