

Tasca S2.01. Nocions bàsiques SQL

Fernando Poblete

Objetivo:

Repasar 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.

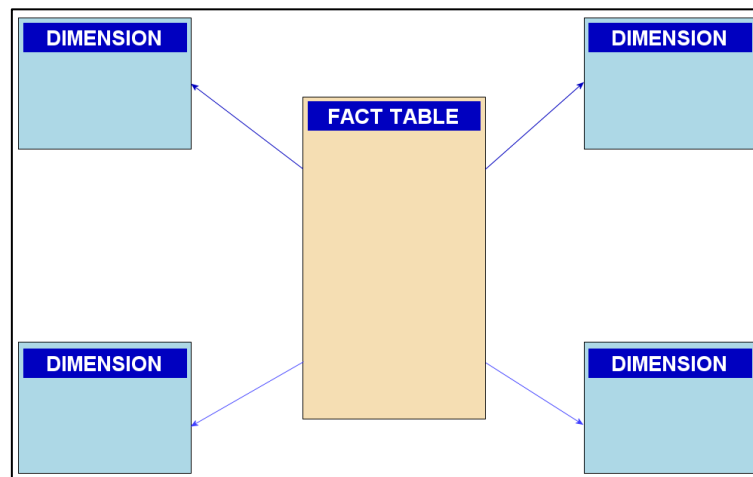
- Para crear la base de datos, primero ejecutamos el archivo 'estructura_dades.sql'. En este archivo se crea la base de datos llamada 'transactions', junto con sus dos tabla, 'company' y 'transaction'.

- Se especifica para cada variable, el tipo de dato que contiene.

- Además se especifica si una variable será una Llave Primaria, o si será una Llave Secundaria, especificando a que Llave Primaria de otra tabla hace referencia.

- Después para cargar los datos ejecutamos el archivo 'dades_introduir.sql'. Allí se introducen datos en las tablas anteriormente creadas. En este caso tenemos 100 registros para la tabla 'company' y 587 registros para la tabla 'transaction'.

- Aunque solamente contamos con 2 tablas, podemos observar que el modelo de datos relacionales sigue el esquema de estrella. La tabla de 'transacciones' es la tabla de hechos, teniendo un gran volumen de datos, y llaves secundarias que pueden conectan a las llaves primarias de diversas tablas de dimensiones, como 'credit_card', 'user' y 'company'. En este ejemplo, solo contamos con la tabla de 'company'



Esquema Diagrama de estrella - Wikipedia

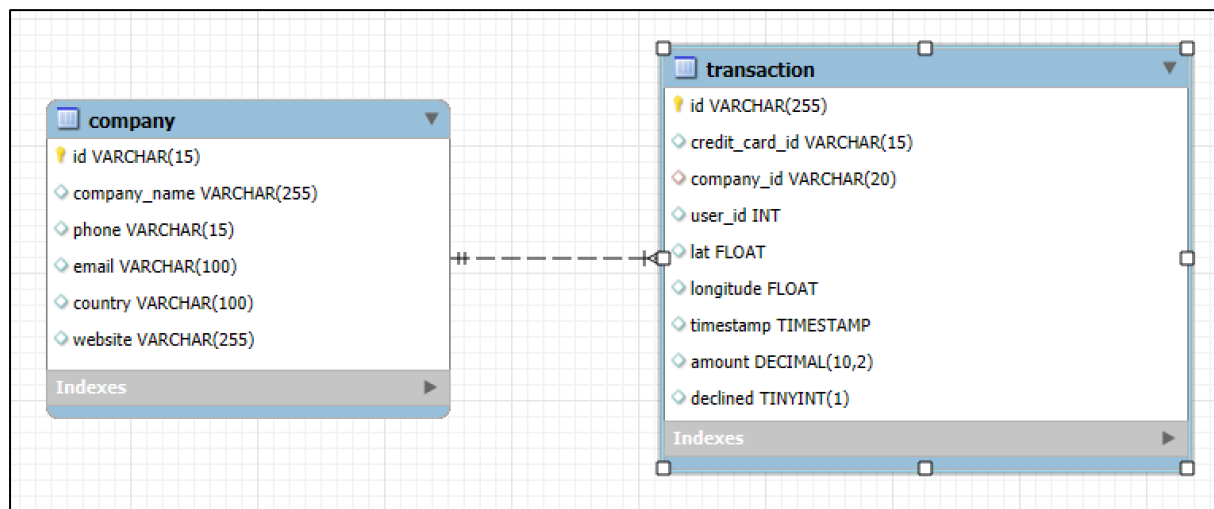


Diagrama entidad relacion

- En el diagrama entidad-relación, o modelo relacional, podemos observar que las tablas 'company' y 'transaction' tienen cada una, un primary key, que identifica los registros en cada tabla de manera única.

- Así mismo, la tabla de transacciones se conecta a la de compañía mediante el 'company_id', que en la tabla de 'company' simplemente se llama 'id'.

- La relación entre 'company' y 'transaction', es de 1:N, porque una compañía puede tener múltiples transacciones, mientras que cada transacción, solo puede ser asignada a 1 compañía.

- A continuación, un listado de las variables de cada tabla con una breve descripción de cada variable:

Variable	Tipo de dato	Descripción
Id	VARCHAR(15)	PRIMARY KEY
company_name	VARCHAR(255)	Nombre de la compañía
Phone	VARCHAR(15)	Telefono
email	VARCHAR(100)	email
country	VARCHAR(100)	pais
website	VARCHAR(255)	Sitio Web

Descripción Tabla 'company'

Variable	Tipo de dato	Descripción
id	VARCHAR(255)	PRIMARY KEY
credit_card_id	VARCHAR(15)	REFERENCES credit_card(id)
company_id	VARCHAR(20)	FOREIGN KEY (company_id) REFERENCES company(id)
user_id	INT	REFERENCES user(id)
lat	FLOAT	Coordenadas latitud
longitude	FLOAT	Coordenadas longitud
timestamp	TIMESTAMP	Tiempo de registro
amount	DECIMAL(10, 2)	Valor de la transacción
declined	BOOLEAN	Valor 1= TRUE -> Transacción declinada

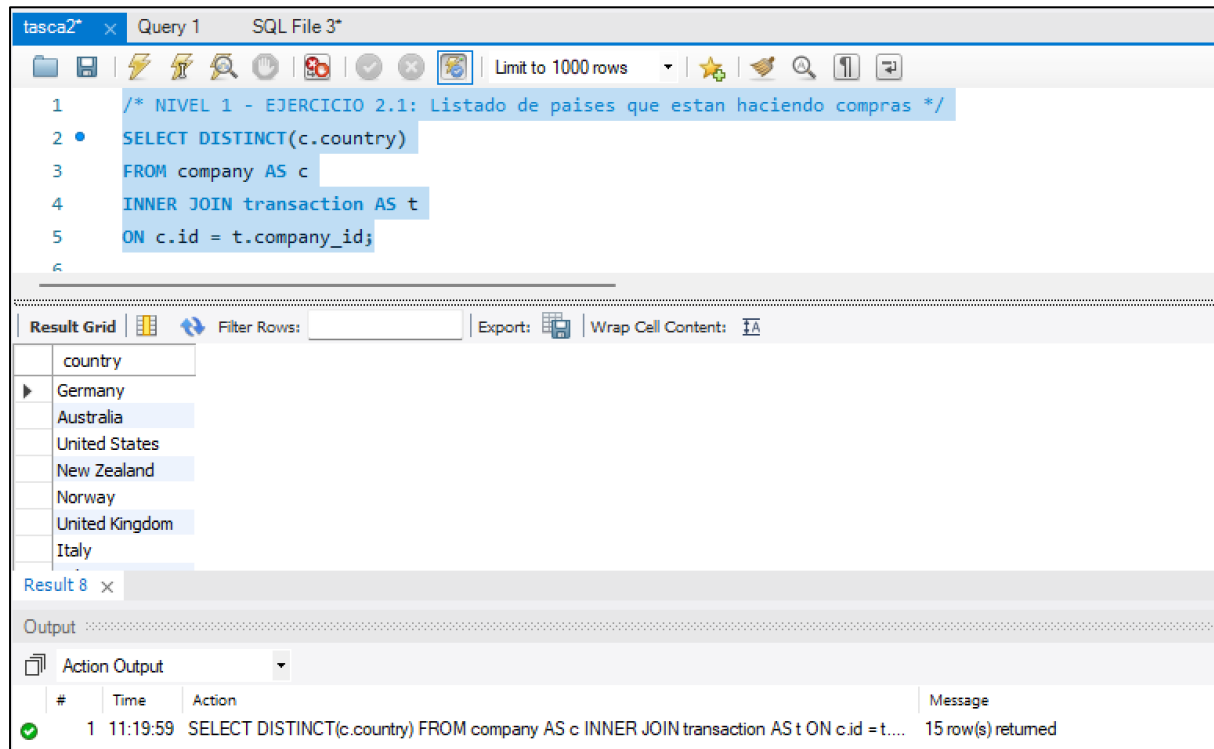
Descripción Tabla 'transaction'

- Exercici 2

Utilitzant JOIN realitzaràs les següents consultes:

- Llistat dels països que estan fent compres.

- Hacemos un INNER JOIN entre la tabla de compañías y de transacciones, para obtener solamente los datos de compañías que tengan transacciones registradas. Luego obtenemos el listado de países sin repeticiones utilizando DISTINCT en la selección de datos.



The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
1 /* NIVEL 1 - EJERCICIO 2.1: Listado de paises que estan haciendo compras */
2 • SELECT DISTINCT(c.country)
3 FROM company AS c
4 INNER JOIN transaction AS t
5 ON c.id = t.company_id;
6
```

The results pane displays the following data:

country
Germany
Australia
United States
New Zealand
Norway
United Kingdom
Italy

The Action Output pane shows the following message:

#	Time	Action	Message
1	11:19:59	SELECT DISTINCT(c.country) FROM company AS c INNER JOIN transaction AS t ON c.id = t....	15 row(s) returned

- Des de quants països es realitzen les compres.

- Para obtener la cantidad de países que realizan compras, simplemente utilizamos la función de conteo, COUNT, en la selección de países sin repetición.

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query is as follows:

```

6
7  /* NIVEL 1 - EJERCICIO 2.2: Cantidad de paises que estan haciendo compras */
8  • SELECT COUNT(DISTINCT(c.country)) AS Cantidad_Paises_Compras
9    FROM company AS c
10   INNER JOIN transaction AS t
11   ON c.id = t.company_id;
12

```

The result grid shows the following data:

Cantidad_Paises_Compras
15

Below the result grid, there is an 'Output' section with a dropdown menu set to 'Action Output'. The output log shows the following message:

#	Time	Action	Message
1	12:24:11	SELECT COUNT(DISTINCT(c.country)) AS Cantidad_Paises_Compras FROM company AS c I...	1 row(s) returned

- Identifica la compañía con la mitjana més gran de vendes.

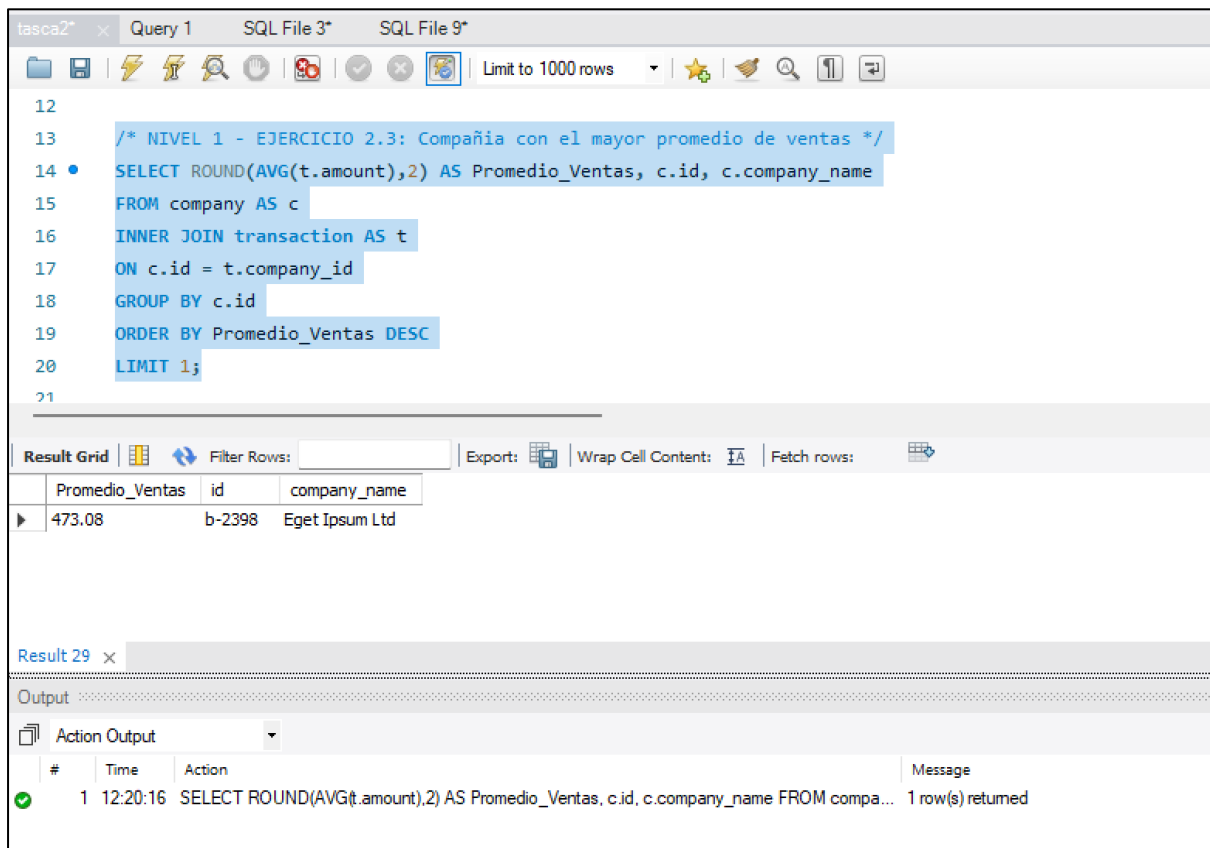
- Para obtener la compañía con la media más grande de ventas, tenemos que extraer el promedio de los montos de las transacciones, con la función AVG sobre la variable 'amount'

- Conservamos la estructura de los JOINS, y agrupamos por el 'id' de la compañía.

- Mostramos tanto el 'id' de la compañía como el 'company_name', sabiendo que la relación de estas dos variables siempre es 1:1.

- Redondeamos el resultado a 2 decimales por claridad, ordenamos los resultados de forma descendente en el promedio calculado, y limitamos los resultados a 1 línea, para obtener la compañía con el mayor valor en esta nueva variable creada.

- La compañía con mayor media de ventas es: 'Eget Ipsum Ltd', con 'id' de compañía 'b-2398'.



The screenshot shows a SQL IDE interface with a query editor and a results grid. The query is as follows:

```

12
13 /* NIVEL 1 - EJERCICIO 2.3: Compañía con el mayor promedio de ventas */
14 • SELECT ROUND(AVG(t.amount),2) AS Promedio_Ventas, c.id, c.company_name
15 FROM company AS c
16 INNER JOIN transaction AS t
17 ON c.id = t.company_id
18 GROUP BY c.id
19 ORDER BY Promedio_Ventas DESC
20 LIMIT 1;
21

```

The results grid shows the following data:

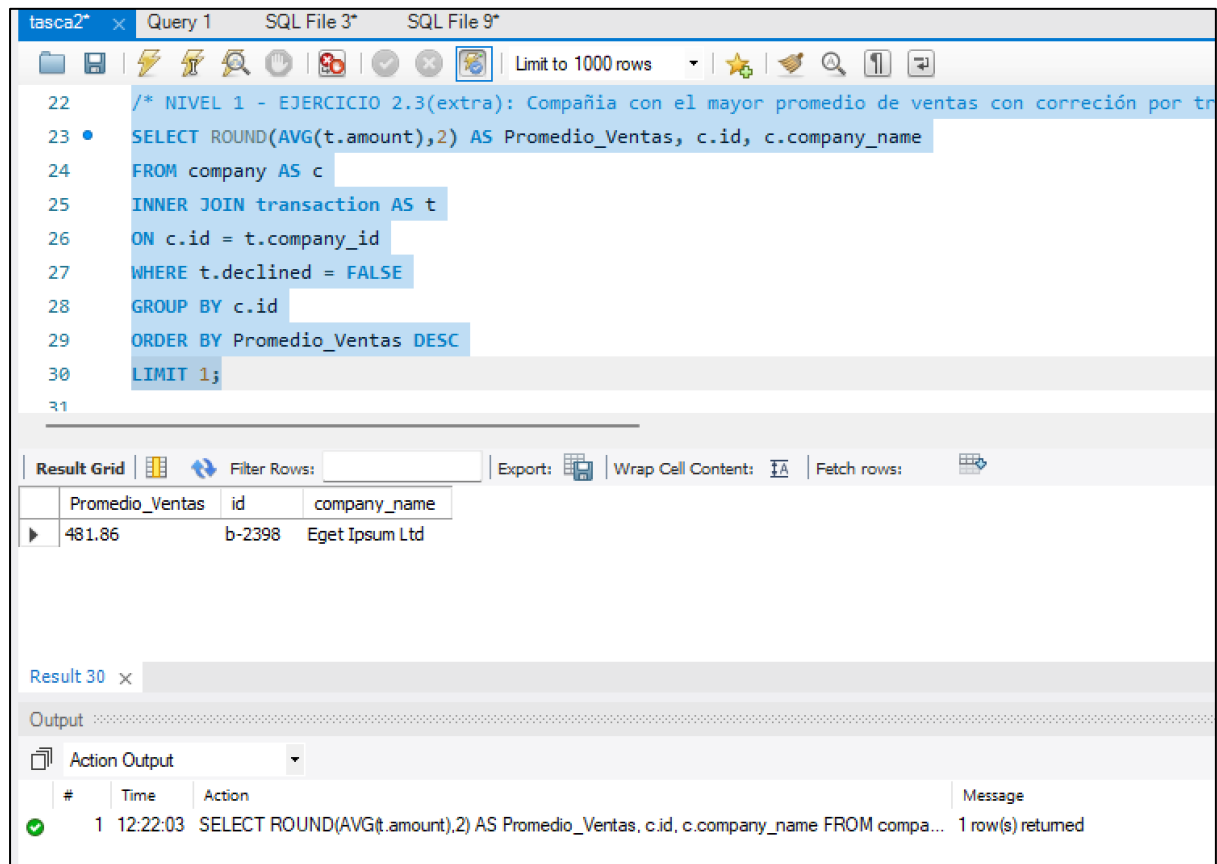
Promedio_Ventas	id	company_name
473.08	b-2398	Eget Ipsum Ltd

Below the results grid, there is an 'Output' section with a table showing the execution details:

#	Time	Action	Message
1	12:20:16	SELECT ROUND(AVG(t.amount),2) AS Promedio_Ventas, c.id, c.company_name FROM compa...	1 row(s) returned

(extra)

- Finalmente verificamos los resultados solamente para las transacciones no declinadas. Vemos que aunque el valor de la media cambia, la compañía con la media más alta sigue siendo la misma.



The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
22  /* NIVEL 1 - EJERCICIO 2.3(extra): Compañía con el mayor promedio de ventas con corrección por tr
23  • SELECT ROUND(AVG(t.amount),2) AS Promedio_Ventas, c.id, c.company_name
24  FROM company AS c
25  INNER JOIN transaction AS t
26  ON c.id = t.company_id
27  WHERE t.declined = FALSE
28  GROUP BY c.id
29  ORDER BY Promedio_Ventas DESC
30  LIMIT 1;
```

The results pane shows a single row of data:

Promedio_Ventas	id	company_name
481.86	b-2398	Eget Ipsum Ltd

Below the results pane, the 'Output' section shows the execution message:

#	Time	Action	Message
1	12:22:03	SELECT ROUND(AVG(t.amount),2) AS Promedio_Ventas, c.id, c.company_name FROM compa...	1 row(s) returned

- Exercici 3

Utilitzant només subconsultes (sense utilitzar JOIN):

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

- Primero, mediante una subconsulta, creamos una tabla que contenga todos los 'id' de las compañías que se encuentren en Alemania (country = 'Germany').

- Después, seleccionamos toda la información de la tabla de transacciones, y agregamos la condición, con la clausula WHERE, que me entregue solamente resultados donde el 'company_id' se encuentre en (IN) la tabla creada en el primera parte, que contiene solo identificadores de compañías en Alemania.

- Así, sin utilizar JOINS, obtenemos todos los datos de transacciones exclusivamente para compañías en Alemania.

The screenshot shows a SQL IDE window with a query editor and a results grid. The query is as follows:

```
31
32  /* NIVEL 1 - EJERCICIO 3.1: Transacciones realizadas por empresas de Alemania - Sin Usar JOINS */
33  SELECT *
34  FROM transaction AS t
35  WHERE t.company_id IN
36  (
37    SELECT c.id
38    FROM company AS c
39    WHERE c.country = 'Germany'
40  )
41
```

The results grid displays the following data:

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-D1B3-4999-899F43AD735A	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

The output section shows the following message:

```
1 11:23:30 SELECT * FROM transaction AS t WHERE t.company_id IN ( SELECT c.id FROM company A... 118 row(s) returned
```


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

- En este ejercicio vamos a hacer consultas anidadas en 3 niveles.

- Primero, en una subconsulta en la tabla 'transaction', calculamos el valor promedio de todas las transacciones.

- Luego utilizamos este valor calculado, como parámetro en una clausula WHERE, para filtrar los registros de la tabla 'transaction', extrayendo únicamente los identificadores de empresa 'company_id', sin repeticiones (las repeticiones se pueden eliminar también en el próximo nivel). Con repeticiones obtenemos 297 registros, y al agregar DISTINCT al SELECT, obtenemos solo 70 registros.

- Finalmente, para obtener los nombres de estas empresas y no sus códigos, utilizamos el resultado del paso anterior, de la misma manera que en el ejercicio 3.1, para extraer los nombres de las compañías, condicionando en la cláusula WHERE, que sean solamente compañías cuyos 'id' se encuentren en el listado creado en el paso anterior.

The screenshot shows a SQL IDE window titled 'tasca2*' with a query editor and a results panel. The query editor contains the following SQL code:

```

41
42  /* NIVEL 1 - EJERCICIO 3.2: Lista de empresas que han realizado transacciones con un monto superior a al pro
43  • SELECT c.company_name
44  FROM company as c
45  WHERE c.id IN
46  (
47  SELECT DISTINCT t.company_id
48  FROM transaction AS t
49  WHERE t.amount >
50  (
51  SELECT AVG(amount) FROM transaction
52  )
53  );
54

```

The results panel shows a table with the following data:

company_name
Ac Fermentum Incorporated
Magna A Neque Industries
Fusce Corp.
Ante Iaculis Nec Foundation
Donec Ltd
Sed Nunc Ltd
Nascetur Ridiculus Mus Inc.

The output panel shows the following message:

```

1 11:24:14 SELECT c.company_name FROM company as c WHERE c.id IN ( SELECT DISTINCT t.comp... 70 row(s) returned

```

(extra)

- De manera adicional, si queremos ser más exactos, podemos omitir las transacciones declinadas, especificando mediante una cláusula WHERE en el primero nivel, que se consideren solamente los montos de transacciones aceptadas (declined = FALSE).

Esto es importante para poder realizar decisiones de negocios, ya que las transacciones declinadas no representan ingresos para la compañía.

Al hacer esta corrección, la cantidad de empresas que han realizado transacciones disminuye de 70 a 68 registros.

The screenshot shows a SQL IDE window titled 'tasca2*' with a tab 'Query 1' and 'SQL File 3*'. The query editor contains the following SQL code:

```
55  /* NIVEL 1 - EJERCICIO 3.2 (extra): Lista de empresas que han realizado transacciones con un monto
56  • SELECT c.company_name
57  FROM company as c
58  WHERE c.id IN
59  (
60  SELECT DISTINCT t.company_id
61  FROM transaction AS t
62  WHERE t.amount >
63  (
64  SELECT AVG(amount) FROM transaction AS t
65  WHERE t.declined = FALSE
66  )
67  )
68
```

The 'Result Grid' shows the following data:

company_name
Ac Fermentum Incorporated
Magna A Neque Industries
Fusce Corp.
Ante Iaculis Nec Foundation
Donec Ltd
Sed Nunc Ltd
Nascetur Ridiculus Mus Inc.

The 'Output' section shows the 'Action Output' for the query:

#	Time	Action	Message
1	11:25:40	SELECT c.company_name FROM company as c WHERE c.id IN (SELECT DISTINCT t.comp...	68 row(s) returned

- Eliminaran del sistema les empreses que no tenen transaccions registrades, entrega el llistat d'aquestes empreses.
- Primero, extraemos un listado de 'id_company' sin repetición de la tabla de transacciones.
- Luego utilizaremos el resultado anterior, como una subconsulta, para filtrar mediante una cláusula WHERE, las empresas que no se encuentren en el listado, con el operador NOT IN.
- No obtenemos registros, lo que quiere decir, que todas las 100 compañías de la tabla 'company' tienen transacciones registradas en la tabla 'transactions'

The screenshot shows a SQL IDE window with a query editor and a results panel. The query editor contains the following SQL code:

```

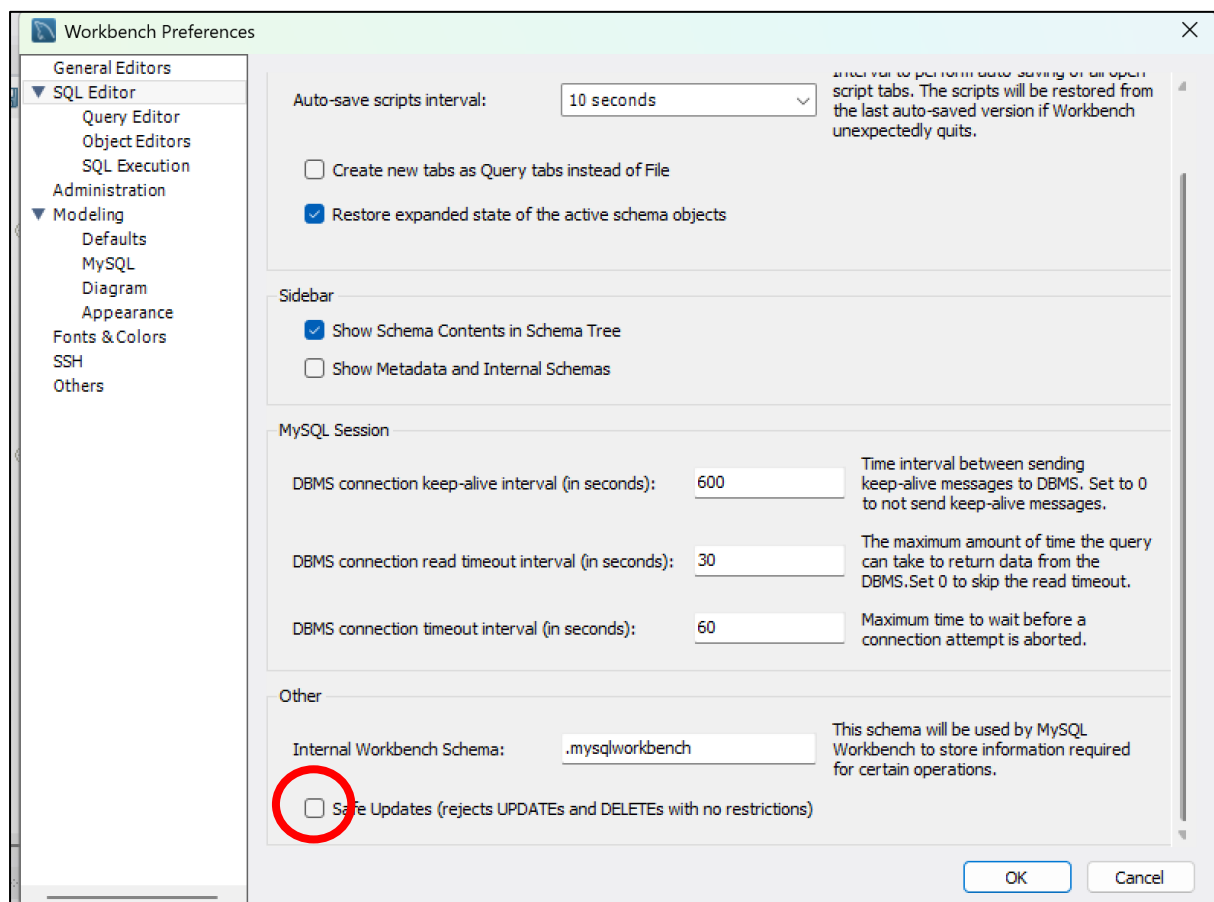
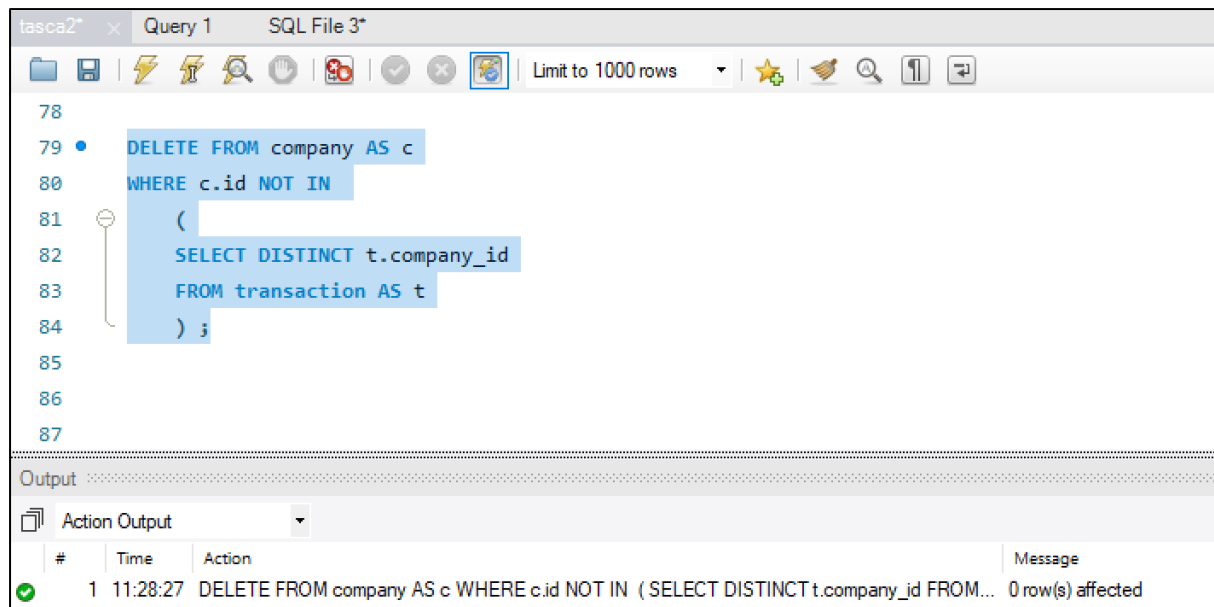
68
69  /* NIVEL 1 - EJERCICIO 3.3: Identificación y eliminacion de empresas que no tienen transacciones regi
70
71  SELECT c.company_name
72  FROM company AS c
73  WHERE c.id NOT IN
74  (
75      SELECT DISTINCT t.company_id
76      FROM transaction AS t
77  ) ;
78

```

The results panel shows a table with one column, 'company_name', and one row with the value 'company 16'. Below the table, the 'Output' section shows the execution details:

#	Time	Action	Message
1	11:26:37	SELECT c.company_name FROM company AS c WHERE c.id NOT IN (SELECT DISTINCT t....	0 row(s) returned

- Como era de esperarse, al ejecutar el código para eliminar las empresas sin transacciones, ningún registro se ve afectado.
- Incluimos para futuras referencias, un pantallazo de la opción que se debe cambiar para poder eliminar datos de una tabla (recordar reconectarse después de cambiar esta opción).

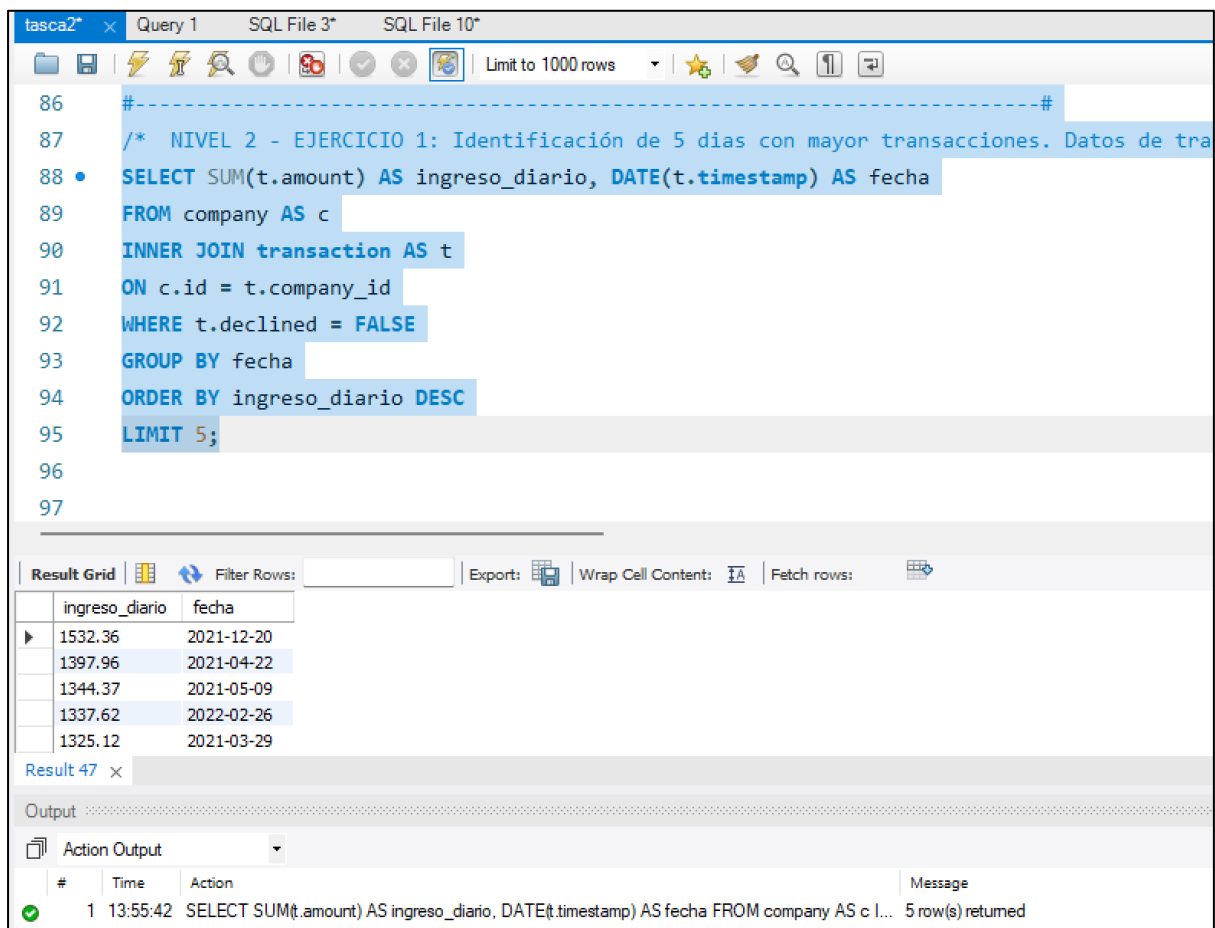


Nivell 2

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.

- Para obtener las ventas diarias, debemos extraer la agregación de valores de 'amount' que indican los ingresos de la compañía con la función SUM(), y agruparlos por día.
- Como no contamos con la variable de 'día' debemos crearla, a partir de la variable timestamp, que es una versión extensa de tiempo. Aplicamos la función DATE().
- Nuevamente consideramos en el análisis solamente las transacciones que no han sido declinadas.



The screenshot shows a SQL IDE interface with a query editor and a results pane. The query is as follows:

```
86 #-----#
87 /* NIVEL 2 - EJERCICIO 1: Identificación de 5 días con mayor transacciones. Datos de tra
88 • SELECT SUM(t.amount) AS ingreso_diario, DATE(t.timestamp) AS fecha
89 FROM company AS c
90 INNER JOIN transaction AS t
91 ON c.id = t.company_id
92 WHERE t.declined = FALSE
93 GROUP BY fecha
94 ORDER BY ingreso_diario DESC
95 LIMIT 5;
96
97
```

The results pane shows a table with two columns: 'ingreso_diario' and 'fecha'. The data is as follows:

ingreso_diario	fecha
1532.36	2021-12-20
1397.96	2021-04-22
1344.37	2021-05-09
1337.62	2022-02-26
1325.12	2021-03-29

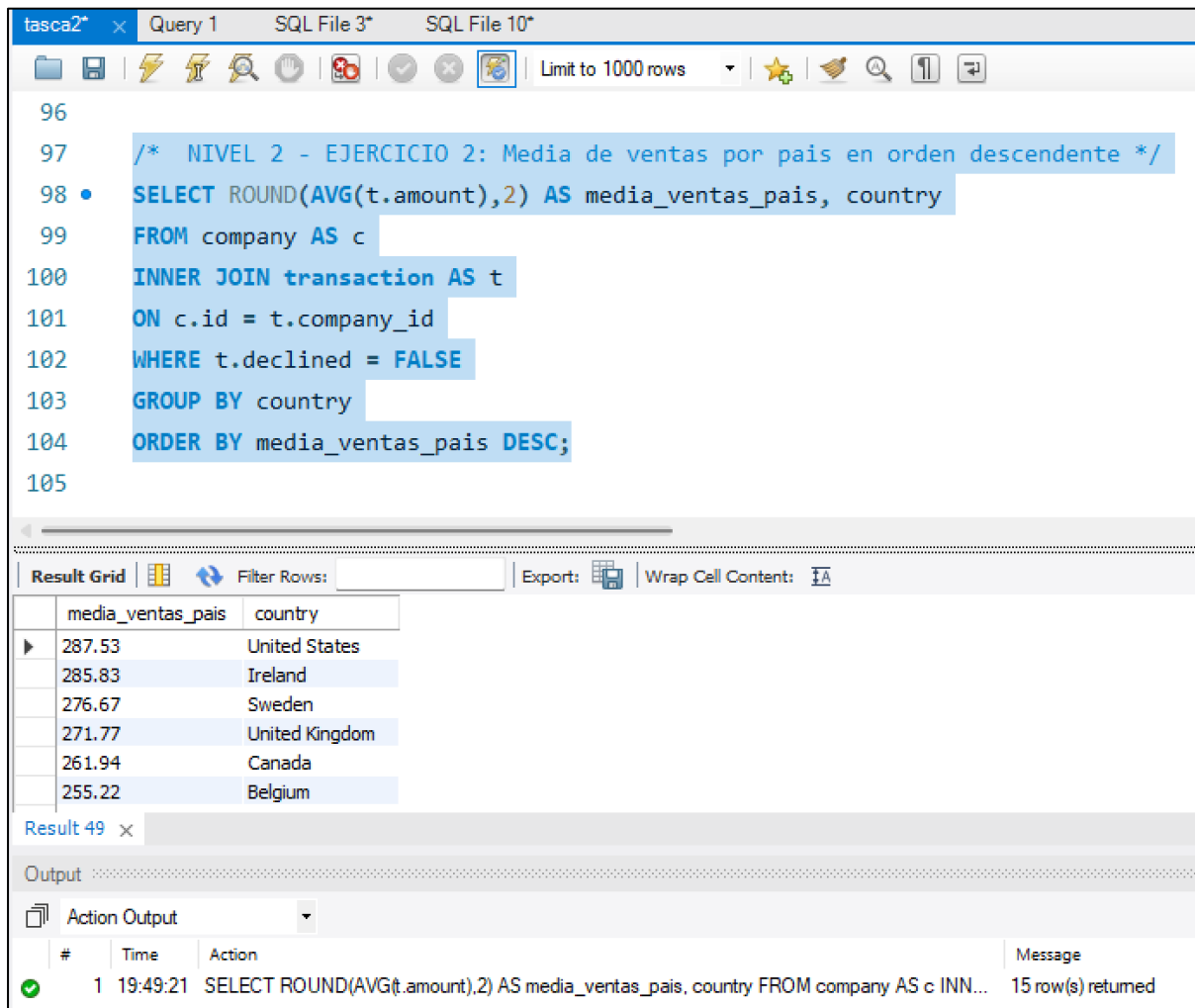
The output pane shows the following message:

```
1 13:55:42 SELECT SUM(t.amount) AS ingreso_diario, DATE(t.timestamp) AS fecha FROM company AS c l... 5 row(s) returned
```

Exercici 2

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

- Para obtener la media de ventas por país, calculamos el promedio de ventas con la función AVG sobre 'amount', y agrupamos los datos por país. Nuevamente excluimos las transacciones declinadas.



The screenshot shows a SQL IDE interface with a query editor and a result grid. The query is as follows:

```
96
97  /* NIVEL 2 - EJERCICIO 2: Media de ventas por pais en orden descendente */
98  • SELECT ROUND(AVG(t.amount),2) AS media_ventas_pais, country
99  FROM company AS c
100  INNER JOIN transaction AS t
101  ON c.id = t.company_id
102  WHERE t.declined = FALSE
103  GROUP BY country
104  ORDER BY media_ventas_pais DESC;
105
```

The result grid displays the following data:

media_ventas_pais	country
287.53	United States
285.83	Ireland
276.67	Sweden
271.77	United Kingdom
261.94	Canada
255.22	Belgium

The output section shows the execution of the query:

#	Time	Action	Message
1	19:49:21	SELECT ROUND(AVG(t.amount),2) AS media_ventas_pais, country FROM company AS c INN...	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.

- Para ambos casos, decidimos incluir las transacciones declinadas, por si presentan valor para el análisis competitivo.

- Primero obtenemos de que país es la compañía 'Non Institute', directamente desde la base de datos, haciendo un SELECT de 'country' de la tabla 'company'.

- Luego de unir las tablas con un JOIN, extraemos las transacciones de la tabla 'transaction', especificando en su clausula WHERE, que el país coincida con lo obtenido en el paso anterior, que usamos como subconsulta.

The screenshot shows a SQL IDE interface with a query editor and a results grid. The query is as follows:

```
/* NIVEL 2 - EJERCICIO 3: Análisis competitivo a empresa "Non Institute USANDO JOINS Y SUBCONSULTAS */
SELECT t.*
FROM transaction AS t
INNER JOIN company AS c
ON c.id = t.company_id
WHERE country =
(
SELECT c.country
FROM company as c
WHERE company_name = 'Non Institute'
);
```

The results grid displays the following data:

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
▶	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
	1479B3D2-B7BA-C7BB-4CE3-8D7C2DE85A8B	CcU-2994	b-2326	133	66.2672	172.399	2021-08-09 00:58:07	309.45	0
	152598C2-029D-D684-4B66-91EDF393EBFF	CcU-2994	b-2326	126	-67.0189	-141.672	2021-07-05 03:10:00	395.43	0
	1B636B58-A2E8-7C69-D9C9-C5453DAFD3B	CcU-2994	b-2326	131	70.2543	-13.1336	2021-07-06 08:48:46	195.06	0
	20418DE5-B804-BE9B-BD7A-A95C1BFD8F5C	CcU-2994	b-2326	126	-79.1145	1.51481	2022-01-03 15:59:29	479.52	0
	239B8576-6C0E-137A-C2F6-3180A188A2D3	CcU-2994	b-2326	126	23.6174	137.222	2021-08-26 06:04:05	43.90	0

The output section shows the following message:

```
1 21:16:38 SELECT t.* FROM transaction AS t INNER JOIN company AS c ON c.id = t.company_id WHE... 100 row(s) returned
```

- Muestra el llistat aplicant solament subconsultes.

- Primero obtenemos de que país es la compañía 'Non Institute', directamente desde la base de datos, haciendo un SELECT de 'country' de la tabla 'company'.

- Luego obtenemos un listado de los 'id' de todas las compañías que se encuentran en ese país. Para eso, extraemos los 'id' de la tabla 'company', indicando en una cláusula WHERE, que solamente queremos los 'id' de compañías cuyo 'country' sea igual al obtenido en la consulta anterior.

- Finalmente en la tabla de transacciones, extraemos el listado completo de columnas, condicionando en la clausula WHERE que se incluyan solamente aquellos registros que incluyen un 'company_id' que se incluya en el listado obtenido en el paso anterior.

- Así con las consultas anidadas, obtenemos las transacciones realizadas solamente en el Reino Unido, país donde se encuentra la compañía 'Non Institute'.

- Al comparar ambos métodos, el uso de JOINS parece mucho más eficiente, conectando la información a través de las diferentes tablas con mayor facilidad.

The screenshot shows a SQL IDE window with a query editor and a result grid. The query is as follows:

```

110 • SELECT t.*
111 FROM transaction AS t
112 WHERE company_id IN
113 (
114     SELECT c.id
115     FROM company AS c
116     WHERE c.country =
117     (
118         SELECT c.country
119         FROM company as c
120         WHERE company_name = 'Non Institute'
121     )
122 );
  
```

The result grid displays the following data:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
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
1479B3D2-B7BA-C7BB-4CE3-8D7C2DE85ABB	CcU-2994	b-2326	133	66.2672	172.399	2021-08-09 00:58:07	309.45	0

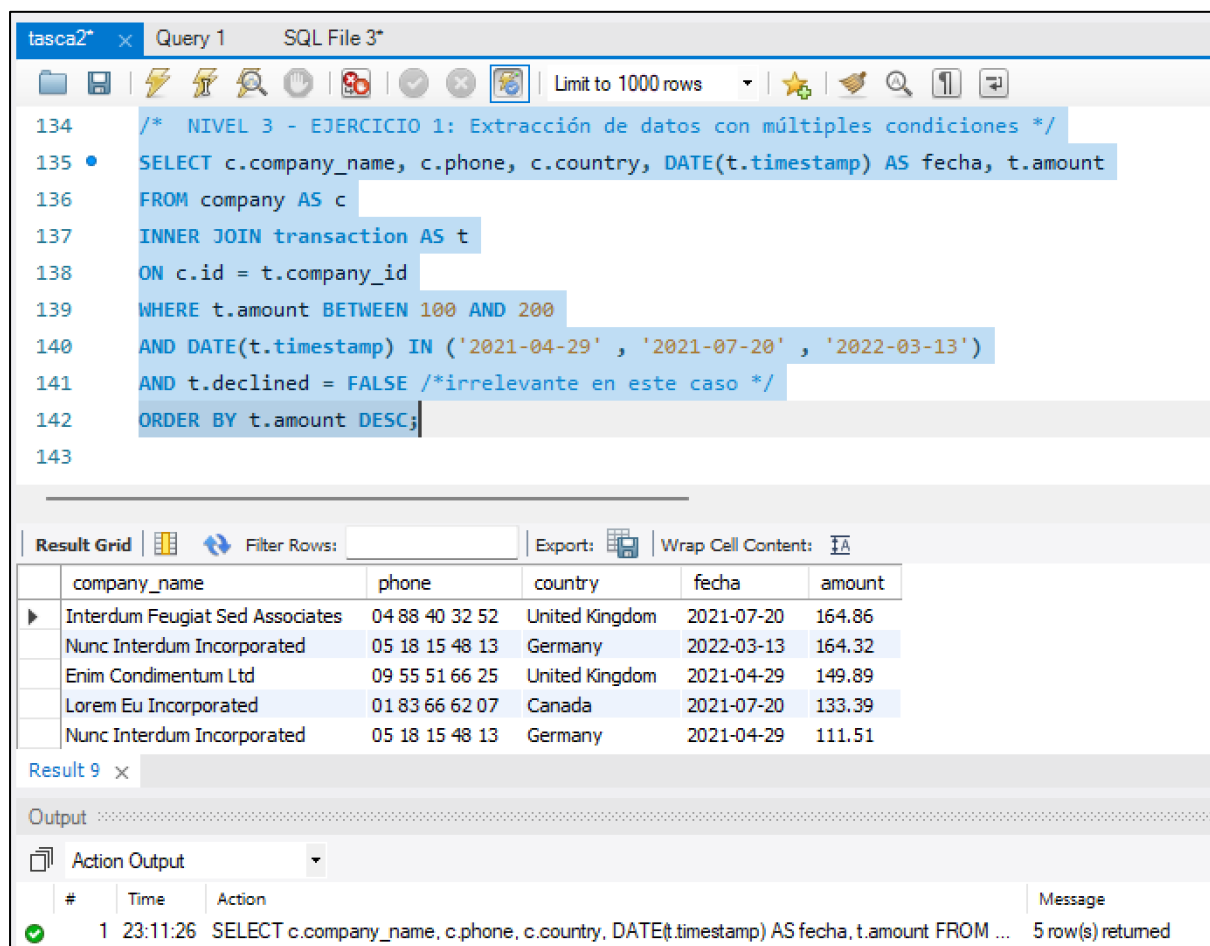
The output section shows the message: "100 row(s) returned".

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.

- Primero definimos todos los datos que queremos extraer de cada tabla, obteniendo la fecha con la función DATE() sobre la variable 'timestamp' de la tabla 'transaction'.
- Luego, después de definir el JOIN, ingresamos todas las condiciones solicitadas en la cláusula WHERE.
- Notamos que no podemos utilizar el alias 'fecha' en la cláusula WHERE, ya que, aunque el 'SELECT' se escribe antes del 'WHERE' en el código, se ejecutan de manera inversa, por lo que si el alias está definido en 'SELECT' al ejecutarse la cláusula 'WHERE' esta no tiene conocimiento del alias.



The screenshot shows a SQL IDE window with a query editor and a results grid. The query is as follows:

```
/* NIVEL 3 - EJERCICIO 1: Extracción de datos con múltiples condiciones */
SELECT c.company_name, c.phone, c.country, DATE(t.timestamp) AS fecha, t.amount
FROM company AS c
INNER JOIN transaction AS t
ON c.id = t.company_id
WHERE t.amount BETWEEN 100 AND 200
AND DATE(t.timestamp) IN ('2021-04-29', '2021-07-20', '2022-03-13')
AND t.declined = FALSE /*irrelevante en este caso */
ORDER BY t.amount DESC;
```

The results grid shows 5 rows of data:

company_name	phone	country	fecha	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

The output section shows the following message:

```
1 23:11:26 SELECT c.company_name, c.phone, c.country, DATE(t.timestamp) AS fecha, t.amount FROM ... 5 row(s) returned
```

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 especifiqueu si tenen més de 4 transaccions o menys.

- Para cumplir con lo solicitado, debemos crear una nueva variable, que nos diga en cuál de las dos categorías se encuentra la empresa listada.
- Aunque el caso de empresas con 4 transacciones no queda claro en la solicitud, en este caso no afecta los resultados.
- Unimos las tablas 'company' y 'transaction' con un JOIN, y le pedimos el nombre de la compañía, la cantidad de transacciones (haciendo uso de la función COUNT()), y finalmente la categoría en la que clasifica.
- La categoría la creamos usando CASE en la cláusula SELECT, para crear columnas calculadas, que dependen del conteo de transacciones.

The screenshot shows a SQL IDE window titled 'tasca2*' with a query named 'Query 1' in an 'SQL File 3*'.

```
143
144  /* NIVEL 3 - EJERCICIO 2: Conteo transacciones por empresa y creación de nueva variable calculada
145  SELECT c.company_name, COUNT(*) AS cantidad_transacciones,
146  CASE
147      WHEN COUNT(*) >= 4 THEN 'Más de 4 transacciones'
148      WHEN COUNT(*) < 4 THEN 'Menos de 4 transacciones'
149  END AS Categoria_Cliente
150  FROM transaction AS t
151  INNER JOIN company AS c
152  ON c.id = t.company_id
153  GROUP BY c.company_name
154  ORDER BY cantidad_transacciones DESC;
```

The 'Result Grid' shows the following data:

company_name	cantidad_transacciones	Categoria_Cliente
Nunc Interdum Incorporated	105	Más de 4 transacciones
Ut Semper Foundation	59	Más de 4 transacciones
Enim Condimentum Ltd	57	Más de 4 transacciones
Arcu LLP	56	Más de 4 transacciones
Lorem Eu Incorporated	54	Más de 4 transacciones
Malesuada PC	52	Más de 4 transacciones
Non Institute	30	Más de 4 transacciones

The 'Output' section shows 'Action Output' with a message: '1 23:27:38 SELECT c.company_name, COUNT(*) AS cantidad_transacciones, CASE WHEN COUNT(*)... 100 row(s) returned'.

- Finalmente, si queremos solo incluir las transacciones no declinadas, agregamos esta condición en la cláusula WHERE, obteniendo el siguiente resultado:

The screenshot shows a SQL IDE window titled 'tasca2*' with a query editor and a results pane. The query is as follows:

```

156  /* NIVEL 3 - EJERCICIO 2: Conteo transacciones por empresa y creación de nueva variable calculada para categorizar */
157  /*SIN TRANSACCIONES DECLINADAS*/
158  SELECT c.company_name, COUNT(*) AS cantidad_transacciones,
159  CASE
160      WHEN COUNT(*) >= 4 THEN 'Más de 4 transacciones'
161      WHEN COUNT(*) < 4 THEN 'Menos de 4 transacciones'
162  END AS Categoria_Cliente
163  FROM transaction AS t
164  INNER JOIN company AS c
165  ON c.id = t.company_id
166  WHERE t.declined = FALSE
167  GROUP BY c.company_name
168  ORDER BY cantidad_transacciones DESC;
  
```

The results pane shows a table with 3 columns: company_name, cantidad_transacciones, and Categoria_Cliente. The data is as follows:

company_name	cantidad_transacciones	Categoria_Cliente
Nunc Interdum Incorporated	104	Más de 4 transacciones
Ut Semper Foundation	58	Más de 4 transacciones
Enim Condimentum Ltd	56	Más de 4 transacciones
Arcu LLP	55	Más de 4 transacciones
Lorem Eu Incorporated	53	Más de 4 transacciones
Malesuada PC	51	Más de 4 transacciones

The bottom pane shows the Action Output with the following message:

```

#    Time    Action
1  23:34:29  SELECT c.company_name, COUNT(*) AS cantidad_transacciones, CASE WHEN COUNT(*)... 100 row(s) returned
  
```