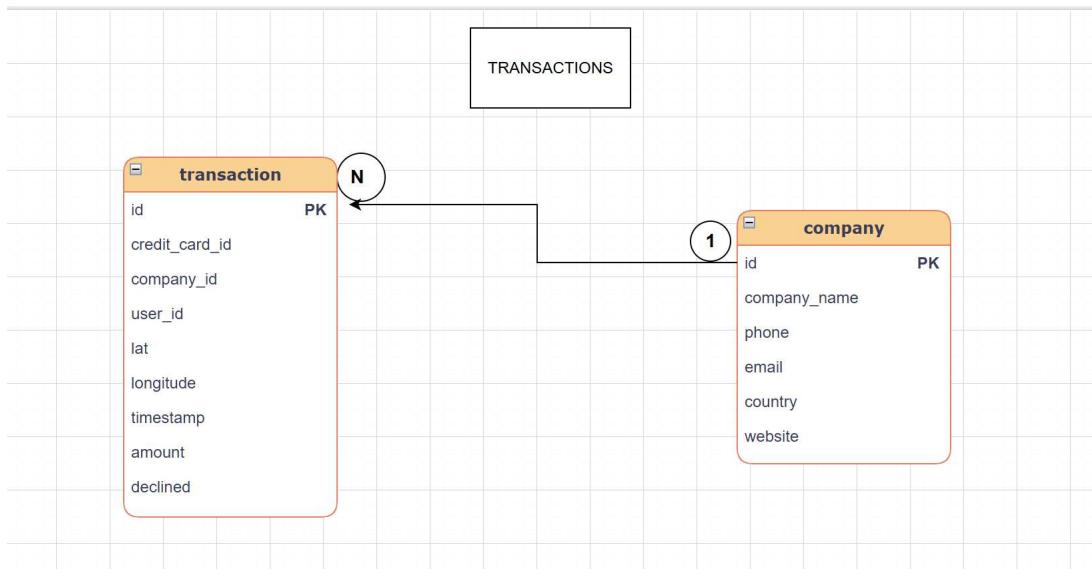


## Capturas Wordbench Sprint S2-01

### # Tasca-S2.01.-Nocions-basiques-SQL

#### Nivel 1: Ejercicio 1

A partir dels documents adjunts (estructura\_dades i dades\_introduir), importa les dues taules. Mostra les característiques principals de l'esquema creat i explica les diferents taules i variables que existeixen. Assegura't d'incloure un diagrama que il·lustri la relació entre les diferents taules i variables.



Una vez importados los dos archivos en Workbench, se obtienen un schema llamado transactions, que a su vez contienen dos tablas:

- **transaction**: la cual muestra los registros de las transacciones de la empresa con otras compañías que compran sus productos, se puede observar columnas como el id de la transaccion, el id de la tarjeta de crédito, id de la compañía que compra, un identificador de esa compañía, las coordenadas como latitud y longitud, fecha de la transaccion, el monto de la transaccion y una columna si la venta finalmente se hizo o no se llegó a concretar.
- **company**: esta tabla muestra los datos de las compañías que compran, desde el nombre de la compañía, telefono, email, pais de origen, y pagina web.

La relacion es mediante los Primary Keys de cada tabla( en el grafico esta marcado como PK) y es una relacion de 1 a muchos ( desde company hacia transaction), es decir que una compañía puede tener muchas transacciones.

En los ejercicios he usado la condicion declined = 0 para las preguntas que solo pedian las compras, pero en las que piden transacciones lo he omitido, porque aunque esten declinadas las transacciones son transacciones igualmente.

## Exercici 2

Utilitzant JOIN realitzaràs les següents consultes:

### 1) Llistat dels països que estan fent compres.

```
8  -- Exercici 2
9  -- Utilitzant JOIN realitzaràs les següents consultes:
10 -- 1) Llistat dels països que estan fent compres.
11 • SELECT DISTINCT country
12   FROM transaction
13   JOIN company ON company_id = company.id -- llamar a tablas solo las que pueden ser columnas ambiguas como compan
14   WHERE declined = 0
15   ORDER BY country ASC;
```

The screenshot shows the MySQL Workbench interface. At the top, there is a code editor window with the SQL query from step 1. Below it is a 'Result Grid' window displaying a list of countries: United States, United Kingdom, Sweden, Spain, Norway, New Zealand, Netherlands, Italy, Ireland, Germany, France, China, Canada, Belgium, and Australia. An 'Output' window at the bottom shows the execution log, indicating the query was run at 12:40:11 and returned 15 rows.

### 2) Des de quants països es realitzen les compres.

```
11  -- 2) Des de quants països es realitzen les compres.
12 • SELECT COUNT(DISTINCT country) AS num_country
13   FROM company
14   JOIN transaction ON company_id = company.id
15   WHERE declined = 0;
16
17  -- 3) Identifica la companyia amb la mitjana més gran de vendes
18 • SELECT company.name, ROUND(AVG(amount), 2) AS media
```

The screenshot shows the MySQL Workbench interface. It displays the result of the second query, which is a single row with the value '15' under the column 'num\_country'. The 'Result Grid' window has a header 'num\_country' and contains one row with the value '15'.

The screenshot shows the MySQL Workbench interface. It displays the result of the third query, which is a single row with the value '1' under the column 'media'. The 'Result Grid' window has a header 'media' and contains one row with the value '1'.

**3) Identifica la companyia amb la mitjana més gran de vendes.**

```
17    -- 3) Identifica la companyia amb la mitjana més gran de vendes
18 •  SELECT company_name, ROUND(AVG(amount),2) as media
19   FROM company
20   JOIN transaction on company_id = company.id
21   WHERE declined = 0
22   GROUP BY company_name
23   ORDER BY media DESC
24   limit 1;
25
26 -- Exercici 3
```

A screenshot of a MySQL query results grid. The grid has two columns: 'company\_name' and 'media'. There is one row of data: 'Eget Ipsum Ltd' with a value of '481.86'. The grid includes standard database navigation buttons like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'.

company_name	media
Eget Ipsum Ltd	481.86

A screenshot of a MySQL query history window titled 'Result 27'. It shows a single entry: a SELECT query that returned 1 row(s). The query details are: 'SELECT company\_name, ROUND(AVG(amount),2) as media FROM company JOIN transaction on company\_id = com...', with a timestamp of '21:43:01'. The status message indicates '1 row(s) returned'.

#	Time	Action	Message
1	21:43:01	SELECT company_name, ROUND(AVG(amount),2) as media FROM company JOIN transaction on company_id = com...	1 row(s) returned

**Exercici 3**

**Utilitzant només subconsultes (sense utilitzar JOIN):**

**1) Mostra totes les transaccions realitzades per empreses d'Alemanya.**

\*\*Se ha usado ANY en lugar de IN que es un poco mas eficiente, pero para mas volumen de datos es necesario usar EXIST, mucho mas eficiente , menos memoria y menos trabajo

```

35  -- Exercici 3
36  -- Utilitzant només subconsultes (sense utilitzar JOIN):
37
38  -- 1) Mostra totes les transaccions realitzades per empreses d'Alemany
39 • |SELECT *
40   FROM transaction
41   WHERE company_id = any (SELECT id
42     FROM company
43     WHERE country = 'Germany');
44

```

transaction 40 ×

Output

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A05DD	CcU-2938	b-2222	275	83.7839	-178.86	2021-07-07 17:43:16	293.57	0
EA2C3281-C9C1-A387-44F8-729FB4B51C76	CcU-2938	b-2222	275	20.2004	-116.84	2021-05-09 10:25:08	119.36	1
0DD2E608-5C9E-D1B3-4999-B99F43AD735A	CcU-2959	b-2234	275	9.68811	130.282	2021-04-17 05:30:17	252.47	1
AB069F53-965E-A2A8-CE06-CA8C4FD92501	CcU-2959	b-2234	275	1.64819	-158.007	2021-04-15 13:37:18	60.99	0
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:29:18	49.53	0

Action Output

#	Time	Action	Message
1	23:16:10	SELECT * FROM transaction WHERE company_id = any (SELECT id FROM company WHERE country = 'Germany')	118 row(s) returned

## FORMA MAS OPTIMIZADA CON EXIST:

```

36  -- FORMA MAS OPTIMA CON EXIST Exercici 3
37 • |SELECT *
38   FROM transaction t
39   WHERE EXISTS (
40     SELECT 1
41     FROM company
42     WHERE id = company_id
43     AND country = 'Germany'
44   );
45
46  -- 2) Llista les empreses que han realitzat transaccions

```

transaction 45 ×

Output

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A05DD	CcU-2938	b-2222	275	83.7839	-178.86	2021-07-07 17:43:16	293.57	0
EA2C3281-C9C1-A387-44F8-729FB4B51C76	CcU-2938	b-2222	275	20.2004	-116.84	2021-05-09 10:25:08	119.36	1
0DD2E608-5C9E-D1B3-4999-B99F43AD735A	CcU-2959	b-2234	275	9.68811	130.282	2021-04-17 05:30:17	252.47	1
AB069F53-965E-A2A8-CE06-CA8C4FD92501	CcU-2959	b-2234	275	1.64819	-158.007	2021-04-15 13:37:18	60.99	0
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:29:18	49.53	0
0A476ED9-0C13-1962-F87B-D3563924B539	CcU-4359	b-2302	221	-56.4901	114.801	2022-02-26 20:33:54	430.49	0

Action Output

#	Time	Action	Message
1	23:56:13	SELECT * FROM transaction t WHERE EXISTS ( SELECT 1 FROM company WHERE id = company_id )	118 row(s) returned

## RESUMEN:

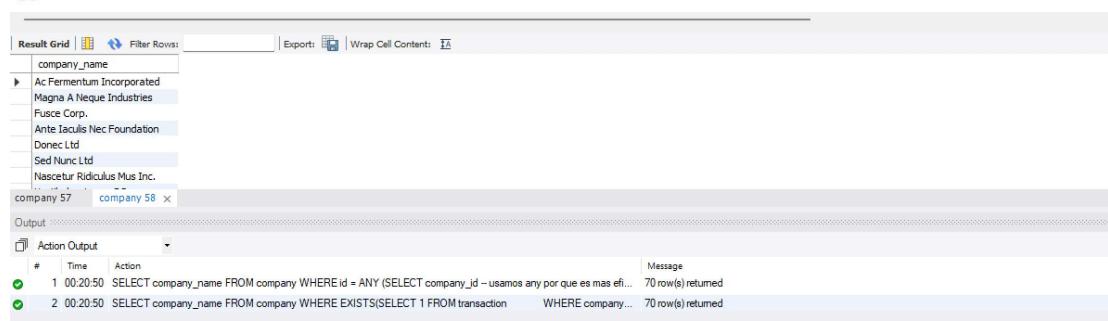
## Resumen corto — EXISTS vs ANY

- ANY / IN: ejecuta el subquery completo y compara con una lista → más lento si devuelve muchas filas.
- EXISTS: evalúa por cada fila y se detiene al encontrar la primera coincidencia → más eficiente.
- No devuelve valores del subquery, solo comprueba existencia.
- SELECT 1 se usa por claridad: no importa qué selecciones, solo si hay filas.
- SELECT \* funciona igual, pero es mala práctica.
- Usa EXISTS para verificar relaciones y JOIN si necesitas datos de ambas tablas.

2) Llista les empreses que han realitzat transaccions per un amount superior a la mitjana de totes les transaccions.

Hay dos forma , la mas eficiente es con exist

```
46 -- 2) Llista les empreses que han realitzat transaccions
47 -- per un amount superior a la mitjana de totes les transaccions.
48 • SELECT company_name
49   FROM company
50   WHERE id = ANY (SELECT company_id -- usamos any por que es mas eficiente que like y este se usa en cada
51                   from transaction
52                   WHERE amount >(SELECT AVG(amount) FROM transaction));
53
54 -- forma mas optima con exist
55 • SELECT company_name
56   FROM company
57   WHERE EXISTS(SELECT 1
58                 FROM transaction
59                 WHERE company_id = company.id and amount >(SELECT AVG(amount) FROM transaction));
60
```



The screenshot shows the MySQL command-line interface. At the top, there's a 'Result Grid' header with buttons for 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the header is a table with a single column labeled 'company\_name'. The table contains 70 rows, each representing a company name. The companies listed are: Ac Fermentum Incorporated, Magna A Neque Industries, Fusce Corp., Ante Iaculis Nec Foundation, Donec Ltd, Sed Nunc Ltd, Nascentur Ridiculus Mus Inc., and several others. At the bottom of the screen, there's an 'Output' section with a table titled 'Action Output'. It shows two entries: entry 1 at 00:20:50 with the query 'SELECT company\_name FROM company WHERE id = ANY (SELECT company\_id -- usamos any por que es mas eficiente que like y este se usa en cada from transaction WHERE amount >(SELECT AVG(amount) FROM transaction))' and entry 2 at 00:20:50 with the query 'SELECT company\_name FROM company WHERE EXISTS(SELECT 1 FROM transaction WHERE company\_id = company.id and amount >(SELECT AVG(amount) FROM transaction))'. Both entries show '70 row(s) returned' in the 'Message' column.

3) Eliminaran del sistema les empreses que no tenen transaccions registrades, entrega el llistat d'aquestes empreses.

```

73 •   SELECT company_name
74     FROM company
75     WHERE id NOT IN (SELECT company_id from transaction);
76
77 -- Nivell 2

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
company_name					

company 3 x		
Output		
Action Output		
#	Time	Action
1	11:45:40	SELECT company_name FROM company WHERE id NOT IN (SELECT company_id from transaction)

Message  
0 row(s) returned

## Nivell 2

### Exercici 1

**Identifica els cinc dies que es va generar la quantitat més gran d'ingressos a l'empresa per vendes. Mostra la data de cada transacció juntament amb el total de les vendes.**

```

55 -- Identifica els cinc dies que es va generar la quantitat més gran d'ingressos a l'empresa per vendes.
56 -- Mostra la data de cada transacció juntament amb el total de les vendes.
57
58 •   SELECT DATE(timestamp) AS FECHA, SUM(amount) AS VENTAS_TOTAL
59     FROM transaction
60     WHERE declined = 0
61     GROUP BY FECHA
62     ORDER BY VENTAS_TOTAL DESC
63     LIMIT 5;
64

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
FECHA	VENTAS_TOTAL			
2021-12-20	1532.36			
2021-04-22	1397.96			
2021-05-09	1344.37			
2022-02-26	1337.62			
2021-03-29	1325.12			

Result 40 x		
Output		
Action Output		
#	Time	Action
1	11:40:47	SELECT DATE(timestamp) AS FECHA, SUM(amount) AS VENTAS_TOTAL FROM transaction WHERE declined ...

Message  
5 row(s) returned

### Exercici 2

**Quina és la mitjana de vendes per país? Presenta els resultats ordenats de major a menor mitjà.**

```

90      -- Exercici 2
91      -- Quina és la mitjana de vendes per país? Presenta els resultats ordenats de major a menor mitjà
92 •  SELECT country, round(AVG(amount),2) as monto_medio
93  FROM transaction
94  JOIN company ON company.id = company_id
95  GROUP BY 1
96  ORDER BY 2 DESC;
97

```

**Result Grid**

country	monto_medio
United States	309.18
Ireland	277.31
United Kingdom	270.73
Canada	266.36
Sweden	260.62

**Action Output**

#	Time	Action	Message
1	17:38:24	SELECT country, round(AVG(amount),2) as monto_medio FROM transaction JOIN company ON company.id = company_id GROUP BY 1 ORDER BY 2 DESC;	16 row(s) returned

### Exercici 3

**En la teva empresa, es planteja un nou projecte per a llançar algunes campanyes publicitàries per a fer competència a la companyia "Non Institute".**

**Per a això, et demanen la llista de totes les transaccions realitzades per empreses que estan situades en el mateix país que aquesta companyia.**

**Mostra el llistat aplicant JOIN i subconsultes.**

```

102      -- Mostra el llistat aplicant JOIN i subconsultes.
103
104 •  SELECT *
105    FROM transaction
106   JOIN company ON company.id = company_id
107   WHERE country = (SELECT DISTINCT country
108     FROM company
109    WHERE company_name = "Non Institute");
110

```

**Result Grid**

<b>id</b>	<b>credit_card_id</b>	<b>company_id</b>	<b>user_id</b>	<b>lat</b>	<b>longitude</b>	<b>timestamp</b>	<b>amount</b>	<b>declined</b>	<b>id</b>	<b>company_name</b>	<b>phone</b>	<b>email</b>	<b>cou</b>
28928E1C-EC14-A760-0A75-871477649D6A	CcJ-2980	b-2246	275	-41.0496	161.685	2021-08-10 08:14:49	383.73	0	b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	Unit
ACD2011A-A2B1-C365-41E1-2AB00C651A7A	CcJ-2980	b-2246	275	-54.4792	-92.7974	2022-03-05 20:41:20	60.07	1	b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	Unit
4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	CcJ-3092	b-2310	275	-20.4859	150.87	2021-05-03 22:37:23	458.74	0	b-2310	Non Magna LLC	06 71 73 13 17	nisi quisque.fringilla@hotmail.ca	Unit
BC289A38-77B4-28CD-1F88-140ED863E773	CcJ-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95	1	b-2310	Non Magna LLC	06 71 73 13 17	nisi quisque.fringilla@hotmail.ca	Unit

**Action Output**

#	Time	Action	Message	Duration /
1	17:50:53	SELECT * FROM transaction JOIN company ON company.id = company_id WHERE country = (SELECT DISTINCT country FROM company WHERE company_name = "Non Institute")	100 row(s) returned	0.000 sec

**Mostra el llistat aplicant solament subconsultes.**

```

112      -- Mostra el llistat aplicant solament subconsultes.
113 •   SELECT *
114     FROM transaction
115     WHERE company_id = ANY (SELECT id
116                               FROM company
117                               WHERE country = (SELECT DISTINCT country
118                                     FROM company
119                                     WHERE company_name = "Non Institute"));
120

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	dedine
▶	2B928E1C-EC14-A760-0A75-871477649D6A	CcU-2980	b-2246	275	-41.0496	161.685	2021-08-10 08:14:49	383.73	0
	ACD2011A-A2B1-C365-41E1-2AB00C65147A	CcU-2980	b-2246	275	-54.4792	-82.7974	2022-03-05 20:41:20	60.07	1
	4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	CcU-3092	b-2310	275	-20.4859	150.87	2021-05-03 22:37:23	458.74	0
	BC2B9A38-77B4-28CD-1FE8-14DED863E773	CcU-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95	1
	147983D2-B7BA-C7B8-4CE3-8D7C2DE85ABB	CcU-2994	b-2326	133	66.2672	172.399	2021-08-09 00:58:07	309.45	0

transaction 12 ×

Output:

Action Output

#	Time	Action	Message
1	17:52:02	SELECT * FROM transaction WHERE company_id = ANY (SELECT id FROM company WHERE country = (SELECT DISTINCT country FROM company WHERE company_name = "Non Institute"));	100 row(s) returned

### Nivel 3

#### Ejercicio 1

Presenta el nombre, teléfono, país, fecha y amount, de aquellas empresas que realizaron transacciones con un valor comprendido entre 100 y 200 euros y en alguna de estas fechas: 29 de abril de 2021, 20 de julio de 2021 y 13 de marzo de 2022. Ordena los resultados de mayor a menor cantidad.

```

127 •   SELECT company_name, phone, country, DATE(timestamp) AS FECHA, amount
128     FROM transaction
129     JOIN company ON company.id = company_id
130     WHERE (amount BETWEEN 100 AND 200) AND (DATE(timestamp) IN ('2021-04-29', '2021-07-20', '2022-03-13'))
131     ORDER BY 4 DESC;
132

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	company_name	phone	country	FECHA	amount
▶	Nunc Interdum Incorporated	05 18 15 48 13	Germany	2022-03-13	164.32
	Lorem Eu Incorporated	01 83 66 62 07	Canada	2021-07-20	133.39
	Interdum Feugiat Sed Associates	04 88 40 32 52	United Kingdom	2021-07-20	164.86
	Enim Condimentum Ltd	09 55 51 66 25	United Kingdom	2021-04-29	149.89
	Nunc Interdum Incorporated	05 18 15 48 13	Germany	2021-04-29	111.51

Result 16 ×

Output:

Action Output

#	Time	Action	Message
1	18:12:17	SELECT company_name, phone, country, DATE(timestamp) AS FECHA, amount FROM transaction JOIN company O...	5 row(s) returned

#### Ejercicio 2

Necesitamos optimizar la asignación de los recursos y dependerá de la capacidad operativa que se requiera, por lo que te piden la información sobre la cantidad de transacciones que realizan las empresas, pero el departamento de recursos humanos es exigente y quiere un listado de las empresas en las que especifiques si tienen más de 4 transacciones o menos.

```

138 • | SELECT company_name, COUNT(*),
139   CASE
140     WHEN COUNT(*) < 4 THEN 'Poca actividad'
141     ELSE 'Mucha actividad'
142   END AS actividad
143 FROM transaction
144 JOIN company ON company.id = company_id
145 GROUP BY company_name
146 ORDER BY 2 DESC;

```

Result Grid | Filter Rows: \_\_\_\_\_ | Export: | Wrap Cell Content:

	company_name	COUNT(*)	actividad
▶	Nunc Interdum Incorporated	105	Mucha actividad
	Ut Semper Foundation	59	Mucha actividad
	Enim Condimentum Ltd	57	Mucha actividad
	Arcu LLP	56	Mucha actividad
	Lorem Eu Incorporated	53	Mucha actividad

Result 19 ×

Output

Action Output

#	Time	Action	Message
1	18:18:42	SELECT company_name, COUNT(*), CASE WHEN COUNT(*) < 4 THEN 'Poca actividad' ELSE 'Mucha actividad' END AS actividad FROM transaction JOIN company ON company.id = company_id GROUP BY company_name ORDER BY 2 DESC;	101 row(s) returned