

SPRINT 2

NIVEL 1

EJERCICIO 1

IMPORTAR ESTRUCTURA

Se crea la BBDD “transactions” e Importamos las tablas “company” y “transaction”

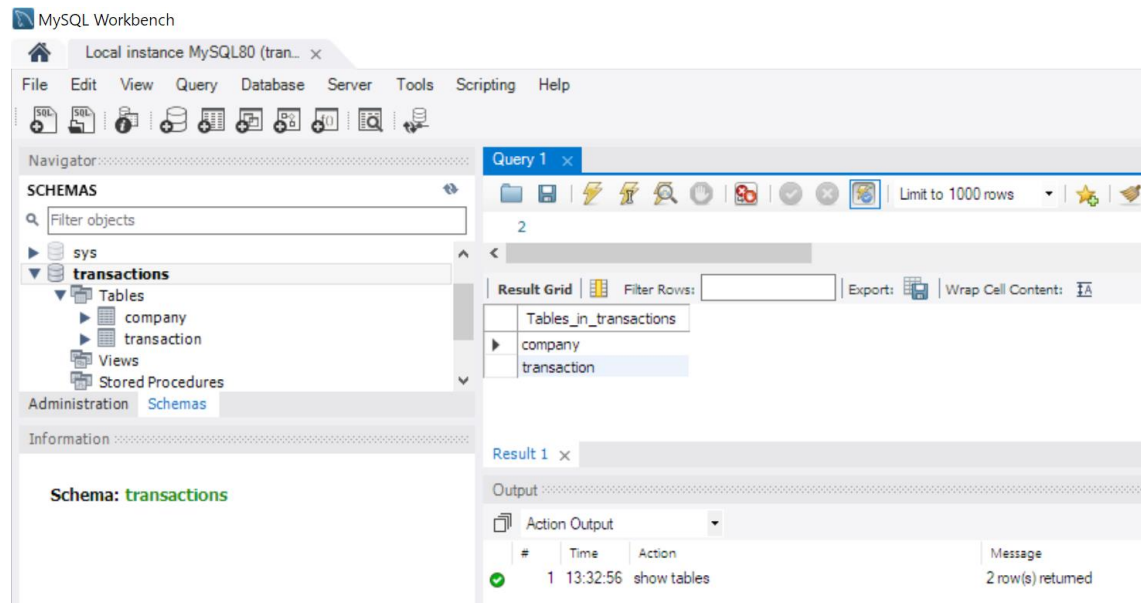
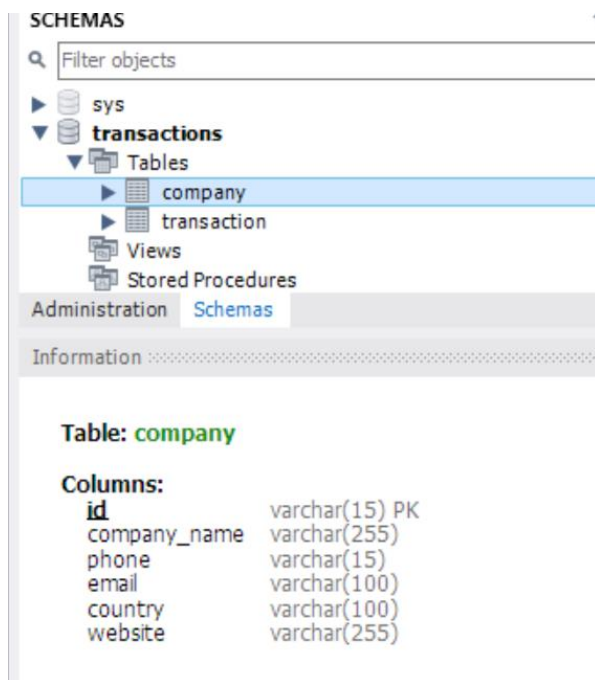


Tabla company:



La tabla company tiene los siguientes campos (columnas) de datos

- id : cadena de datos de longitud máxima 15 y valores únicos -> Primary key
- company_name: cadena de datos de longitud máxima de 255
- phone: cadena de datos de longitud máxima de 15
- email: cadena de datos de longitud máxima de 100
- country: cadena de datos de longitud máxima de 100
- website: cadena de datos de longitud máxima de 255

la tabla “transaction” :

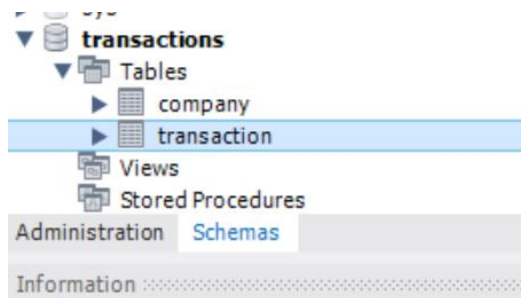


Table: transaction

Columns:

id	varchar(255) PK
credit_card_id	varchar(15)
company_id	varchar(20)
user_id	int
lat	float
longitude	float
timestamp	timestamp
amount	decimal(10,2)
declined	tinyint(1)

La tabla “transaction” tiene los siguientes campos (columnas) de datos:

- id : cadena de datos de longitud máxima 255 y valores únicos -> Primary key
- credit_card_id: cadena de datos de longitud máxima de 15
- company_id varchar: cadena de datos de longitud máxima de 20
- user_id : Número entero
- lat : Número decimal
- longitude: Número decimal
- timestamp: Fecha
- amount: número decimal de 10 dígitos, dos decimales de precision
- declined: variable booleana

IMPORTAR DATOS

La tabla company tiene 100 filas

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view with 'sakila', 'sys', and 'transactions' databases. Under 'transactions', the 'company' table is selected. The 'Information' pane shows the table's columns: id (varchar(15) PK), company_name (varchar(255)), phone (varchar(15)), email (varchar(100)), country (varchar(100)), and website (varchar(255)). The main query window shows a query: `select * FROM transactions.company`. The 'Result Grid' displays the first five rows of the table. The 'Output' pane shows the query execution details: 1 row returned, 100 row(s) returned.

id	company_name	phone	email	country	website
b-2222	Ac Fermentum Incorpor...	06 85 56 52 33	donec.porttitor.tellus@yahoo...	Germany	https://instagram.com/site
b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia	https://whatsapp.com/group/9
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.com/sub/cars
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.co.uk	Germany	https://cnn.com/user/110
b-2238	Ante Iaculis Nec Founda...	08 23 04 99 53	sed.dictum.oroin@outlook.ca	New Zealand	https://netflix.com/settings

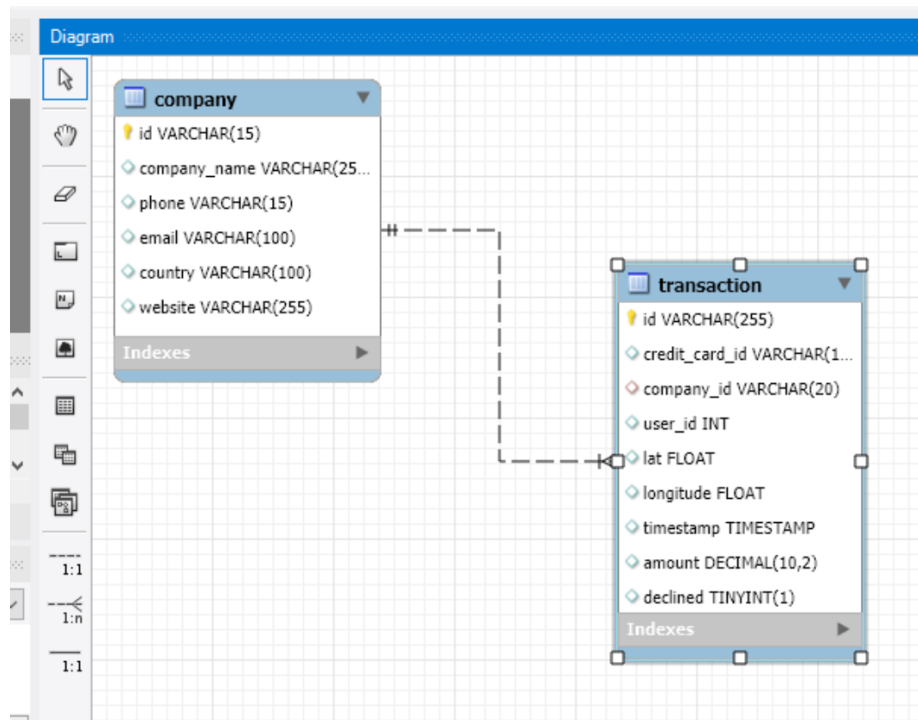
La tabla transaction tiene 587 filas

The screenshot shows the same database management tool interface. The 'transaction' table is selected in the 'SCHEMAS' pane. The 'Information' pane shows the table's columns: id (varchar(15) PK), company_id (varchar(15)), user_id (varchar(15)), lat (varchar(100)), longitude (varchar(100)), timestamp (varchar(255)), and amount (varchar(255)). The main query window shows a query: `select * FROM transactions.transaction`. The 'Result Grid' displays the first five rows of the table. The 'Output' pane shows the query execution details: 1 row returned, 587 row(s) returned.

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	92	81.9185	-12.5276	2021-08-28 23:42:24	466.9
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:29:18	49.53
063FBA79-99EC-66FB-29F7-25726D1764A5	CcU-2987	b-2250	275	-81.2227	-129.05	2022-01-06 21:25:27	92.61
0668296C-CD89-A883-76BC-2E4C4F8C8AE	CcU-3743	b-2618	265	-34.3593	-100.556	2022-01-26 02:07:14	394.1

RELACIONES ENTRE TABLAS

El campo 'company_name' de la tabla 'company' se relaciona 1:N con el campo 'id' de la tabla 'transaction'.



EJERCICIO 2

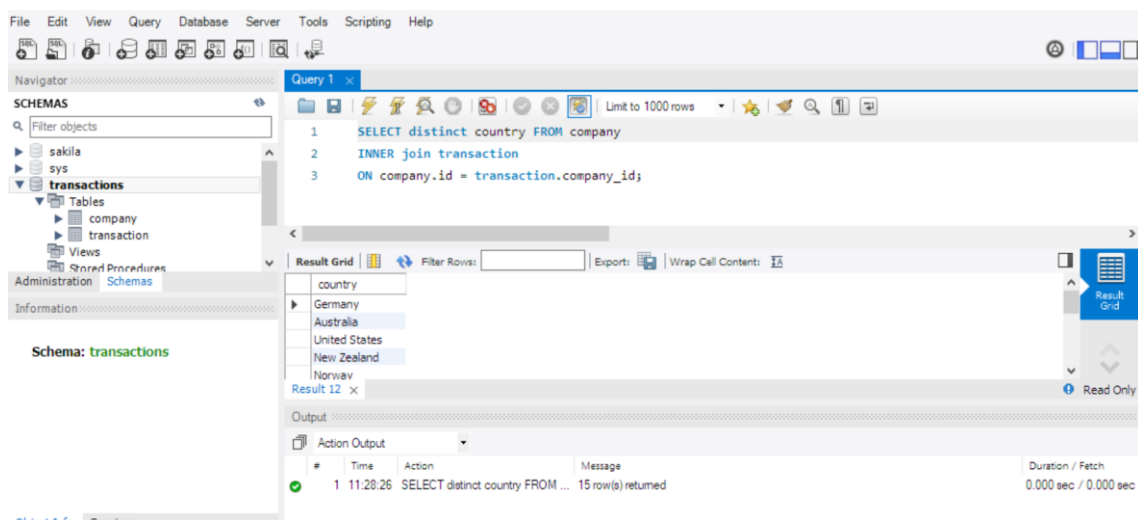
LISTADO DE PAISES QUE ESTÁN HACIENDO COMPRAS UTILIZANDO JOIN

Interpreto que nos piden todos los países que han hecho transacciones (estén 'declined' o no) ya que son los países que interactúan con la página.

La primary key es 'id' que identifica de modo único cada empresa con su país 'country' en la tabla 'company'.

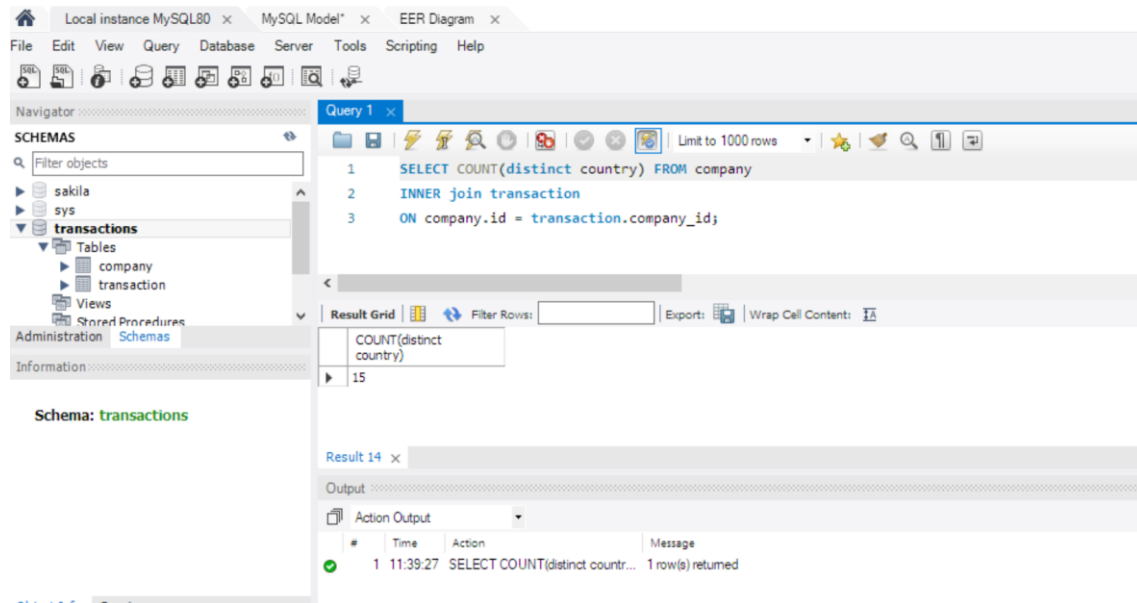
Como mantienen una relación 1:N tendremos que saber las compañías de la tabla 'transaction' que son comunes a la tabla 'country'. Por tanto aplicamos un INNER JOIN siendo 'transaction' Table 1 y 'company' Table 2. En la selección de columnas a mostrar aplicamos un filtro 'DISTINCT' al campo 'country'

Aun no sabiendo si todas las compañías de la tabla 'transaction' están incluidas en la tabla 'company' no serviría de nada utilizar un OUTER JOIN ya que solo sabríamos los países de la tabla 'company' y 'transaction' comunes ya que campo 'country' solo aparece en 'company'



DESDE CUANTOS PAISES SE REALIZAN LAS COMPRAS

Sería contabilizar el número de filas que tiene la query anterior.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'sakila', 'sys', and 'transactions' schemas. The 'transactions' schema is expanded, showing 'company' and 'transaction' tables. The main window displays 'Query 1' with the following SQL code:

```
1 SELECT COUNT(distinct country) FROM company
2 INNER join transaction
3 ON company.id = transaction.company_id;
```

The 'Result Grid' shows the query results:

	COUNT(distinct country)
15	

The 'Output' pane shows the execution log:

#	Time	Action	Message
1	11:39:27	SELECT COUNT(distinct countr...	1 row(s) returned

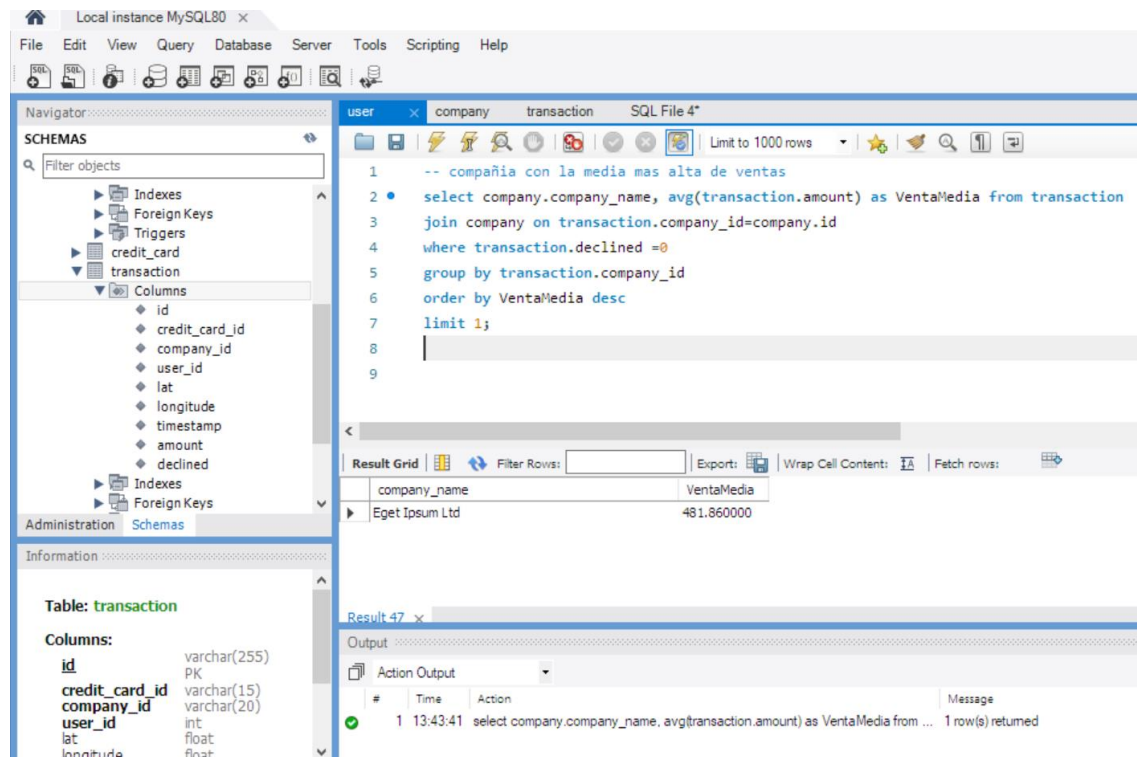
IDENTIFICAR LA COMPAÑÍA CON MEDIA MAS ALTA DE VENTAS

Considero que las ventas son las transacciones realizadas pero que no hayan sido rechazadas o 'declined'. Por tanto tendré que utilizar un filtro con 'declined'=0

Nos piden el nombre de la compañía que está en 'company' y la media de las ventas que es el campo de 'amount' pero filtrado con 'declined'=0 y que están en 'transacciones'

Por tanto, hacemos un join entre ambas `transaction.company_id = company.id`

Agrupamos por 'company-id' y ordenamos por la media de las ventas.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'transaction' table selected. The 'Columns' list for 'transaction' includes: id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, and declined. The main editor shows a SQL query:

```
1 -- compañía con la media mas alta de ventas
2 select company.company_name, avg(transaction.amount) as VentaMedia from transaction
3 join company on transaction.company_id=company.id
4 where transaction.declined =0
5 group by transaction.company_id
6 order by VentaMedia desc
7 limit 1;
```

The 'Result Grid' shows the following data:

company_name	VentaMedia
Eget Ipsum Ltd	481.860000

The 'Information' panel shows the structure of the 'transaction' table:

Table: transaction

Columns:

- id: varchar(255) PK
- credit_card_id: varchar(15)
- company_id: varchar(20)
- user_id: int
- lat: float
- longitude: float

The 'Output' panel shows the execution of the query:

```
1 13:43:41 select company.company_name, avg(transaction.amount) as VentaMedia from ... 1 row(s) returned
```


EJERCICIO 3

LISTADO DE TRANSACCIONES REALIZADAS POR ALEMANIA SIN UTILIZAR JOIN

Realizamos una ‘inner subquery’ para que nos identifique las Primary key (id) que mantienen la condición de que ‘country’ sea igual a “germany”.

Cómo ‘outer subquery’ mediante el operador “in” seleccionamos todos los campos en ‘transaction’ donde la Foreign key cumpla el condicionante anterior.

The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'company' and 'transaction'. The 'company' table structure is detailed in the 'Information' pane below it.

The main area displays a SQL query in the 'SQL File 4*' editor:

```
1 select * from transaction
2 where company_id in
3   (select id
4    from company
5    where country = "germany")
```

Below the query editor, the 'Result Grid' shows the results of the query. The grid has columns: id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, and declined. The results list transactions for companies in Germany.

At the bottom, the 'Output' pane shows the execution log with two entries:

- 2 19:24:15 SELECT company_name from company, transaction where transaction.company_id = company.id AND amount > (select avg(amount) from tran... 70 row(s) returned
- 3 19:25:06 select * from transaction where company_id in (select id from company where country = "germany") LIMIT 0, 1000 118 row(s) returned

LISTA EMPRESAS CON TRANSACCIONES SUPERIORES A LA MEDIA DEL TOTAL SIN UTILIZAR JOIN

Creo la primera subquery para obtener la media de todas las transacciones

```
select avg(amount) from transaction)
```

La siguiente query sera seleccionar los distintos company_id donde el amount sea mayor que el valor obtenido antes

Estos company_id en transaction corresponden a los id en company mediante la condicion where ... in

The screenshot shows the MySQL Workbench interface for a local instance of MySQL 8.0. The 'Schemas' pane on the left shows the 'transaction' table with columns: id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, and declined. The 'Information' pane shows the structure of the 'transaction' table: id (varchar(255), PK), credit_card_id (varchar(15)), company_id (varchar(20)), user_id (int), lat (float), and longitude (float).

The SQL editor contains the following query:

```
1 • select company_name from company
2   where id in (
3     select distinct company_id
4     from transaction
5     where amount >
6         (select avg(amount)
7          from transaction)
8   );
```

The 'Result Grid' shows the following data:

company_name
Ac Fermentum Incorporated
Magna A Neque Industries
Fusce Corp.
Ante Iaculis Nec Foundation
Donec Ltd

The 'Output' pane shows the execution of the query, indicating that 70 row(s) were returned.

EMPRESAS QUE NO TIENEN TRANSACCIONES SIN USAR JOIN

la primera subquery que realizo es identificar los valores únicos de 'company_id' en la tabla transaction.

La segunda subquery será ver en la tabla 'company' cuales de estos valores de 'company_id' no corresponden a los 'id' de tabla 'company'

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'sakila' and 'sys' databases. The 'sakila' database is expanded, showing 'Tables' and 'transaction'. The 'transaction' table is selected, and its columns are listed: id, company_name, phone, email, country, website. The 'Information' pane at the bottom left shows the structure of the 'company' table, including columns like id, company_name, phone, email, country, website, and their data types and constraints.

The main query editor shows the following SQL query:

```
1 select * from company
2 where id not in
3 (select distinct company_id
4  from transaction);
5
```

The 'Result Grid' shows the query results. The columns are id, company_name, phone, email, country, website. The results are empty, indicating that no companies were found that do not have transactions.

The 'Output' pane at the bottom right shows the execution log, indicating that the query was executed successfully and returned 0 rows.

El resultado es negativo y no hay empresas que no tengan transacciones.

NIVEL 2

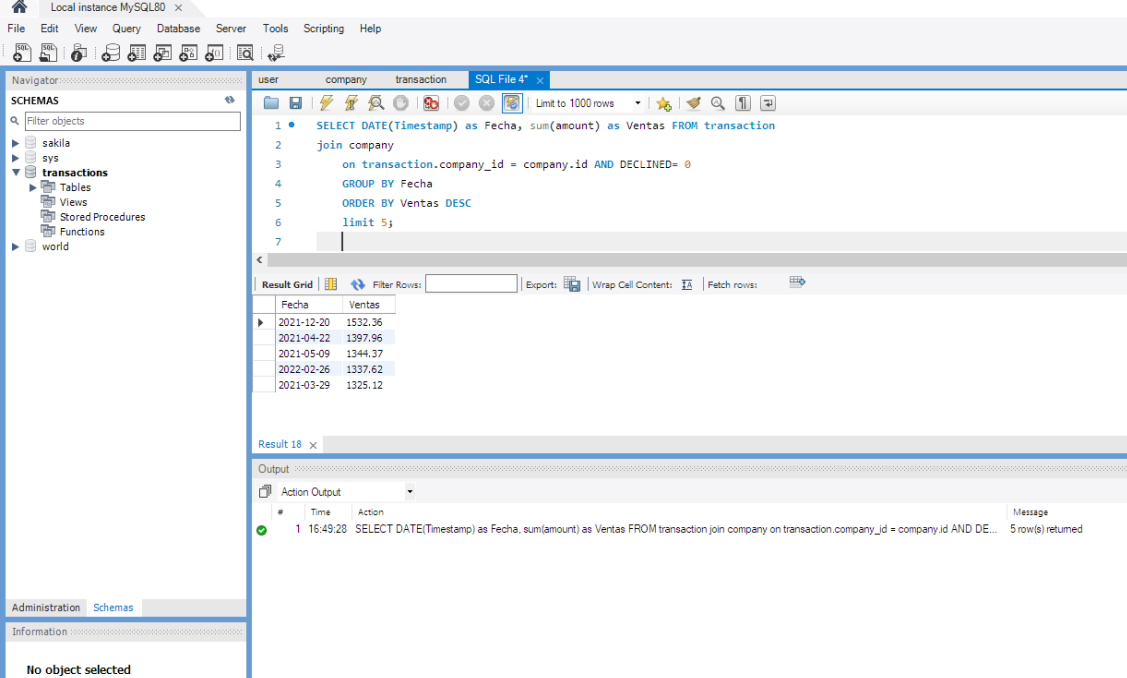
EJERCICIO 1

LOS CINCO DÍAS QUE MAYOR NÚMERO DE INGRESOS SE REGISTRARON

USANDO JOIN

Seleccionamos los campos 'Timestamp' le aplicamos el operador DATE y el operador suma con amount de la tabla 'Transaction'

Aplicamos join con las key y añadimos la condicion DECLINED=0 para obtener las ventas y no incluir las transacciones que no se hayan realizado. Cojemos las 5 primeras filas.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'sakila' selected. The main window shows a SQL query in the 'SQL File 4*' editor:

```
1 SELECT DATE(Timestamp) as Fecha, sum(amount) as Ventas FROM transaction
2
3 join company
4   on transaction.company_id = company.id AND DECLINED= 0
5
6 GROUP BY Fecha
7 ORDER BY Ventas DESC
8
9 limit 5;
```

Below the query editor, the 'Result Grid' displays the results of the query:

Fecha	Ventas
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

At the bottom, the 'Output' tab shows the execution message: '1 16:49:28 SELECT DATE(Timestamp) as Fecha, sum(amount) as Ventas FROM transaction join company on transaction.company_id = company.id AND DE... 5 row(s) returned'.

SIN USAR JOIN

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

sakila

sys

transactions

- Tables
 - company
 - credit_card
 - transaction
 - user
- Views
- Stored Procedures
- Functions

world

Administration Schemas

Information

No object selected

user company transaction SQL File 4*

Limit to 1000 rows

1 SELECT date(timestamp) as Fecha, sum(amount) as Ingresos FROM transaction

2 WHERE declined= 0

3 GROUP BY Fecha

4 ORDER BY Ingresos DESC

5 LIMIT 5;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

	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 6

Output

Action Output

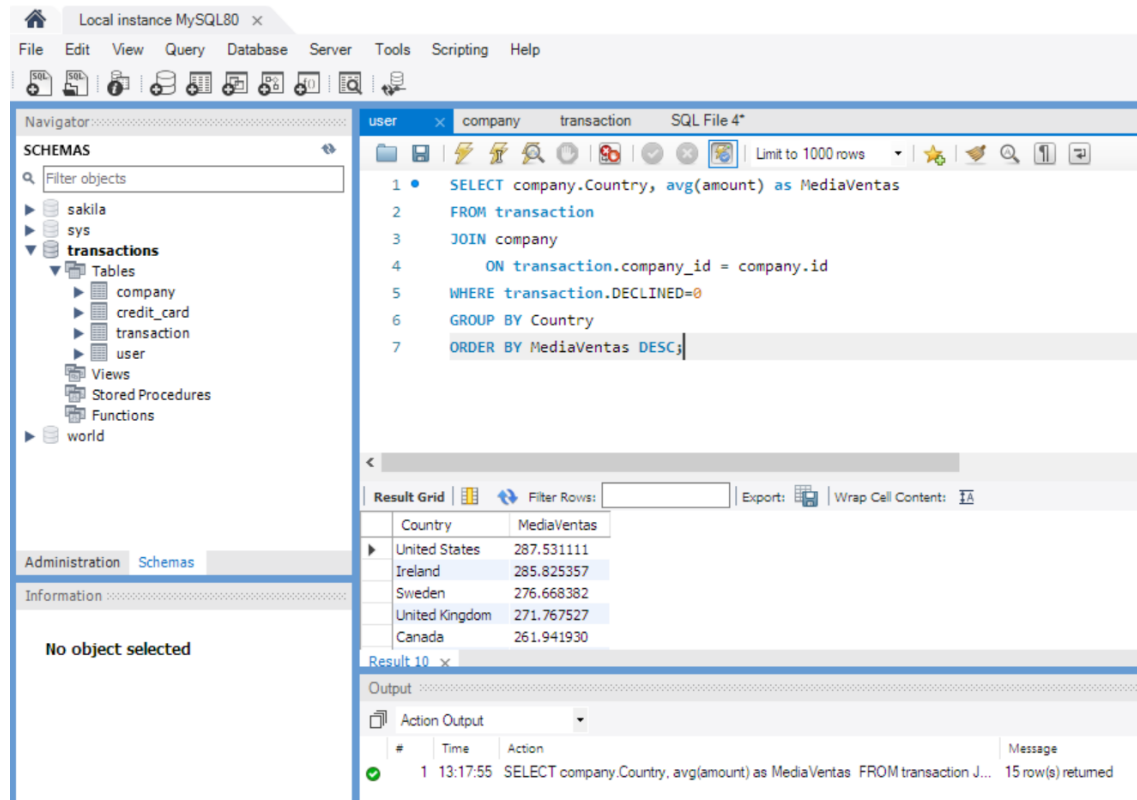
#	Time	Action	Message
✓ 1	13:11:09	SELECT date(timestamp) as Fecha, sum(amount) as Ingresos FROM transacti...	5 row(s) returned

EJERCICIO 2

RANKING POR PAIS DEL PROMEDIO DE LAS VENTAS

USANDO JOIN

Aplicamos la condición utilizando las Primary Keys ya que tenemos que usar ambas tablas, agrupamos por país y ordenamos por la media de las ventas



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'sakila' database selected. The main editor window shows a SQL query in 'SQL File 4*' that uses a JOIN to calculate the average sales amount per country. The query is as follows:

```
1 SELECT company.Country, avg(amount) as MediaVentas
2 FROM transaction
3 JOIN company
4 ON transaction.company_id = company.id
5 WHERE transaction.DECLINED=0
6 GROUP BY Country
7 ORDER BY MediaVentas DESC;
```

Below the query editor, the 'Result Grid' displays the results of the query. The columns are 'Country' and 'MediaVentas'. The results are ordered by the average sales amount in descending order.

Country	MediaVentas
United States	287.531111
Ireland	285.825357
Sweden	276.668382
United Kingdom	271.767527
Canada	261.941930

The bottom of the interface shows the 'Output' tab with 'Action Output' selected, displaying a message: '1 13:17:55 SELECT company.Country, avg(amount) as MediaVentas FROM transaction J... 15 row(s) returned'.

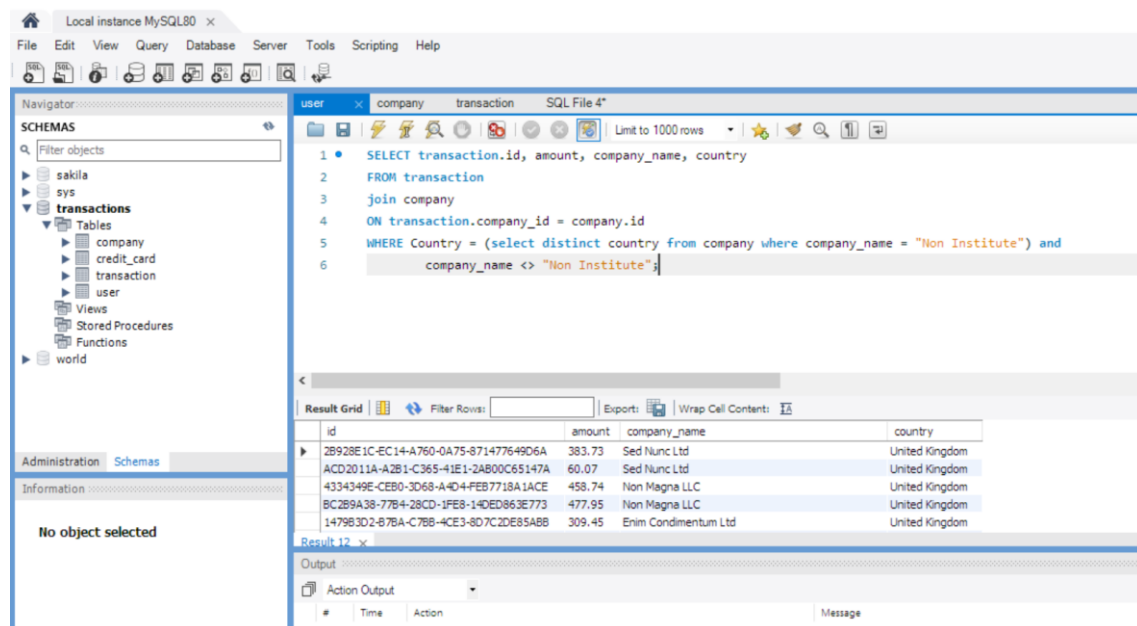
EJERCICIO 3

Listado de las transacciones realizadas en el mismo país que ‘Non Institute’

UTILIZANDO JOINS

Mediante la condicion where añadimos las tres condiciones:

- ‘company_id’ = ‘id’
- Country tiene que ser igual al ‘country’ cuando el nombre de la compañía es “Non Institute”
- El nombre de la compañía no es igual a “Non Institute”



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'sakila' selected. The main editor shows a SQL query in 'SQL File 4*':

```
1 SELECT transaction.id, amount, company_name, country
2 FROM transaction
3 join company
4 ON transaction.company_id = company.id
5 WHERE Country = (select distinct country from company where company_name = "Non Institute") and
6 company_name <> "Non Institute";
```

Below the query, the 'Result Grid' displays the results of the query. The table has four columns: id, amount, company_name, and country. The results are as follows:

id	amount	company_name	country
2B928E1C-EC14-A760-0A75-871477649D6A	383.73	Sed Nunc Ltd	United Kingdom
ACD2011A-A2B1-C365-41E1-2A800C65147A	60.07	Sed Nunc Ltd	United Kingdom
4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	458.74	Non Magna LLC	United Kingdom
8C2B9A38-77B4-28CD-1FE8-14DED863E773	477.95	Non Magna LLC	United Kingdom
1479B3D2-B7BA-C7BB-4CE3-8D7C2DE85ABB	309.45	Enim Conditum Ltd	United Kingdom

The bottom of the interface shows the 'Output' tab with 'Action Output' selected, displaying a table with columns for #, Time, Action, and Message.

UTILIZANDO QUERIES

Realizo la primera subquery para obtener el pais al que pertenece la empresa
“Non Institute”

Despues con la segunda query busco la id de las empresas que tienen ese pais
excluyendo a “Non Institute”

Con la outer query obtengo la tabla transaction donde la company_id es igual a la
id de antes seleccionada

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
1 select * from transaction
2 where company_id in
3 (select id from company
4 where country=
5 (select distinct country
6 from company
7 where company_name = "Non Institute") and 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-1FEB-14DED863E773	CcU-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95	1
1479B3D2-87BA-C7BB-ACE3-8D7C2DE85ABB	CcU-2994	b-2326	133	66.2672	172.399	2021-08-09 00:58:07	309.45	0

The Output tab shows the following message:

```
1 13:45:27 select * from transaction where company_id in (select id from company where c... 70 row(s) returned
```


NIVEL 3

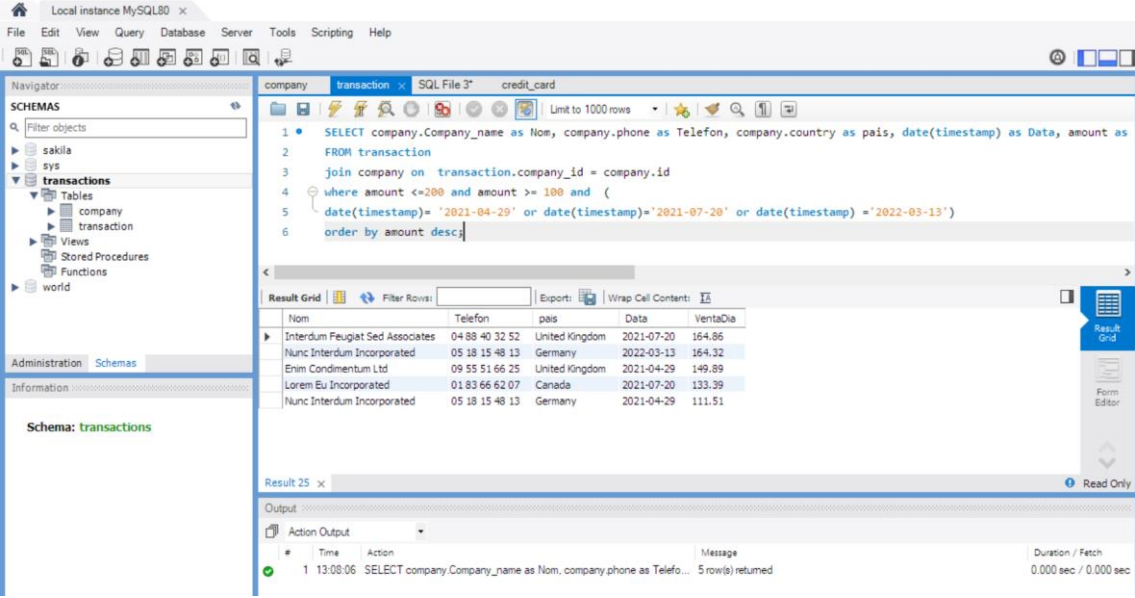
EXERCICI 1

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 los campos pedidos en el enunciado y como algunos pertenecen a la tabla de datos 'company' realizamos una join con ella mediante la keys que hemos definido y que las relacionan.

Para obtener las transacciones pedidas primero realizamos un filtro con 'where' donde introducimos mediante 'and' que tengan valor entre 200 y 100 y que estén dentro de una fecha determinada. Esta última condicion es un 'or' ya que son validas varias fechas a la vez.

Finalmente ordenamos descendente por el campo 'amount' para ordenar la vista por orden descendente.



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
1 SELECT company.Company_name as Nom, company.phone as Telefon, company.country as pais, date(timestamp) as Data, amount as  
2 FROM transaction  
3 join company on transaction.company_id = company.id  
4 where amount <=200 and amount >= 100 and (  
5 date(timestamp)= '2021-04-29' or date(timestamp)='2021-07-20' or date(timestamp) = '2022-03-13')  
6 order by amount desc;
```

The Results window displays the following data:

Nom	Telefon	pais	Data	VentaDia
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

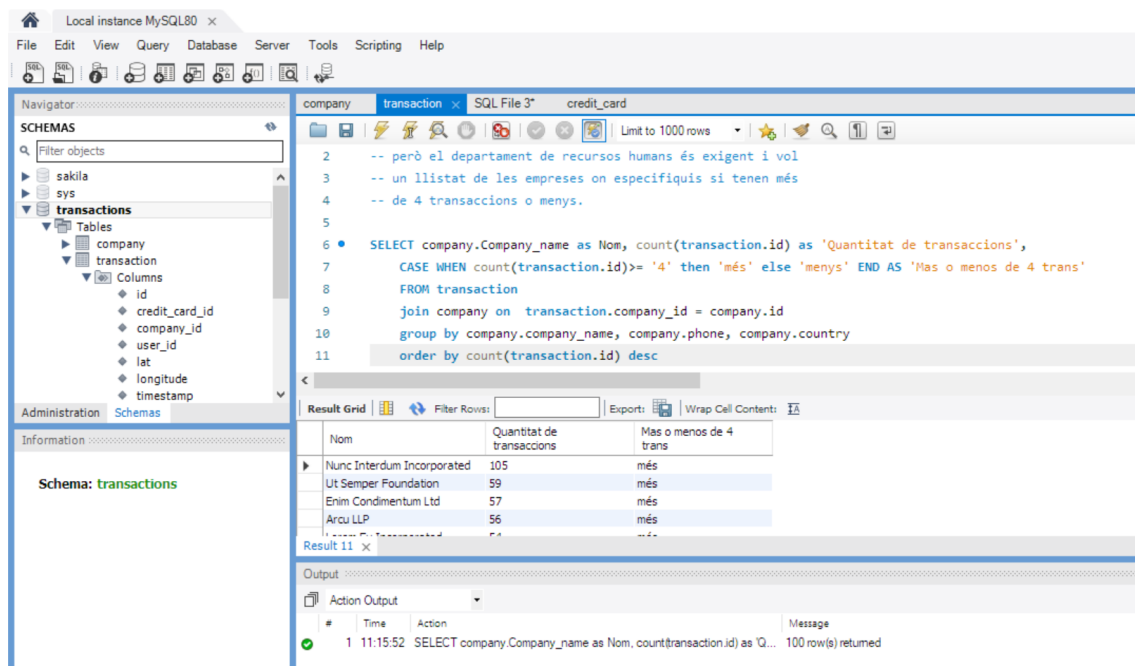
The Output window shows the execution message: "1 13:08:06 SELECT company.Company_name as Nom, company.phone as Telefono... 5 row(s) returned".

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 especifiquis si tenen més de 4 transaccions o menys.

Para realizar esta vista usaremos los campos nombre de empresa, numero de transacciones y habrá que agregar una columna que especifique para cada una si tiene mas o menos de 4 transacciones.

Para añadir una columna de informacion a la vista creada con la información de si hay mas de 4 o menos de 4 añadiremos en el select con la función 'case' la condicion y la respuesta que debe dar según cumpla la condición o no.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'sakila' and 'sys' databases. Under 'sakila', there is a 'transactions' schema containing 'company' and 'transaction' tables. The 'company' table has columns: id, credit_card_id, company_id, user_id, lat, longitude, and timestamp. The 'transaction' table has columns: id, company_id, and amount. The main window shows a SQL query in the 'SQL File 3*' editor. The query is as follows:

```
2 -- però el departament de recursos humans és exigent i vol
3 -- un llistat de les empreses on especifiquis si tenen més
4 -- de 4 transaccions o menys.
5
6 SELECT company.Company_name as Nom, count(transaction.id) as 'Quantitat de transaccions',
7       CASE WHEN count(transaction.id)>= '4' then 'més' else 'menys' END AS 'Mas o menos de 4 trans'
8 FROM transaction
9 join company on transaction.company_id = company.id
10 group by company.company_name, company.phone, company.country
11 order by count(transaction.id) desc
```

The 'Result Grid' shows the following data:

Nom	Quantitat de transaccions	Mas o menos de 4 trans
Nunc Interdum Incorporated	105	més
Ut Semper Foundation	59	més
Enim Condimentum Ltd	57	més
Arcu LLP	56	més

The 'Output' pane at the bottom shows the execution of the query, indicating that 100 rows were returned.