Repassar les nocions bàsiques per a l'ús de base de dades relacionals. En aquest sprint, iniciaràs la teva experiència pràctica amb una base de dades que conté informació d'una empresa dedicada a la venda de productes en línia. En aquesta activitat, t'enfocaràs en dades relacionades amb les transaccions efectuades i la informació corporativa de les empreses que van participar.

Nivell 1

- 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.

Base de datos relacional de dos tablas una de hechos y una de dimensiones:

- **Tabla de hechos**: Transaction con id (Primary Key), credit_card_id, company_id (Foreign Key), user_id, lat, longitude, timestamp, amount, declined. Contiene datos medibles y dinámicos, es donde ocurren las operaciones principales.

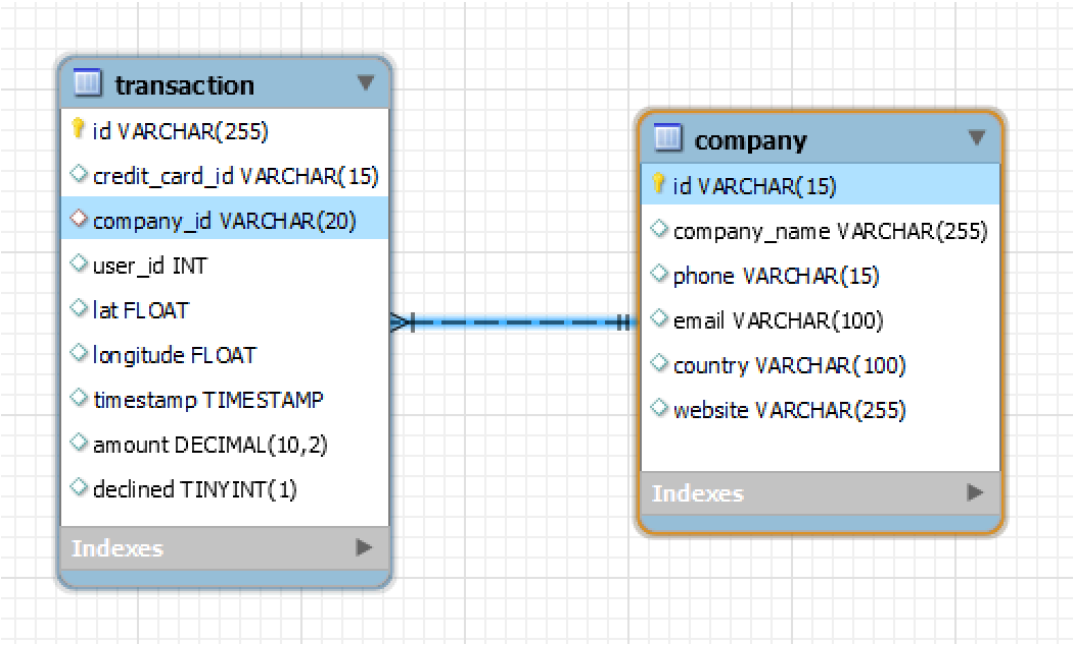
Columna	**Tipo de Dato**	**Características**
id	VARCHAR (255) (PK)	Identificador único de la transacción, clave primaria.
crit_card_id	VARCHAR(150)	Identificador de la tarjeta de crédito. (Información sensible).
company_id	VARCHAR (20) (FK)	Clave foránea que referencia Company(id), establece la relación 1:N con la tabla Company por su id.
user_id	INT	Identificador del usuario que realizó la transacción.
lat	FLOAT	Latitud de la ubicación de la transacción .
longitude	FLOAT	Longitud de la ubicación de la transacción.
timestamp	TIMESTAMP	Fecha y hora exacta de la transacción.
amount	DECIMAL(10,2)	Monto de la transacción con 2 decimales
declined	TINYINT o BOOLEAN (1)	Indica si la transacción fue rechazada (TRUE = rechazada, FALSE = aprobada).

- **Tabla de dimensiones**: Company con id (Primary Key), company_name, phone, email, country, website. Contiene información estática y descriptiva de las empresas, para contextualizar los datos de la tabla de hechos.

Columna	**Tipo de Dato**	**Características**
id	VARCHAR (15) (PK)	Identificador único, clave primaria.

company_name	VARCHAR(255)	Nombre de la empresa, texto de longitud variable (hasta 255 caracteres).
phone	VARCHAR(15)	Teléfono de la empresa, texto de longitud variable (hasta 255 caracteres).
email	VARCHAR(100)	Correo electrónico de la empresa. (Información sensible).
country	VARCHAR(100)	País de la empresa.
website	VARCHAR(255)	URL del sitio web de la empresa

Las dos tablas se relacionan Company 1 a N Transaction a través de la PK de la tabla Company, la columna id y la FK de la tabla Transaction, la columna company_id. Una empresa puede tener muchas transacciones, pero cada transacción sólo tiene una única empresa. La tabla Transaction es la tabla de hechos mientras que la tabla Company es de dimensiones.



Tipos de datos:
VARCHAR(n): Tipo de dato para almacenar cadenas de texto de longitud variable, donde n define el número máximo de caracteres permitidos.
INT: Tipo de dato para almacenar números enteros (positivos o negativos) sin decimales.
FLOAT: Tipo de dato para almacenar números con decimales, pero con precisión variable, lo que puede generar pequeños errores de redondeo.
TINYINT(1): Tipo de dato para almacenar números enteros muy pequeños (de -128 a 127 si es con signo, o de 0 a 255 si es sin signo). Se usa comúnmente para valores booleanos (1 = verdadero, 0 = falso).
TIMESTAMP: Tipo de dato para almacenar fechas y horas en un formato compacto, basado en el número de segundos desde 1970-01-01 00:00:00 UTC (Epoch). Ejemplo: 2025-03-10 14:30:00.

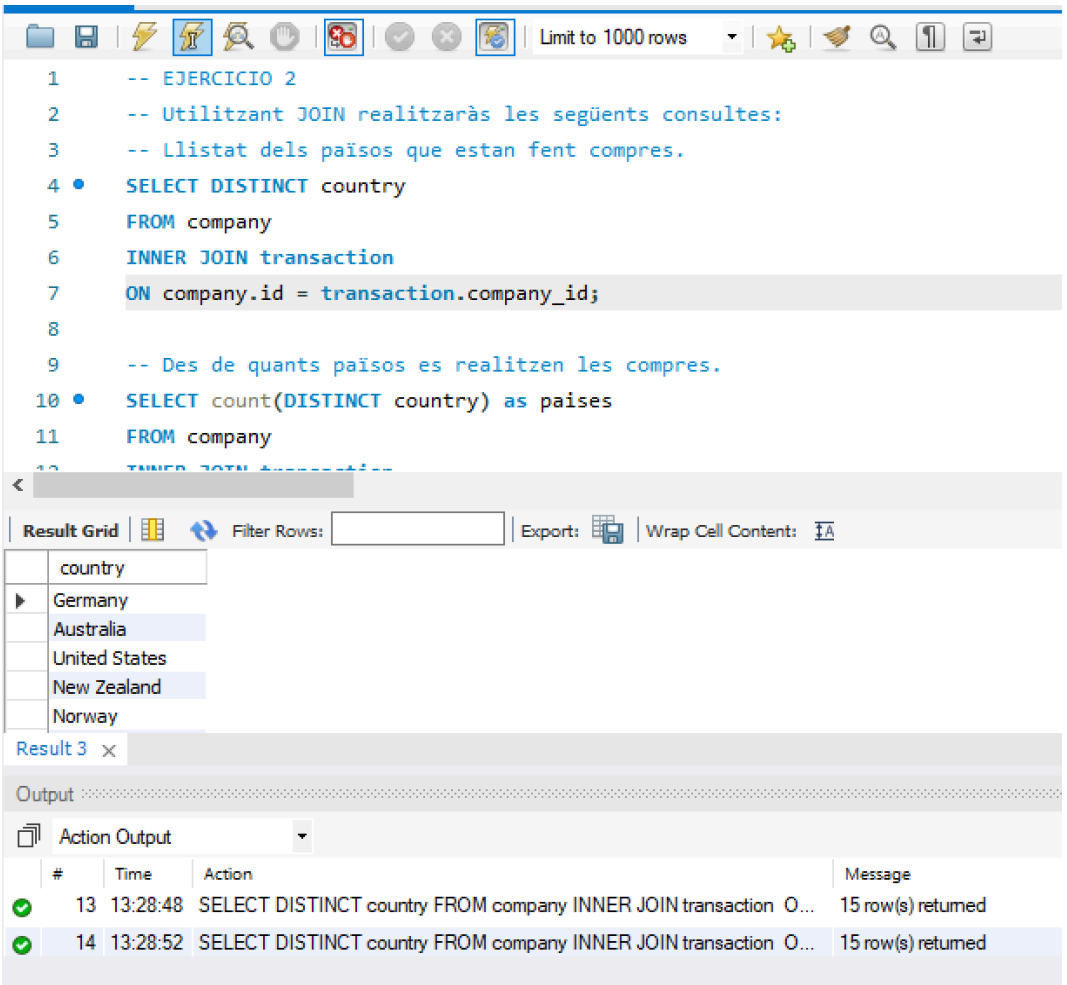
Exercici 2

Utilitzant JOIN realitzaràs les següents consultes:

- *Llistat dels països que estan fent compres.*

Seleccionamos países uniendo las dos tablas por id de la compañía. Utilizamos distinct para evitar duplicados.

SELECT DISTINCT country
FROM company
INNER JOIN transaction
ON company.id = transaction.company_id;

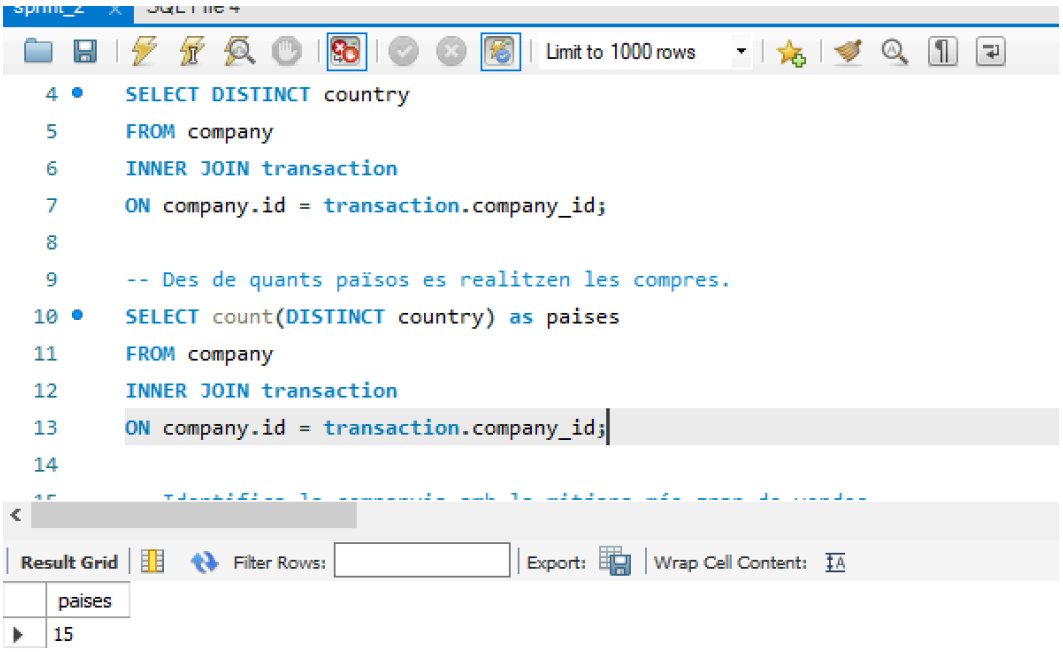


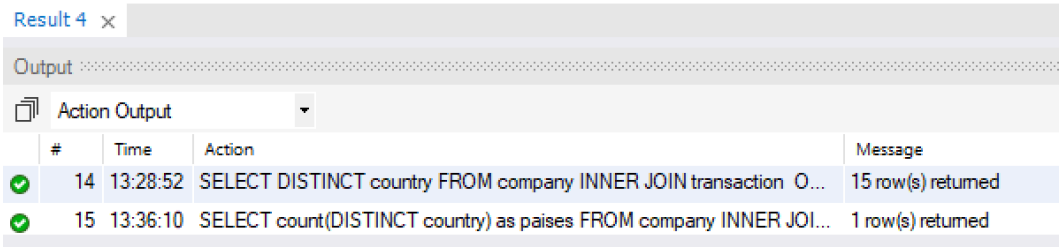
- *Des de quants països es realitzen les compres.*

Contamos la cantidad de países con distinct para evitar duplicados, donde se están realizando compras. De nuevo unimos las dos tablas por el id de la compañía.

SELECT count(DISTINCT country) as paises
FROM company
INNER JOIN transaction

ON company.id = transaction.company_id;





- *Identifica la companyia amb la mitjana més gran de vendes.*

Hacemos la media del monto de dinero agrupado por el nombre e id de la compañía y ordenamos por el resultado de la media descendentemente y limitamos al primer resultado que es el más alto. La media la redondeamos a 2 decimales, ya que se trata de dinero y filtramos por declined para descartar aquellas transacciones rechazadas.

```
SELECT company_id, company_name, ROUND(AVG(amount),2) as media_ventas_usd
FROM company
INNER JOIN transaction
ON company.id = transaction.company_id
WHERE declined = 0
GROUP BY company_id, company_name
ORDER BY media_ventas_usd DESC
LIMIT 1;
```

```
-- Identifica la companyia amb la mitjana més gran de vendes.
SELECT company_id, company_name, ROUND(AVG(amount),2) as media_ventas_usd
FROM company
INNER JOIN transaction
ON company.id = transaction.company_id
WHERE declined = 0
GROUP BY company_id, company_name
ORDER BY media_ventas_usd DESC
LIMIT 1;
```

Result Grid

company_id	company_name	media_ventas_usd
b-2398	Eget Ipsum Ltd	481.86

Result 5

Output

Action Output

#	Time	Action	Message
15	13:36:10	SELECT count(DISTINCT country) as paises FROM company INNER JOI...	1 row(s) returned
16	13:40:39	SELECT company_id, company_name, ROUND(AVG(amount),2) as media...	1 row(s) returned

Exercici 3

Utilitzant només subconsultes (sense utilitzar JOIN):

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

Seleccionamos transacción, id de la compañía y el amount para mostrar todas las transacciones y el amount de cada transacción filtrado por el país Alemanía. Utilizamos una subconsulta para acceder a la información de ambas tablas sin el uso de join y poder hacer el filtrado.

```
SELECT transaction.id, company_id, amount as amount_usd
FROM transaction
WHERE company_id IN (SELECT id FROM company WHERE country = 'Germany');
```

```
23      LIMIT 1;
24
25      -- EJERCICIO 3
26      -- Utilitzant només subconsultes (sense utilitzar JOIN):
27      -- Mostra totes les transaccions realitzades per empreses d'Alemanya.
28      SELECT transaction.id, company_id, amount as amount_usd
29      FROM transaction
30      WHERE company_id IN (SELECT id FROM company WHERE country = 'Germany');
31
32      -- Llista les empreses que han realitzat transaccions per un amount superior a la
33      -- Opció 1 on es veuen els noms de les empreses
```

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

id	company_id	amount_usd
108B1D1D-5B23-A76C-55EF-C568E49A05DD	b-2222	293.57
EA2C3281-C9C1-A387-44F8-729FB4B51C76	b-2222	119.36
0DD2E608-5C9E-D1B3-4999-B99F43AD735A	b-2234	252.47
AB069F53-965E-A2A8-CE06-CA8C4FD92501	b-2234	60.99
0466A42E-47CF-8D24-FD01-C0B689713128	b-2302	49.53

transaction 6

Output

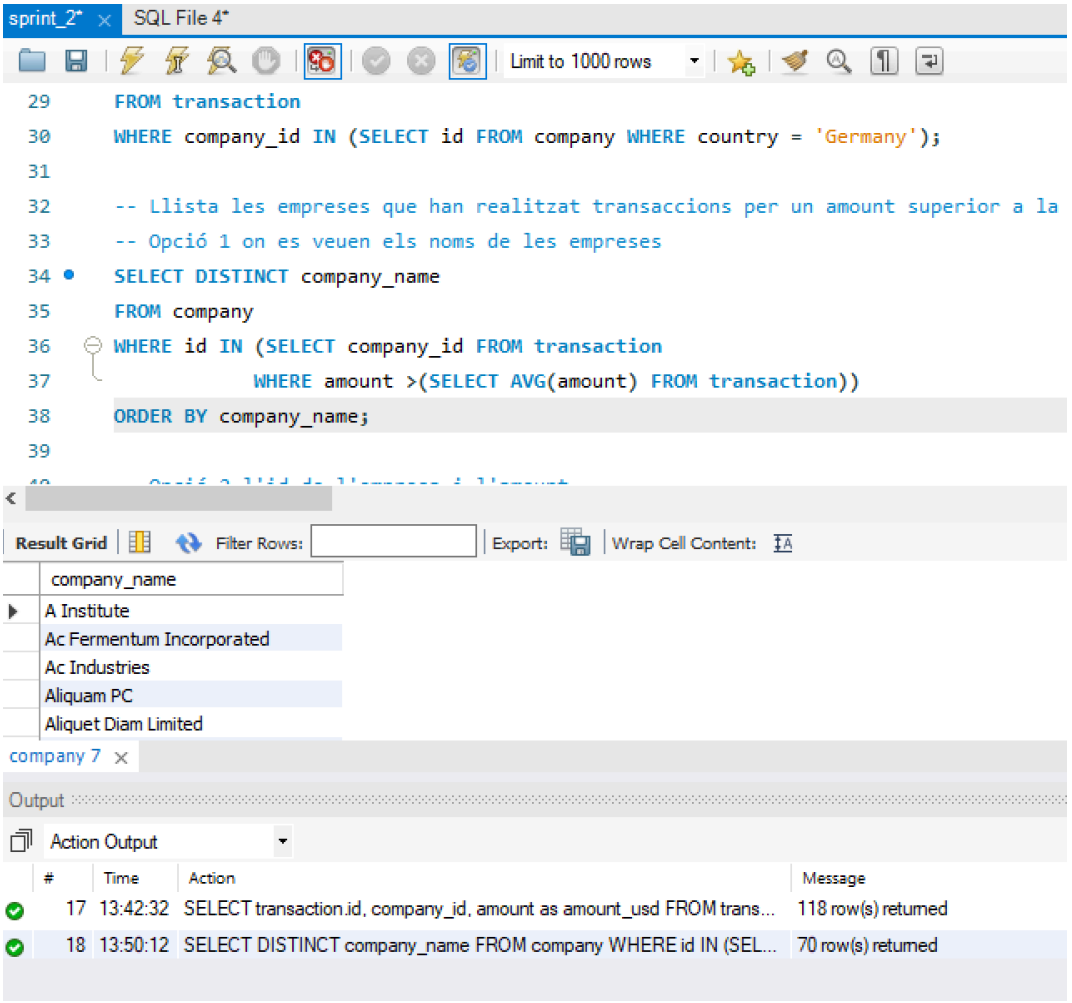
Action Output

#	Time	Action	Message
16	13:40:39	SELECT company_id, company_name, ROUND(AVG(amount),2) as media...	1 row(s) returned
17	13:42:32	SELECT transaction.id, company_id, amount as amount_usd FROM trans...	118 row(s) returned

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

Mostramos los nombres de las empresas que hayan realizado transacciones con un amount superior a la media. Utilizamos un distinct evitar duplicados. Ordenamos por nombre de compañía. de nuevo utilizamos una subconsulta para poder acceder a ambas tablas y poder hacer el filtrado.

```
SELECT DISTINCT company_name
FROM company
WHERE id IN (SELECT company_id FROM transaction
                    WHERE amount >(SELECT AVG(amount) FROM transaction))
ORDER BY company_name;
```



- *Eliminaran del sistema les empreses que no tenen transaccions registrades, entrega el llistat d'aquestes empreses*
-

Opción 1: seleccionamos el nombre de la compañía para mostrar en pantalla con distinct para evitar duplicados y filtramos con el id de la compañía utilizando el operador “is not” en la tabla de transacciones para filtrar las empresas que no tienen transacciones.

```
SELECT DISTINCT company_name
FROM company
WHERE company.id NOT IN (SELECT company_id FROM transaction);
```

```
ORDER BY company_name;

-- Eliminaran del sistema les empreses que no tenen transaccions registrac
-- Opción 1: Lo haría así, pero no me da resultado pero he comparobado y e
-- Todas las compañías tienen transacciones registradas
SELECT company_name
FROM company
WHERE company.id NOT IN (SELECT company_id FROM transaction);

-- Opción 2: Nulls en amount tampoco devuelve nada
SELECT company_name
```

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

company_name

company 12

Output

Action Output

#	Time	Action	Message
23	13:54:58	SELECT company_name FROM company WHERE company.id NOT IN (...	0 row(s) returned
24	13:56:36	SELECT company_name FROM company WHERE company.id NOT IN (...	0 row(s) returned

Opción 2: Igual que el anterior, seleccionamos el nombre de la compañía con distinct para evitar duplicados y filtramos el id de la compañía en la tabla de transaction, pero aquí buscando nulls o valores vacíos.

SELECT DISTINCT company_name
FROM company
WHERE id IN (SELECT company_id FROM transaction WHERE amount IS NULL OR amount = '');

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

-- Opción 2: Nulls en amount tampoco devuelve nada
SELECT company_name
FROM company
WHERE id IN (SELECT company_id FROM transaction
                WHERE amount IS NULL OR amount = '');

-- Opción 3
SELECT DISTINCT company_name
```

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

company_name

company 13

Output

Action Output

#	Time	Action	Message
24	13:56:36	SELECT company_name FROM company WHERE company.id NOT IN (...	0 row(s) returned
25	13:58:58	SELECT company_name FROM company WHERE id IN (SELECT compa...	0 row(s) returned

Opción 3: Igual que en los casos anteriores mostramos el nombre de la compañía evitando duplicados y filtramos por el id de la compañía, pero utilizando como operador el NOT EXISTS para buscar las empresas que no tienen transacciones.

SELECT DISTINCT company_name
FROM company
WHERE NOT EXISTS (SELECT 1 FROM transaction WHERE company_id = company.id);

```
WHERE id IN (SELECT company_id FROM transaction
                WHERE amount IS NULL OR amount = '');

-- Opción 3
SELECT DISTINCT company_name
FROM company
WHERE NOT EXISTS (SELECT 1 FROM transaction
                  WHERE company_id = company.id);

-- NIVELL 2
-- Exercici 1
```

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

company_name

company 14

Output

Action Output

#	Time	Action	Message
25	13:58:58	SELECT company_name FROM company WHERE id IN (SELECT compa...	0 row(s) returned
26	14:02:01	SELECT DISTINCT company_name FROM company WHERE NOT EXIS...	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.

Opción 1: Seleccionamos la fecha utilizando el método DATE para separar la fecha de la hora en timestamp, sumamos el amount, filtramos por las transacciones rechazadas, agrupamos por fecha y ordenamos por la suma del amount. Limitamos 5 para mostrar en pantalla los días con más cantidad de ingresos.

```
SELECT DATE(timestamp) AS fecha, sum(amount) as ingresos
FROM transaction
WHERE declined = 0
GROUP BY fecha
ORDER BY ingresos DESC
LIMIT 5;
```

```
-- NIVELL 2
-- Exercici 1
-- Identifica els cinc dies que es va generar la quantitat més gran d'ing
-- Opción 1 total de ingreso por dia:
SELECT DATE(timestamp) AS fecha, sum(amount) as ingresos
FROM transaction
WHERE declined = 0
GROUP BY fecha
ORDER BY ingresos DESC
LIMIT 5;
```

Result Grid

fecha	ingresos
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 52

Output

Action Output

#	Time	Action	Message
67	14:06:03	SHOW columns FROM company	6 row(s) returned
68	14:06:13	SELECT DATE(timestamp) AS fecha, sum(amount) as ingresos FROM tran...	5 row(s) returned

Opción 2: igual que en el caso anterior separamos la fecha de la hora, pero en este caso también seleccionamos la id de transacción y hacemos la suma del amount en una subconsulta para mostrar la suma del amount por fecha y por transacción de los 5 días con más ingresos.

```
SELECT DATE(timestamp) AS fecha, transaction.id, amount
FROM transaction
INNER JOIN (SELECT DATE(timestamp) AS fecha, SUM(amount) AS ingresos
   FROM transaction
   GROUP BY fecha
   ORDER BY ingresos DESC
   LIMIT 5) dias_top
ON DATE(timestamp) = dias_top.fecha
ORDER BY dias_top.fecha, timestamp;
```

```
-- Opción 2 total de ingreso por dia por transacción:
SELECT DATE(timestamp) AS fecha, transaction.id, amount
FROM transaction
INNER JOIN (SELECT DATE(timestamp) AS fecha, SUM(amount) AS ingresos
    FROM transaction
    GROUP BY fecha
    ORDER BY ingresos DESC
    LIMIT 5) dias_top
ON DATE(timestamp) = dias_top.fecha
ORDER BY dias_top.fecha, timestamp;
```

Result Grid

fecha	id	amount
2021-03-29	167DE938-B1C2-428D-FA3D-96A45CD41B22	414.36
2021-03-29	3ED634C2-01CA-D6E9-2DBD-24B53A7912F7	478.54
2021-03-29	8438C66C-4964-D73D-6839-24F2CDAA5828	239.75
2021-03-29	D3C579AE-FA77-6234-9843-6DA39D6E329D	432.22
2021-05-09	79B621A9-3FAA-41E4-2845-3DE6532C743D	400.63

Result 53

Output

Action Output

#	Time	Action	Message
68	14:06:13	SELECT DATE(timestamp) AS fecha, sum(amount) as ingresos FROM tran...	5 row(s) returned
69	14:07:02	SELECT DATE(timestamp) AS fecha, transaction.id, amount FROM transa...	22 row(s) returned

Exercici 2

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

Seleccionamos los países y la media redondeada y limitada a 2 decimales, filtramos por las transacciones rechazadas. Agrupamos por país y ordenamos descendiente para mostrar en pantalla de mayor a menor. Utilizamos una join para unir las tablas a través de la id de las compañías.

```
SELECT country, ROUND(AVG(amount),2) as media_ventas_usd
FROM transaction
INNER JOIN company
ON transaction.company_id = company.id
WHERE declined = 0
GROUP BY country
ORDER BY media_ventas_usd DESC;
```

```
-- Exercici 2
-- Quina és la mitjana de vendes per país? Presenta els resultats ordenat
SELECT country, ROUND(AVG(amount),2) as media_ventas_usd
FROM transaction
INNER JOIN company
ON transaction.company_id = company.id
WHERE declined = 0
GROUP BY country
ORDER BY media_ventas_usd DESC;

-- Exercici 3
```

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

country	media_ventas_usd
United States	287.53
Ireland	285.83
Sweden	276.67
United Kingdom	271.77
Canada	261.94

Result 54

Output

Action Output

#	Time	Action	Message
69	14:07:02	SELECT DATE(timestamp) AS fecha, transaction.id, amount FROM transa...	22 row(s) returned
70	16:20:27	SELECT country, ROUND(AVG(amount),2) as media_ventas_usd FROM t...	15 row(s) returned

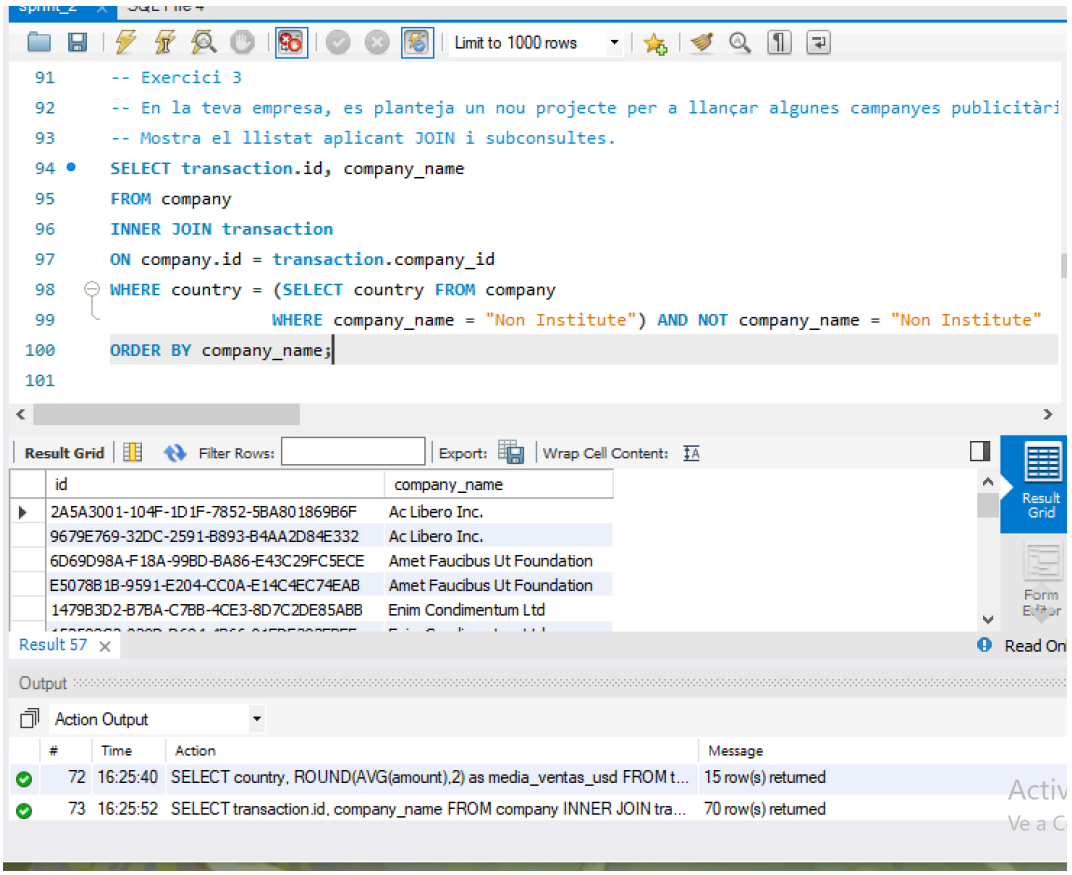
Exercici 3

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.*

Seleccionamos la id de transición y el nombre de la compañía, hacemos una join entre las dos tablas por id de compañía y filtramos primero el país de la compañía “Non Institute” y luego para que muestre los nombres de las compañías que no son “Non Institute” para ver cuales son las compañías que están en el mismo país, pero que en pantalla no se muestren las de la empresa competidora. Ordenamos por nombre de compañía.

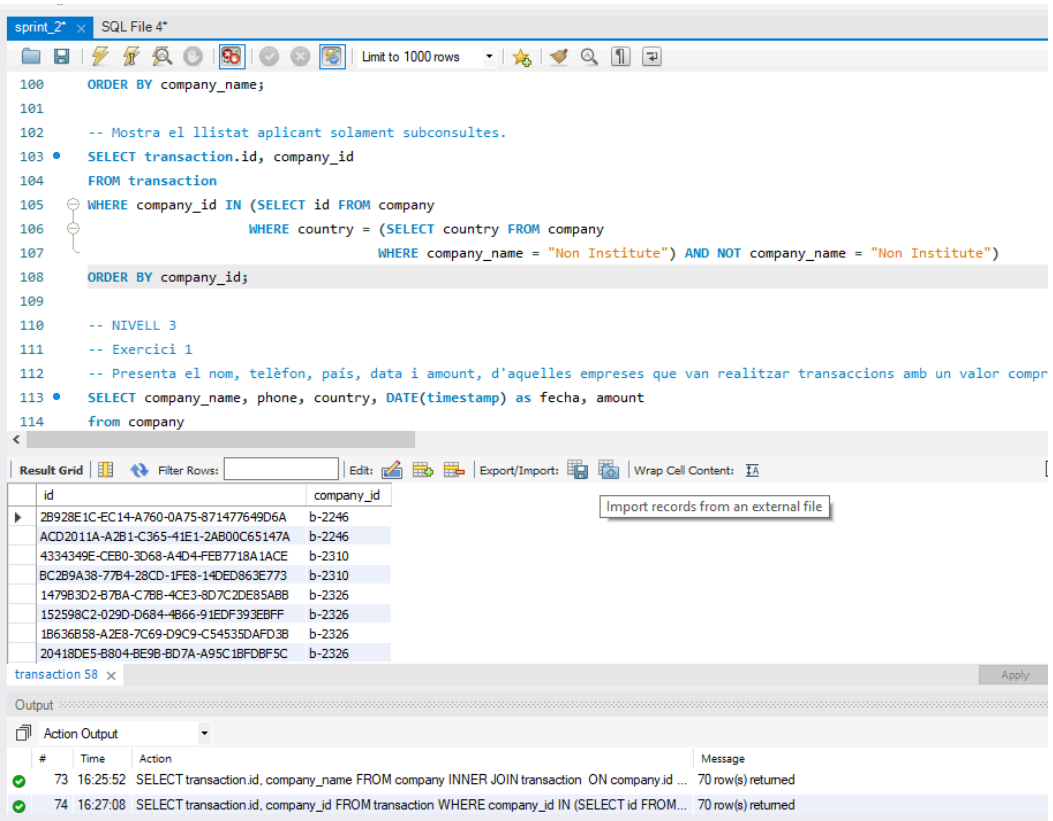
```
SELECT transaction.id, company_name
FROM company
INNER JOIN transaction
ON company.id = transaction.company_id
WHERE country = (SELECT country FROM company
                 WHERE company_name = "Non Institute") AND NOT company_name = "Non
                 Institute"
ORDER BY company_name;
```



- *Mostra el llistat aplicant solament subconsultes.*

Lo mismo que en el ejercicio anterior, pero en vez de con una join utilizamos una subconsulta para hacer el filtrado con los datos de la otra tabla utilizando el operador IN.

```
SELECT transaction.id, company_id
FROM transaction
WHERE company_id IN (SELECT id FROM company
                     WHERE country = (SELECT country FROM company
                                      WHERE company_name = "Non Institute") AND NOT
                                      company_name = "Non Institute")
ORDER BY company_id;
```



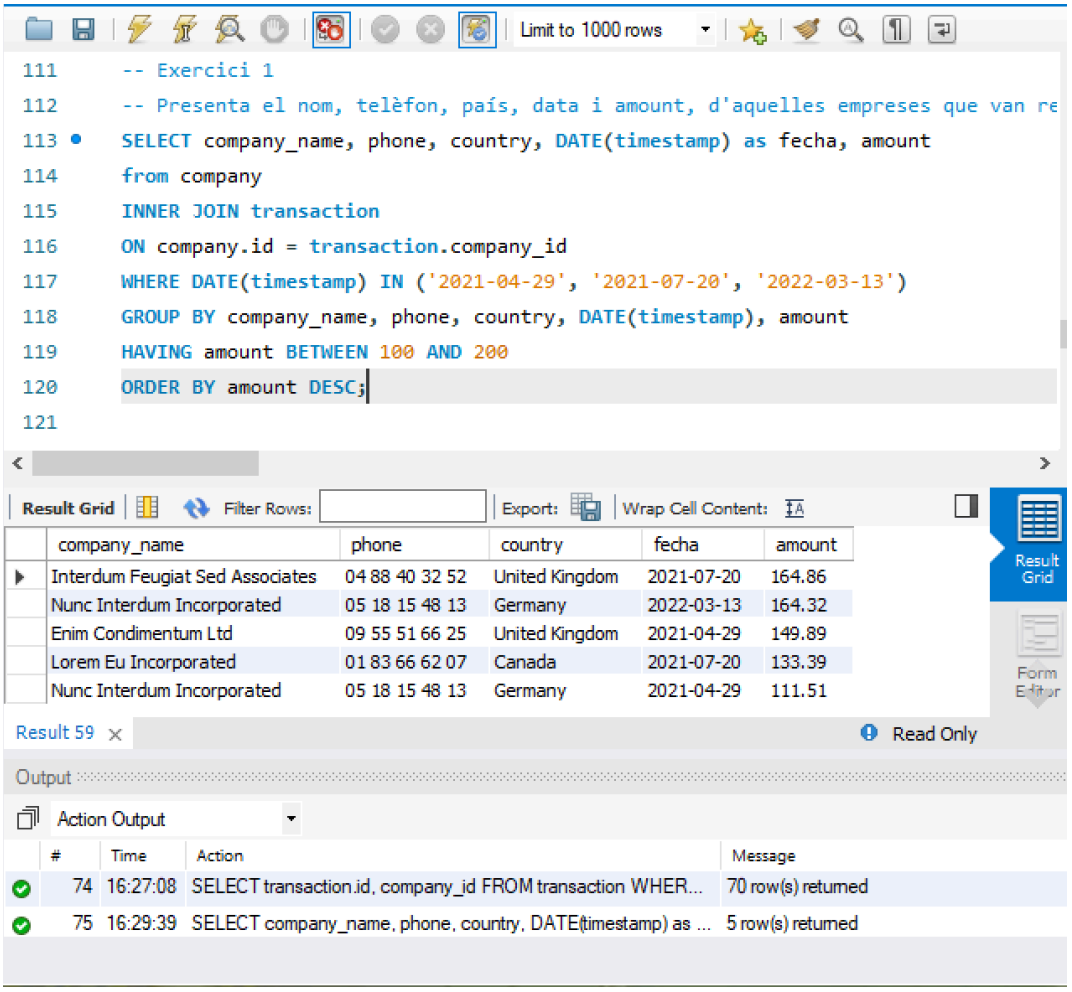
Nivell 3

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.

Seleccionamos el nombre de la empresa, el teléfono, la fecha separada de las horas en el timestamp y el amount. Unimos las tablas por la id de las compañías y filtramos por las fechas que se solicitan. También filtramos por los amounts que están entre 100 y 200. Se ha hecho con un Having, pero puede hacerse con otro WHERE. Se ordena de forma descendente por amount.

```
SELECT company.company_name, company.phone, company.country, DATE(transaction.timestamp)
AS fecha, transaction.amount
FROM company
INNER JOIN transaction ON company.id = transaction.company_id
WHERE transaction.amount BETWEEN 100 AND 200 AND DATE(transaction.timestamp) IN
('2021-04-29', '2021-07-20', '2022-03-13')
ORDER BY transaction.amount DESC;
```



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.

Se selecciona el nombre de la compañía, hacemos un LEFT JOIN para asegurarnos de que aparezcan todas las transacciones con el id de las compañías. Hacemos un recuento de las transacciones para filtrar y separar entre más de 4 y menos de 4 utilizando el CASE WHEN que nos mostrará una nueva columna con los resultados del filtrado. Agrupamos por el nombre de la compañía y ordenamos por la cantidad de transacciones siendo primero las que son menos de 4 y luego las de más. Si hicieramos DESC será a la inversa.

```
SELECT company.company_name,
  CASE
    WHEN COUNT(transaction.id) > 4 THEN 'Más de 4'
    ELSE 'Menos de 4'
  END AS cantidad_transacciones
FROM company
LEFT JOIN transaction ON company.id = transaction.company_id
GROUP BY company.id, company.company_name
ORDER BY cantidad_transacciones;
```

```
-- Necessitem optimitzar l'assignació dels recursos i dependrà de la capaci
-- Opción 1
SELECT company_name,
    CASE
        WHEN COUNT(transaction.id) > 4 THEN 'Más de 4'
        ELSE 'Menos de 4'
    END AS cantidad_transacciones
FROM company
LEFT JOIN transaction ON company.id = transaction.company_id
GROUP BY company_name
ORDER BY cantidad_transacciones;
```

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

company_name	cantidad_transacciones
Enim Condimentum Ltd	Más de 4
Nunc Interdum Incorporated	Más de 4
Ut Semper Foundation	Más de 4
Lorem Eu Incorporated	Más de 4
Arcu LLP	Más de 4

Result 60 | Read Only

Output

Action Output

#	Time	Action	Message
76	16:30:56	SELECT company_name, total_transacciones FROM (SELECT ...	Error Code: 1248. Every derived ta
77	16:31:09	SELECT company_name, CASE WHEN COUNT(transa...	100 row(s) returned