

# Sprint 2: Nivel 1

## Exercici 1

Mostra totes les transaccions realitzades per empreses d'Alemanya.

Para resolver este exercici utilizo el siguiente código:

```
SELECT *
FROM transactions.company
JOIN transactions.transaction
ON transaction.company_id = company.id
WHERE company.country = "germany";
```

Utilizo la función WHERE para seleccionar solo las empresas de Alemania. Este código devuelve 118 filas que contienen información de las empresas que han realizado transacciones desde Alemania.

The screenshot shows a SQL IDE interface. The top pane contains a SQL query with comments in Catalan. The bottom pane displays the results of the query in a table format.

**SQL Query:**

```
1  ##### SPRINT 2 #####
2  ##### NIVEL 1 #####
3  ## Exercici 1
4  /* Mostra totes les transaccions realitzades per empreses d'Alemanya.*/
5
6  SELECT *
7  FROM transactions.company
8  JOIN transactions.transaction
9  ON transaction.company_id = company.id
10 WHERE company.country = "germany";
11
```

**Result Grid:**

id	company_name	phone	email	country	website	id	credit_card_id	company_id	user_id	lat	longitude	tn
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.telus@yahoo.net	Germany	https://instagram.com/site	1088 ID ID-5B23-A76C-55EF-C568E49A0500	CcU-2938	b-2222	275	83.7839	-178.86	20:
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.telus@yahoo.net	Germany	https://instagram.com/site	EA2C3281-C9C1-A387-44F8-729FB4B51C76	CcU-2938	b-2222	275	20.2004	-116.84	20:
b-2234	Conwallis In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany	https://cnn.com/user/110	0002E608-5C9E-D1B3-4999-699F43AD735A	CcU-2959	b-2234	275	9.68811	130.282	20:
b-2234	Conwallis In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany	https://cnn.com/user/110	AB069F53-969E-A2A8-CE06-CABC4FD92501	CcU-2959	b-2234	275	1.64819	-158.007	20:
b-2302	Nunc Interdum Incorporated	05 18 15 48 13	non@outlook.com	Germany	https://wikipedia.org/en-us	0466A42E-47CF-8024-FD01-C06689713128	CcU-4219	b-2302	170	-43.9695	-117.525	20:
b-2302	Nunc Interdum Incorporated	05 18 15 48 13	non@outlook.com	Germany	https://wikipedia.org/en-us	0A476ED9-0C13-1963-F87B-D35639246539	CcU-4359	b-2302	221	-56.4901	114.801	20:
b-2302	Nunc Interdum Incorporated	05 18 15 48 13	non@outlook.com	Germany	https://wikipedia.org/en-us	1220C333-E19F-D629-DCD8-9C54CF1E8B9A	CcU-4366	b-2302	221	29.6372	-166.173	20:
b-2302	Nunc Interdum Incorporated	05 18 15 48 13	non@outlook.com	Germany	https://wikipedia.org/en-us	1352678A-2E7D-957C-C42C-6450A2B3ED54	CcU-4520	b-2302	210	20.6724	14.9732	20:

**Output:**

1 11:23:51 SELECT \* FROM transactions.company JOIN transactions.transaction ON transaction.company\_id = company.id ... 118 row(s) returned

Duration / Fetch: 0.000 sec / 0.000 sec

## Exercici 2

Màrqueting està preparant alguns informes de tancaments de gestió, et demanen que els passis un llistat de les empreses que han realitzat transaccions per una suma superior a la mitjana de totes les transaccions.

Informe de tancament de gestió:

**Paso 1:** Obtener el valor de la media de todas las transacciones

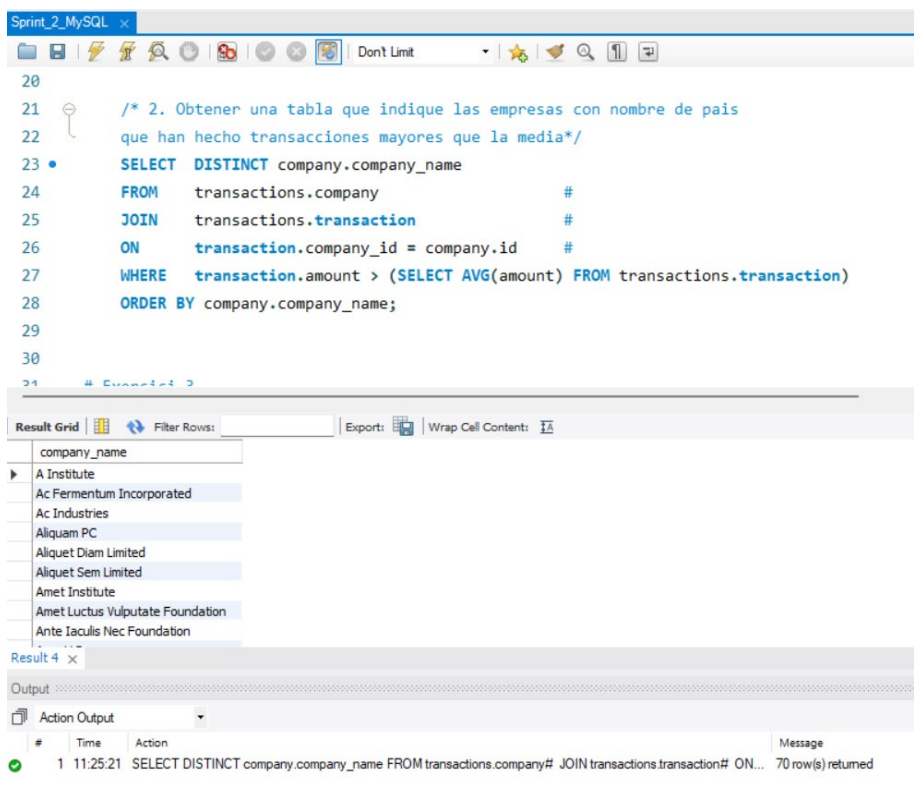
```
SELECT AVG (amount) FROM transactions.transaction;
```

**Paso 2:** Obtener una tabla que indique las empresas que han hecho transacciones mayores que la media

```
SELECT DISTINCT company.company_name
FROM   transactions.company
JOIN   transactions.transaction
ON     transaction.company_id = company.id
WHERE  transaction.amount > (SELECT AVG(amount) FROM transactions.transaction)
ORDER BY company.company_name;
```

Utilizo las funciones AVG() y DISTINCT para calcular la media de las transacciones y obtener una lista única de países respectivamente.

El output es el siguiente:



The screenshot shows a MySQL IDE window titled "Sprint\_2\_MySQL". The SQL editor contains the following query:

```
/* 2. Obtener una tabla que indique las empresas con nombre de pais
que han hecho transacciones mayores que la media*/
SELECT DISTINCT company.company_name
FROM   transactions.company
JOIN   transactions.transaction
ON     transaction.company_id = company.id
WHERE  transaction.amount > (SELECT AVG(amount) FROM transactions.transaction)
ORDER BY company.company_name;
```

Below the editor, the "Result Grid" shows the results of the query. The first column is "company\_name". The results are as follows:

company_name
A Institute
Ac Fermentum Incorporated
Ac Industries
Aliquam PC
Aliquet Diam Limited
Aliquet Sem Limited
Amet Institute
Amet Luctus Vulputate Foundation
Ante Iaculis Nec Foundation

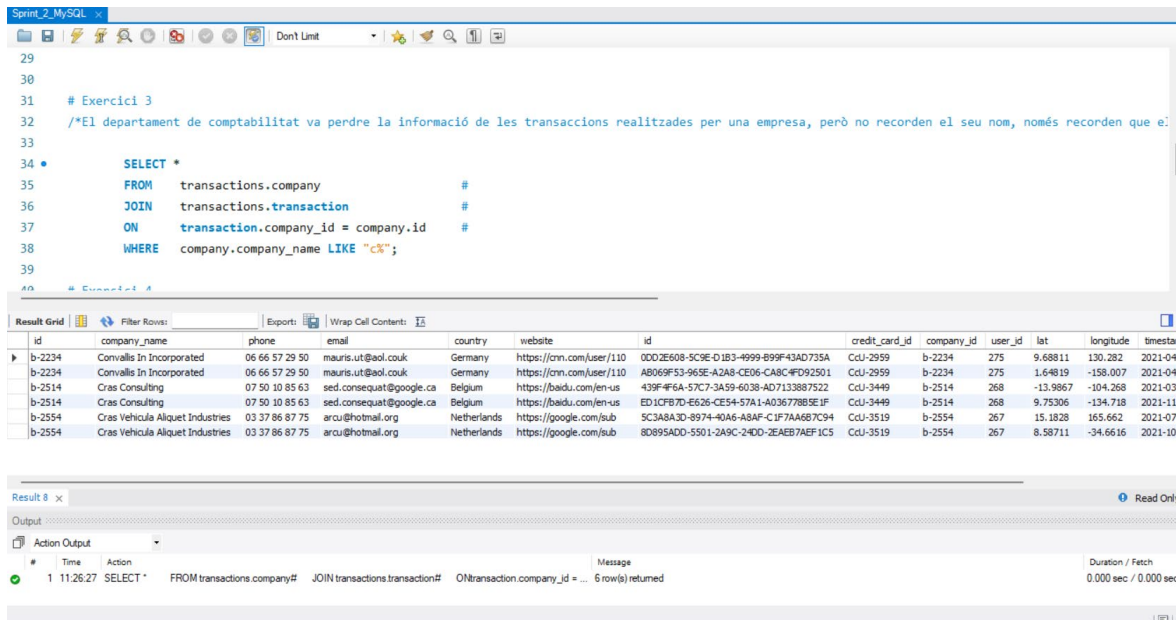
At the bottom, the "Output" pane shows the execution details:

#	Time	Action	Message
1	11:25:21	SELECT DISTINCT company.company_name FROM transactions.company# JOIN transactions.transaction# ON...	70 row(s) returned

## Exercici 3

1. Lo primero es identificar aquellas compañías que cumplen esto. En este caso bastará con un JOIN y un WHERE con el comando LIKE.
2. El comando **LIKE** permite buscar un patrón, que en este caso es que la primera letra comience por "c".

El resultado es la siguiente tabla:



The screenshot shows a MySQL database interface with a SQL query and its results. The query is as follows:

```
SELECT *
FROM transactions.company #
JOIN transactions.transaction #
ON transaction.company_id = company.id #
WHERE company.company_name LIKE "c%";
```

The results are displayed in a table with the following columns: id, company\_name, phone, email, country, website, id, credit\_card\_id, company\_id, user\_id, lat, longitude, and timestamp. The table contains 10 rows of data, all of which start with the letter 'C' in the company\_name column.

id	company_name	phone	email	country	website	id	credit_card_id	company_id	user_id	lat	longitude	timestamp
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.co.uk	Germany	https://cnn.com/user/110	0002E608-5C9E-Q1B3-4999-899F43AD735A	CcU-2959	b-2234	275	9.68811	130.282	2021-04
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.co.uk	Germany	https://cnn.com/user/110	AB069F53-965E-A2A8-CE06-CABCFD92501	CcU-2959	b-2234	275	1.64819	-158.007	2021-04
b-2514	Cras Consulting	07 50 10 85 63	sed.consequat@google.ca	Belgium	https://baidu.com/en-us	439F4F6A-57C7-3A59-6038-AD7133887522	CcU-3449	b-2514	268	-13.9867	-104.268	2021-03
b-2514	Cras Consulting	07 50 10 85 63	sed.consequat@google.ca	Belgium	https://baidu.com/en-us	ED1CFB7D-E626-CE54-57A1-A03677885E1F	CcU-3449	b-2514	268	9.75306	-134.718	2021-11
b-2554	Cras Vehicula Aliquet Industries	03 37 86 87 75	arcu@hotmail.org	Netherlands	https://google.com/sub	5C3A8A3D-8974-40A6-ABAF-C1F7AA667C94	CcU-3519	b-2554	267	15.1828	165.662	2021-07
b-2554	Cras Vehicula Aliquet Industries	03 37 86 87 75	arcu@hotmail.org	Netherlands	https://google.com/sub	8D895ADD-5501-2A9C-24D0-2EAE87AEF1C5	CcU-3519	b-2554	267	8.58711	-34.6616	2021-10

## Exercici 4

Van eliminar del sistema les empreses que no tienen transacciones registradas, lliura el llistat d'aquestes empreses.

Hago una comparación manual del número de compañías en cada tabla:

```
SELECT DISTINCT company_id FROM transaction;
SELECT id FROM company;
```

Con esto observo que tanto transaction como company tienen el mismo numero de filas (100). Lo que indica que es posible que esten todas contenidas.

## Opcion 1

Para ello, debería tener un tipo de join en el que tenga las emrpeas que no tienen transacciones. Puedo usar un left join y buscar valores que sean nulo.

Es decir, con un left join tendré una tabla con aquellas compañías que estando en la tabla company, en la tabla transacion no estan y por lo mismo, en este join el valor será null

```
SELECT company.id,
```

```
transaction.amount
FROM transactions.company
LEFT JOIN transactions.transaction
ON transaction.company_id = company.id
WHERE transactions.company.id IS NULL;
```

## Opcion 2

Tambien podría utilizar una subquery para realizar la misma consulta

```
SELECT company_id, transaction.amount
FROM transaction
WHERE company_id NOT IN (SELECT id FROM company);
```

- El *IN* sirve para specify multiple possible values for a column,
- Puedo generar una subquery con los valores de una tabla

En otras palabras, esta parte de la consulta selecciona solo aquellas filas de la tabla **transaction** cuyo *company\_id* no está presente en la columna *id* de la tabla **company**.

En ambos casos el resultado es null o vacio.

```
56
57 •      SELECT      company.id,
58                      transaction.amount
59      FROM      transactions.company
60      LEFT JOIN  transactions.transaction
61      ON        transaction.company_id = company.id
62      WHERE     transactions.company.id IS NULL;
63
```

Result Grid			Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:
id	amount				