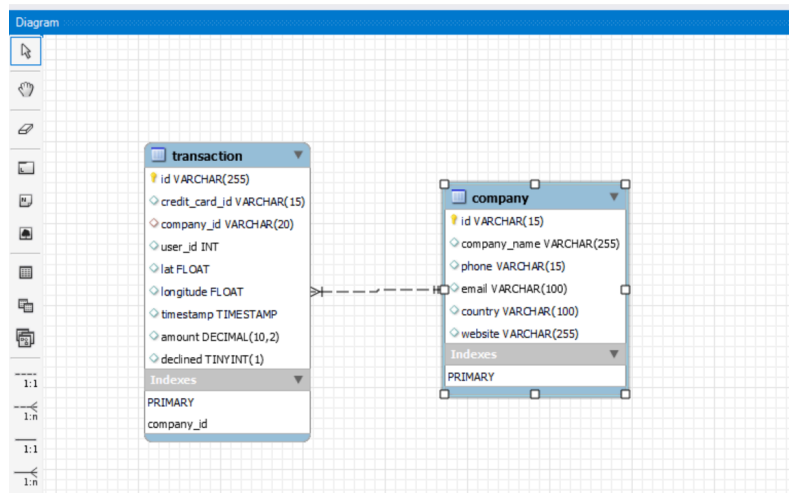


# Hecho por: Yatmelis Freites

## Exercici 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.



La relación entre ambas tablas es de una a muchas de company hacia transaction ya que solo hay una compañía por cada transacción.

La tabla de hechos es la tabla company en este caso, que se relaciona con la tabla de dimensión transaction desde company.id (PK) a transaction.company\_id(FK). Por otra parte la PK de la tabla transaction es transaction.id

## Exercici 2

Utilitzant JOIN realitzaràs les següents consultes:

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

```
10
11 • SELECT DISTINCT company.country
12 FROM company
13 RIGHT JOIN transaction
14 ON company.id=transaction.company_id;
```

Result Grid

country
Germany
Australia
United States
New Zealand
Norway
United Kingdom

Result 1

Output

#	Time	Action	Message
1	19:52:14	SELECT * FROM transactions.company	100 row(s) returned
2	19:52:28	SELECT * FROM transactions.transaction	587 row(s) returned
3	19:58:16	SELECT DISTINCT company.country FROM company RIGHT JOIN transaction ON ...	15 row(s) returned

Esta query selecciona los países únicos de la tabla company que hayan realizado transacciones en la tabla transaction a través de un DISTINCT, por lo cual se necesita un RIGHT JOIN para tomar todos los datos estrictamente relacionados con la tabla de transaction

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

```
16 -- Des de quants països es realitzen les compres.
17 • SELECT COUNT( DISTINCT company.country) AS Païses_Comprando
18 FROM company
19 RIGHT JOIN transaction
20 ON company.id=transaction.company_id;
21
```

Result Grid

Païses_Comprando
15

Result 2 x

Output

Action Output

#	Time	Action	Message
1	19:52:14	SELECT * FROM transactions.company	100 row(s) returned
2	19:52:28	SELECT * FROM transactions.transaction	587 row(s) returned
3	19:58:16	SELECT DISTINCT company.country FROM company RIGHT JOIN transaction ON ...	15 row(s) returned
4	20:03:37	SELECT COUNT(DISTINCT company.country) AS Païses_Comprando FROM comp...	1 row(s) returned

Esta query realiza un COUNT DISTINCT para hacer recuento de los países únicos de la tabla company que hayan realizado transacciones en la tabla transaction para etiquetarlos como Païses\_Comprando, por lo cual se necesita un RIGHT JOIN para tomar todos los datos estrictamente relacionados con la tabla de transaction

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

```
24 • SELECT company.company_name, transaction.company_id, AVG(transaction.amount) as ventas
25 FROM transaction
26 JOIN company
27 ON company.id=transaction.company_id
28 GROUP BY company_id
29 ORDER BY ventas DESC
30 LIMIT 1;
31
```

Result Grid

company_name	company_id	ventas
Eget Ipsum Ltd	b-2398	473.075000

Result 3 x

Output

Action Output

#	Time	Action	Message
2	19:52:28	SELECT * FROM transactions.transaction	587 row(s) returned
3	19:58:16	SELECT DISTINCT company.country FROM company RIGHT JOIN transaction O...	15 row(s) returned
4	20:03:37	SELECT COUNT( DISTINCT company.country) AS Païses_Comprando FROM com...	1 row(s) returned
5	20:06:23	SELECT company.company_name, transaction.company_id, AVG(transaction.amo...	1 row(s) returned

Esta query selecciona el nombre de la compañía (company\_name) de la tabla company y los datos transaction.company\_id y realiza una función para encontrar el promedio (AVG()) de transaction.amount para catalogar este último

como ventas usando alias. se hace un Join de la tabla transaction con la de company y agrupa por el company\_id para recoger todas las transacciones de cada compañía, se ordena por monto de ventas de manera descendente y se limita por 1 para encontrar la empresa con mayor promedio de ventas.

## Exercici 3

**1) Utilitzant només subconsultes (sense utilitzar JOIN):  
Mostra totes les transaccions realitzades per empreses d'Alemanya.**

```

36 • SELECT id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined
37 FROM transactions.transaction
38 WHERE company_id IN (SELECT id
39 FROM transactions.company
40 WHERE country = 'Germany');
41

```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount
10881D1D-5B23-A76C-55EF-C568E49A05DD	CdU-2938	b-2222	275	83.7839	-178.86	2021-07-07 17:43:16	293.57
EA2C3281-C9C1-A387-44F8-729FB4B51C76	CdU-2938	b-2222	275	20.2004	-116.84	2021-05-09 10:25:08	119.36
0DD2E608-5C9E-D1B3-4999-B99F43AD735A	CdU-2959	b-2234	275	9.68811	130.282	2021-04-17 05:30:17	252.47
AB069F53-965E-A2A8-CE06-CA8C4FD92501	CdU-2959	b-2234	275	1.64819	-158.007	2021-04-15 13:37:18	60.99

Message: 118 row(s) returned

Selecciono de la tabla transaction los campos id, credit\_card\_id, company\_id, user\_id, lat, longitude, timestamp, amount, declined (esto último como paso extra para saber si hubo o no hubo venta), filtrando el company\_id que se encuentra IN la subquery. Dentro de la subquery selecciono los id de la tabla company filtrando con WHERE para solo tener como resultado aquellos id que se encuentren ubicados en Alemania.

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

```

• SELECT id, company_name
from company
where (select max(amount)
from transaction
where company_id = company.id) > (
SELECT AVG(amount)
FROM transaction
);

```

id	company_name
2222	Ac Fermentum Incorporated
2226	Magna A Neque Industries
2230	Fusce Corp.
2238	Ante Iaculis Nec Foundation
2242	Donec Ltd
2246	Sed Nunc Ltd

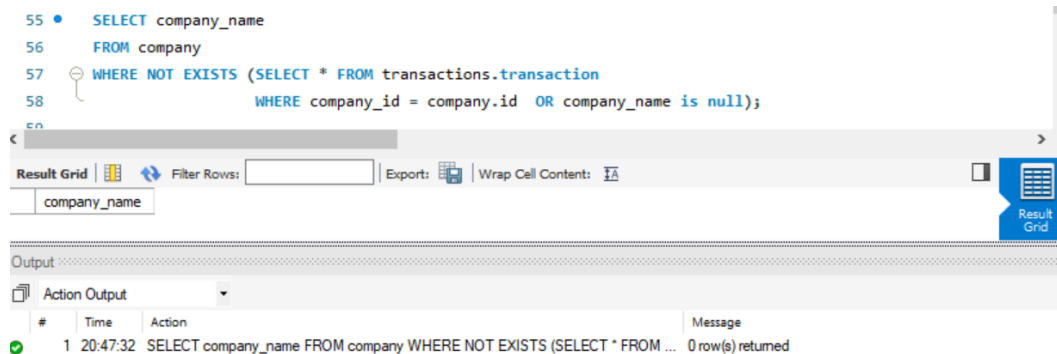
Message: 70 row(s) returned

Para esta consulta usé dos queries anidadas. Esta query busca el id y el company\_name de la tabla company realizando un filtro con WHERE donde para

evitar realizar un JOIN, va una subconsulta que selecciona con una función de cifra máxima (MAX()) el amount de la tabla transaction, realizando de nuevo otro filtro con WHERE donde se indica que transaction.company\_id es igual a company.id y se compara esta subconsulta con el operador mayor que (>) con la última subquery. La query más interna busca el promedio de todas las transacciones de la tabla transaction.

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

```
55 • SELECT company_name
56 FROM company
57 WHERE NOT EXISTS (SELECT * FROM transactions.transaction
58                  WHERE company_id = company.id OR company_name is null);
59
```



The screenshot shows a database query interface. The SQL query is displayed in a text area, and below it, the 'Result Grid' is shown with a single column header 'company\_name'. The 'Output' section at the bottom shows the execution details: a green checkmark, the time '20:47:32', the query text, and the message '0 row(s) returned'.

Acá selecciono el company\_name de la tabla company realizando un filtro con WHERE y escribiendo los operadores NOT y EXISTS (para buscar el nombre de compañía cumpliendo con la condición de no tener transacciones) y escribo una subquery donde selecciono todas las transacciones de la tabla transaction donde (WHERE) company\_id es igual a company.id O (or) company\_name es nulo. El resultado está vacío, lo que indica que no existen compañías sin transacciones registradas.

## 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.

```

65 • SELECT SUM(amount) as ventas, DATE(timestamp) AS dia
66 FROM transaction
67 GROUP BY DATE(timestamp)
68 ORDER BY ventas DESC
69 LIMIT 5;
70

```

ventas	dia
1564.87	2021-03-29
1532.36	2021-12-20
1469.90	2021-06-15
1463.73	2021-05-09
1443.11	2021-06-21

#	Time	Action	Message
4	21:42:13	SELECT SUM(amount) as ventas, DATE(timestamp) AS dia FROM transaction GR...	301 row(s) returned
5	21:43:04	SELECT SUM(amount) as ventas, DATE(timestamp) AS dia FROM transaction GR...	5 row(s) returned

Con esta query realizo una suma de las cantidades con la función SUM() y cambio el formato de la fecha a solo fecha del timestamp con la función DATE, agrupo por fechas, ordeno por ventas de mayor a menor cantidad y limito 5 para tener los 5 días con mayor cantidad en ventas

## Exercici 2

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

```

74 • SELECT company.country, AVG(transaction.amount) as ventas
75 FROM transaction
76 JOIN company
77 ON company.id=transaction.company_id
78 GROUP BY company.country
79 ORDER BY ventas DESC;

```

country	ventas
United States	309.179412
Ireland	277.308387
United Kingdom	270.731700
Canada	269.647869
Sweden	260.615063
Norway	254.216324

#	Time	Action	Message
1	21:49:44	SELECT company.country, AVG(transaction.amount) as ventas FROM transaction J...	15 row(s) returned

Para esta consulta selecciono company.country y hago un promedio con la función AVG() de la amount renombrándolo como ventas. Hago una JOIN de las tablas company y transaction mediante company.id = transaction.company\_id. Luego agrupo por país (company.country) y ordeno por el promedio de forma descendente.

## -- 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.

### 1) Mostra el llistat aplicant JOIN i subconsultes.

```
87 • SELECT transaction.*
88 FROM transaction
89 JOIN company ON transaction.company_id = company.id
90 WHERE company.country = (
91     SELECT country
92     FROM company
93     WHERE company_name = 'Non Institute'
94 );
```

The screenshot shows a database query interface. The query is a JOIN between the 'transaction' table and the 'company' table, filtered by 'company.country' and a subquery for 'company\_name = \'Non Institute\''. The results are displayed in a table with columns: id, credit\_card\_id, company\_id, user\_id, lat, longitude, timestamp, and amount. The first few rows are visible, showing transaction details for various companies.

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount
2B928E1C-EC14-A760-0A75-871477649D6A	CcU-2980	b-2246	275	-41.0496	161.685	2021-08-10 08:14:49	383.73
ACD2011A-A2B1-C365-41E1-2AB00C65147A	CcU-2980	b-2246	275	-54.4792	-82.7974	2022-03-05 20:41:20	60.07
4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	CcU-3092	b-2310	275	-20.4859	150.87	2021-05-03 22:37:23	458.74
BC2B9A38-77B4-28CD-1FE8-14DED863E773	CcU-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95
1479B3D2-87BA-C7BB-4CE3-8D7C2DDE85AB8	CcU-2994	b-2326	133	66.2672	172.399	2021-08-09 00:58:07	309.45

Para esta subconsulta selecciono todos los elementos de la tabla transacción y realizo una join mediante `company.id = transaction.company_id`. Luego filtro `company.country` con un `WHERE` igualándolo a mi subquery, dentro de la cual seleccionará el país de la tabla `company` filtrando también con `WHERE` para igualar el nombre de la compañía a **"Non Institute"**.

### 2) Mostra el llistat aplicant solament subconsultes.

```
97 • SELECT *
98 FROM transaction
99 WHERE company_id IN (
100     SELECT id
101     FROM company
102     WHERE country = (
103         SELECT country
104         FROM company
105         WHERE company_name = 'Non Institute'
106     ));
```

The screenshot shows a database query interface. The query is a subquery-based query that filters 'transaction' by 'company\_id' using a subquery that filters 'company' by 'country' and a further subquery that filters 'company' by 'company\_name = \'Non Institute\''. The results are displayed in a table with columns: id, credit\_card\_id, company\_id, user\_id, lat, longitude, timestamp, and amount. The first few rows are visible, showing transaction details for various companies.

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount
2B928E1C-EC14-A760-0A75-871477649D6A	CcU-2980	b-2246	275	-41.0496	161.685	2021-08-10 08:14:49	383.73
ACD2011A-A2B1-C365-41E1-2AB00C65147A	CcU-2980	b-2246	275	-54.4792	-82.7974	2022-03-05 20:41:20	60.07
4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	CcU-3092	b-2310	275	-20.4859	150.87	2021-05-03 22:37:23	458.74
BC2B9A38-77B4-28CD-1FE8-14DED863E773	CcU-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95

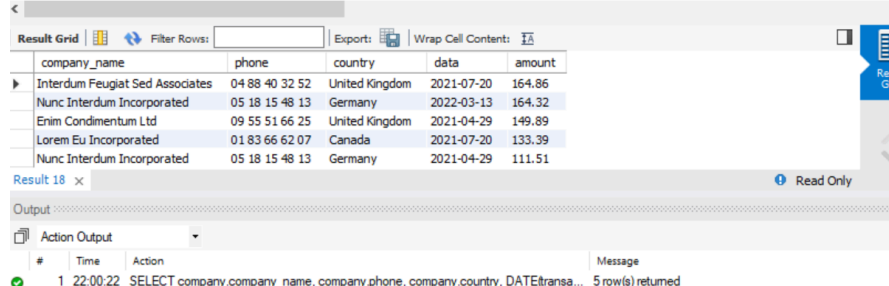
Para esta subconsulta selecciono todos los elementos de la tabla transacción filtrando `company_id` con `WHERE` para que mi `company_id` se encuentre dentro de la subquery. En la subquery interna selecciono la id de la tabla compañía donde `country` sea igualada a la query más interna. En esta última subquery selecciono el país de la tabla `company` filtrando también con `WHERE` para igualar el nombre de la compañía a **"Non Institute"**.

## Nivell 3

### Exercici 1

Presenta el nom, telèfon, país, data i amount, d'aquelles empreses que van realitzar transaccions amb un valor comprès entre 100 i 200 euros i en alguna d'aquestes dates: 29 d'abril del 2021, 20 de juliol del 2021 i 13 de març del 2022. Ordena els resultats de major a menor quantitat.

```
114 • SELECT company.company_name, company.phone, company.country, DATE(transaction.timestamp) AS data,  
115 transaction.amount  
116 FROM company  
117 RIGHT JOIN transaction  
118 ON transaction.company_id = company.id  
119 WHERE transaction.amount BETWEEN 100 AND 200  
120 AND DATE(transaction.timestamp) IN ('2021-04-29','2021-07-20','2022-03-13')  
121 ORDER BY transaction.amount DESC;
```



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

Selecciono company.company\_name, company.phone, company.country y coloco la función DATE() para cambiar el formato de transaction.timestamp, selecciono además transaction.amount. Y hago un RIGHT JOIN de las tablas company y transaction para colocar a los dos últimos campos como la información más importante de mi consulta (tomando todos sus datos con esta join) mediante company.id = transaction.company\_id.

Luego realizo un filtro con WHERE y los operadores BETWEEN y AND para delimitar mi búsqueda de transaction.amount en el rango de 100 A 200.

Uso otro operador AND con la función DATE y el operador IN para encerrar entre paréntesis las fechas que necesito buscar. Uso and porque permite filtrar para que todos los resultados deban cumplir con todas las condiciones.

Finalmente ordeno por transaction.amount de manera Descendente

### Exercici 2

Necessitem optimitzar l'assignació dels recursos i dependrà de la capacitat operativa que es requereixi, per la qual cosa et demanen la informació sobre la quantitat de transaccions que realitzen les empreses, però el departament de recursos humans és exigent i vol un llistat de les empreses on especifiquis si tenen més de 4 transaccions o menys.

```

130 • SELECT company.company_name, COUNT(transaction.id) AS cantidad_transacciones,
131       CASE
132         WHEN COUNT(transaction.id) > 4 THEN 'Más de 4 transacciones'
133         ELSE 'Menos de 4 transacciones'
134       END AS cantidad_transacciones_status
135 FROM company
136 LEFT JOIN transaction ON company.id = transaction.company_id
137 GROUP BY company.company_name
138 ORDER BY cantidad_transacciones;

```

Result Grid

company_name	cantidad_transacciones	cantidad_transacciones_status
Dui Quis Institute	1	Menos de 4 transacciones
Nibh Phasellus Corporation	1	Menos de 4 transacciones
Orci Adipiscing Limited	1	Menos de 4 transacciones
Aliquet Diam Limited	1	Menos de 4 transacciones
Tempor Diam Institute	1	Menos de 4 transacciones

Result 20 x

Output

Action Output

#	Time	Action	Message
1	22:03:49	SELECT company.company_name, COUNT(transaction.id) AS cantidad_transaccion...	100 row(s) returned

Para esta query, debo seleccionar `company.company_name`, y realizar un recuento de `transaction.id` con la función `COUNT()`, que renombro a `cantidad_transacciones`. Utilizo una expresión case para hacer iteraciones con cada fila. La condición a cumplirse es `WHEN` (cuando) el count que seleccioné inicialmente es evaluado con el operador mayor que (`>`) comparado con 4, entonces como resultado se escribirá la frase “mas de 4 transacciones” Si la condición no se cumple se muestra el mensaje “menos de 4 transacciones” Y se terminan las iteraciones con ambos mensajes en la columna `cantidad_transacciones_status`. Se hace un `LEFT JOIN` entre ambas tablas mediante `company.id = transaction.company_id`. Agrupo por `company.company_name` y ordeno por la `cantidad_transacciones`