

# SPRINT 2

## NIVEL 1

### EJERCICIO 1

#### IMPORTAR ESTRUCTURA

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

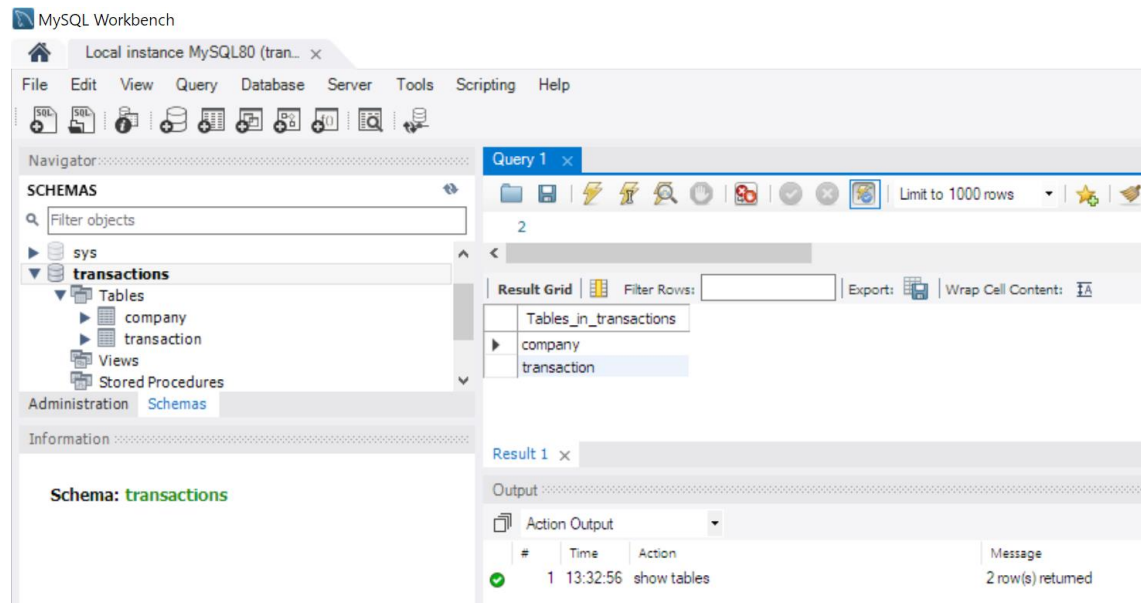
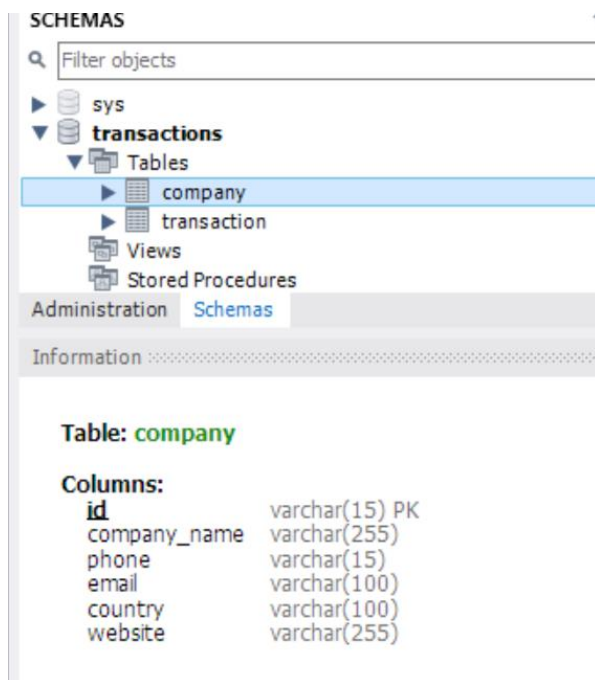


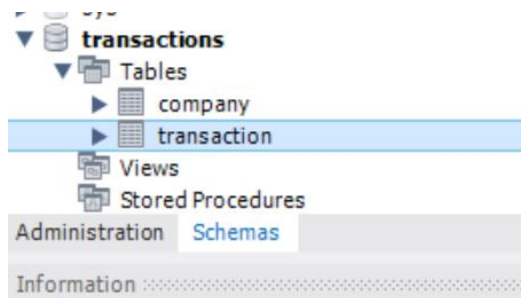
Tabla company:



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

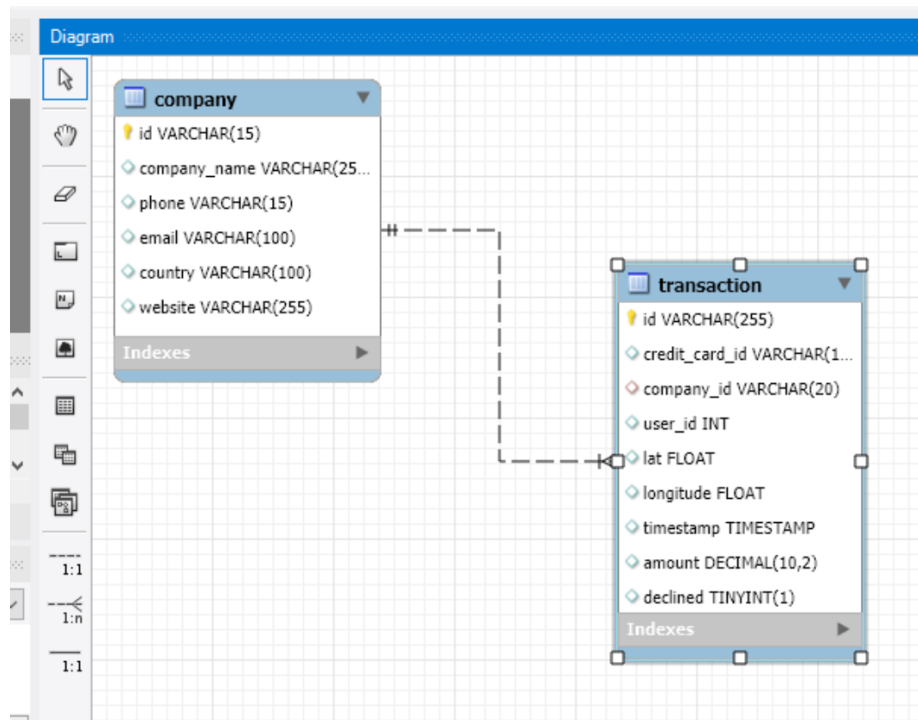
La tabla transaction tiene 587 filas

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)), credit\_card\_id (varchar(100)), 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: 587 row(s) returned, 0.000 sec.

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

Interpreto que nos piden todos los países que han hecho transacciones (estén 'declined' o no)

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 'company'. 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'

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view with 'sakila', 'sys', and 'transactions' schemas. Under 'transactions', the 'company' and 'transaction' tables are listed. The main query editor shows the following SQL query:

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

Below the query editor, the 'Result Grid' displays the results of the query. The first column is 'country', and the results are:

country
Germany
Australia
United States
New Zealand
Norway

At the bottom, the 'Output' pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	11:28:26	SELECT distinct country FROM ...	15 row(s) returned	0.000 sec / 0.000 sec

## DESDE CUANTOS PAISES SE REALIZAN LAS COMPRAS

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

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left displays the 'transactions' schema, which contains tables 'company' and 'transaction'. The 'Query 1' editor shows the following SQL query:

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

The 'Result Grid' pane 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

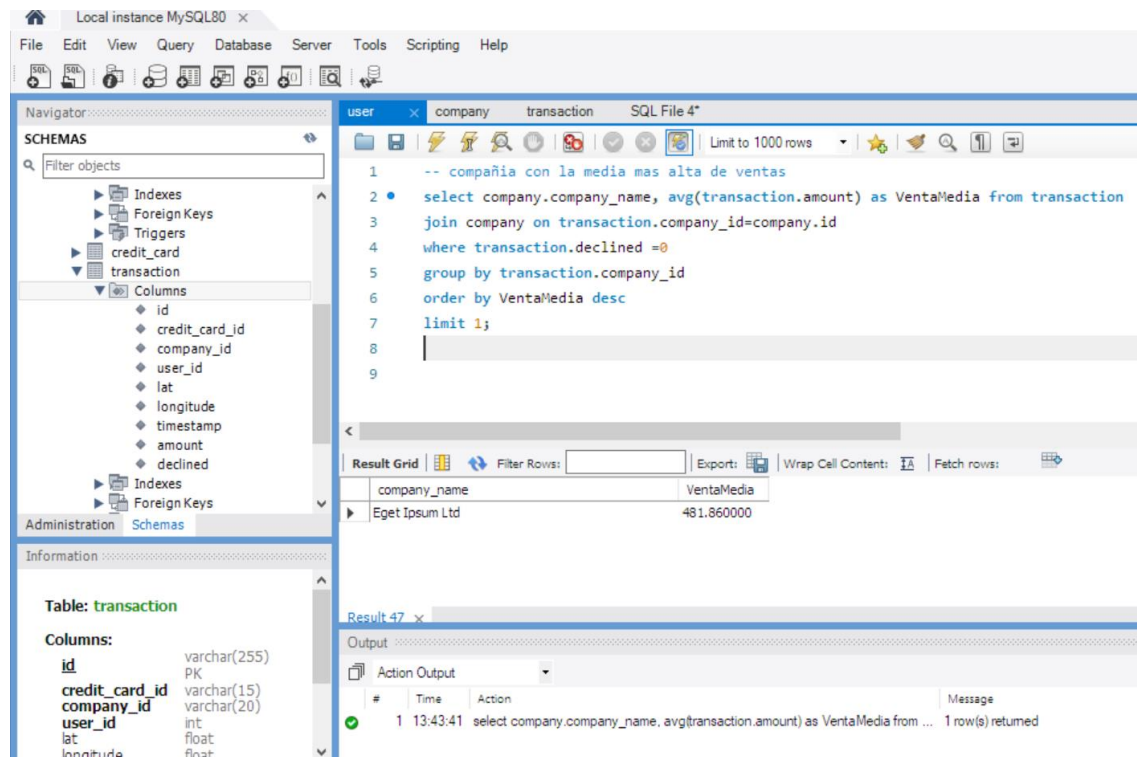
## IDENTIFICAR LA COMPAÑÍA CON MEDIA MAS ALTA DE VENTAS

Considero que las ventas son las transacciones realizadas pero que no hayan sido rechazadas o 'declined'.

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 under the 'Columns' section. The main window shows a SQL query in the 'SQL File 4\*' editor:

```
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;
8
9
```

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

company_name	VentaMedia
Eget Ipsum Ltd	481.860000

The bottom section of the interface shows the 'Information' tab for the 'transaction' table, listing its columns and data types:

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

The 'Output' tab at the bottom shows the execution of the query, indicating that 1 row(s) were 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, the '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 are filtered to show transactions for companies in Germany.

The 'Output' pane at the bottom shows the execution log, indicating that the query was executed successfully and returned 118 rows.

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
10861D1D-5B23-A76C-55EF-C568E49A05DD	CoU-2938	b-2222	275	83.7839	-178.86	2021-07-07 17:43:16	293.57	0
EA2C3281-C9C1-A387-44F8-729FB4B51C76	CoU-2938	b-2222	275	20.2004	-116.84	2021-05-09 10:25:08	119.36	1
0002E608-C9E0-1B37-4999-B99F43AD735A	CoU-2959	b-2234	275	9.68811	130.282	2021-04-15 05:30:17	252.47	1
AB069F53-965E-A2A8-CE06-CABC4FD92501	CoU-2959	b-2234	275	1.64819	-158.007	2021-04-15 13:37:18	60.99	0
0466A42E-47CF-8D24-FD01-C0B689713128	CoU-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:29:18	49.53	0
0A476ED9-0C13-1962-F878-D3563924B539	CoU-4359	b-2302	221	-56.4901	114.801	2022-02-26 20:33:54	430.49	0
122DC333-E19F-D629-DCD8-9C54CF1EB89A	CoU-4366	b-2302	221	29.6372	-166.173	2021-06-09 06:04:14	172.01	0
1352678A-2E7D-957C-C42C-6450A2B3ED54	CoU-4520	b-2302	210	20.6724	14.9732	2021-12-29 20:38:23	17.97	0
14CAE585-8FB1-3E4A-4C85-0EA4167534F4	CoU-4849	b-2302	189	-53.6202	93.0533	2021-12-31 00:29:42	388.04	0
158A3ACB-541C-0BCC-65BD-6373CC678F1C	CoU-4849	b-2302	183	42.5424	-170.347	2022-03-08 05:02:19	240.29	0
162C7E78-2B68-7971-A1E4-D212-4E732451	CoU-4527	b-2302	210	-69.1381	58.0017	2021-04-11 05:59:18	231.26	0
1717FD6B-ADAD-7082-A748-9112B892CCC	CoU-4219	b-2302	172	69.4892	-138.411	2021-12-29 16:18:54	249.91	0
1753A288-9FC1-52E6-5C39-A1FFB978003A	CoU-4345	b-2302	222	57.9422	-114.729	2021-08-17 05:32:08	497.84	0
186F530E-DE27-B1FE-882F-15B61CEB7726	CoU-4310	b-2302	225	-72.7448	36.6211	2021-12-20 13:13:45	238.16	0
18C4E2D0-1E4C-F35E-2198-C660B081DC25	CoU-4849	b-2302	177	68.0133	91.4839	2021-09-24 18:55:25	237.04	0
18CCBA7C-ABC1-813D-FAP3-4EAB97429368	CoU-4219	b-2302	173	51.3881	-156.371	2021-06-21 03:21:34	58.16	0
19E1EC3E-2119-1EFD-8AAE-5930D4A4E63F	CoU-4219	b-2302	154	-56.0839	116.987	2021-07-06 21:40:15	29.63	0
1B117D49-936C-8A6C-E94E-30C3293AA239	CoU-4219	b-2302	163	-75.099	109.034	2021-05-09 23:56:04	371.35	0
1B521826-5860-5A86-5364-6EB6A5CC21B7	CoU-4226	b-2302	231	-53.4613	49.1484	2021-08-14 02:39:50	476.33	0
1C3B3CC4-B919-7616-6A57-EEBDD7B76868	CoU-4415	b-2302	217	81.1737	62.8835	2021-09-27 16:07:34	462.35	0
2075B12D-69AB-7022-57BB-33ED49872766	CoU-4247	b-2302	229	-9.03682	88.9005	2022-02-13 04:07:29	87.44	0

## 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.

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 results:

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. Under 'sakila', the 'transactions' folder is expanded, showing tables 'company' and 'transaction'. The 'company' table structure is visible, including columns: id, company\_name, phone, email, country, website. The 'transaction' table structure is also visible, including columns: id, credit\_card\_id, company\_id, user\_id, lat, longitude, timestamp, amount, declined. 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 shows the execution details:

#	Time	Action	Message
1	19:52:19	select * from company where id not in (select distinct company_id from transaction) LIMIT 0, 1000	0 row(s) returned

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

# NIVEL 2

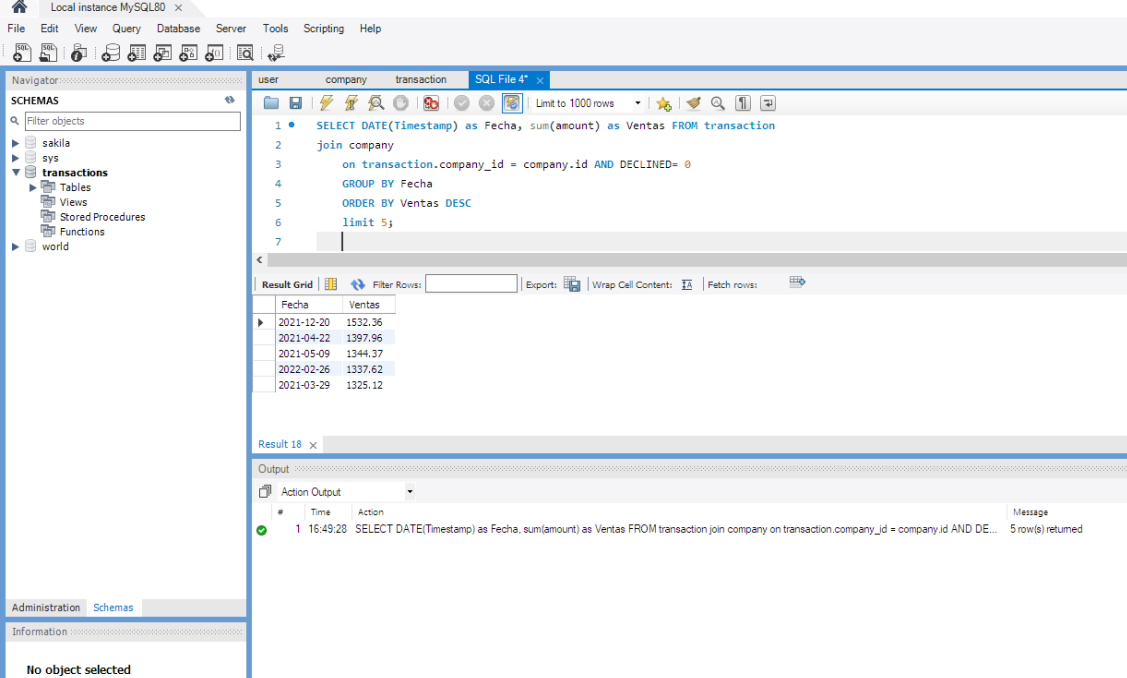
## EJERCICIO 1

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 editor shows a SQL query in 'SQL File 4' that joins the 'transaction' and 'company' tables, filters for 'DECLINED=0', groups by date, orders by total sales, and limits the results to 5 rows.

```
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, the 'Result Grid' shows the output 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

The 'Output' tab at the bottom 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

▶ Views

▶ Stored Procedures

▶ Functions

▶ world

user company transaction SQL File 4\*

1

2

3

4

5

6

7

SELECT DATE(Timestamp) as Fecha, sum(amount) as Ventas from transaction

where declined =0

group by Fecha

order by sum(amount) desc

limit 5;

Result Grid

Filter Rows:

Export:

Wrap Cell Contents:

Fetch rows:

	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

Result 16

Output

Action Output

#	Time	Action	Message
1	16:48:04	SELECT DATE(Timestamp) as Fecha, sum(amount) as Ventas from transaction where declined =0 group by Fecha order by sum(amount) desc limit 5	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 'sakila' and 'sys' databases. The 'transactions' table is selected under 'sakila'. The main window shows a SQL query in 'SQL File 4':

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

The 'Result Grid' shows the following data:

Country	MediaVentas
United States	287.531111
Ireland	285.825357
Sweden	276.668382
United Kingdom	271.767527
Canada	261.941930
Belgium	255.217500
Norway	251.114918
Italy	243.342222

The 'Output' pane shows the execution message: '1 17:09:14 SELECT company.Country, avg(amount) as MediaVentas FROM transaction JOIN company ON transaction.company\_id = company.id AND DECL... 15 row(s) returned'.

### UTILIZANDO SUBQUERIES

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'sakila' and 'sys' databases. The 'transactions' table is selected under 'sakila'. The main window shows a SQL query in 'SQL File 4' using a subquery:

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

The 'Result Grid' shows the following data:

Country	MediaVentas
United States	287.531111
Ireland	285.825357
Sweden	276.668382
United Kingdom	271.767527
Canada	261.941930
Belgium	255.217500
Norway	251.114918
Italy	243.342222
Germany	242.239189
Netherlands	240.940000
China	222.240000
Australia	177.331667
France	169.410000
New Zealand	167.061667
Spain	26.220000

The 'Output' pane shows the execution message: '1 18:27:17 SELECT company.Country, avg(amount) as MediaVentas FROM company.transaction WHERE transaction.company\_id = company.id AND DECL... 15 row(s) returned'.

## EJERCICIO 3

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

### SIN JOINS

Mediante la condición 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. On the left, the 'SCHEMAS' pane displays the database structure, including tables 'company' and 'transaction'. The 'company' table has columns: id, credit\_card\_id, company\_name, user\_id, lat, longitude, and timestamp. The 'transaction' table has columns: id, amount, company\_name, and country. The main query editor shows the following SQL query:

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

The 'Result Grid' shows the following data:

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
BC2B9A38-77B4-28CD-1FE8-14DED863E773	477.95	Non Magna LLC	United Kingdom
147983D2-87B4-C7B8-4CE3-8D7CDE85A8B	309.45	Enim Condimentum Ltd	United Kingdom
152598C2-029D-D684-4B66-91EDF393EBFF	395.43	Enim Condimentum Ltd	United Kingdom
1B636858-A2E8-7C69-D9C9-C5453SDAFD3B	195.06	Enim Condimentum Ltd	United Kingdom
20418DE5-B804-8E98-BD7A-A95C18FDBF5C	479.52	Enim Condimentum Ltd	United Kingdom
239B8576-6C0E-137A-C2F6-3180A188A2D3	43.90	Enim Condimentum Ltd	United Kingdom
267C4A86-7BA7-1C5E-0718-2824983C87DD	122.63	Enim Condimentum Ltd	United Kingdom
3142C93E-83B7-49E4-EE2D-29CA834B198D	91.59	Enim Condimentum Ltd	United Kingdom

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

# UTILIZANDO QUERIES

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

sakila

sys

transactions

- Tables
  - company
  - credit\_card
  - transaction
  - user
- Views
- Stored Procedures
- Functions

world

user company transaction SQL File 4\*

Limit to 1000 rows

1 SELECT transaction.id, amount, company\_name, country

2 FROM transaction

3 join company

4 ON transaction.company\_id = company.id and

5 Country = (select distinct country from company where company\_name = "Non Institute") and

6 company\_name <> "Non Institute";

7

8

9

Result Grid

Filter Rows:

Export:

Wrap Cell Content: 15

	id	amount	company_name	country
▶	2B928E1C-EC14-A760-0A75-871477649D6A	383.73	Sed Nunc Ltd	United Kingdom
	ACD2011A-A2B1-C365-41E1-2AB00C65147A	60.07	Sed Nunc Ltd	United Kingdom
	4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	458.74	Non Magna LLC	United Kingdom
	BC2B9A38-77B4-28CD-1FEB-14DED863E773	477.95	Non Magna LLC	United Kingdom
	147983D2-87BA-C7B8-4CE3-8D7C2DE85AB8	309.45	Enim Conditum Ltd	United Kingdom
	152598C2-029D-D684-4B66-91EDF393EBFF	395.43	Enim Conditum Ltd	United Kingdom
	1B636858-A2E8-7C69-D9C9-C5453SDAFD3B	195.06	Enim Conditum Ltd	United Kingdom
	20418DE5-B804-BE98-8D7A-A95C1BFD8F5C	479.52	Enim Conditum Ltd	United Kingdom
	239B8576-6C0E-137A-C2F6-3180A188A2D3	43.90	Enim Conditum Ltd	United Kingdom
	267C4A86-7BA7-1C5E-0718-2824983C87DD	122.63	Enim Conditum Ltd	United Kingdom
	3142C93E-B387-49E4-EE2D-29CA834B198D	91.59	Enim Conditum Ltd	United Kingdom

Result 46 x

Output

Action Output

#	Time	Action	Message
1	18:30:57	SELECT transaction.id, amount, company_name, country FROM transaction join company ON transaction.company_id = company.id and	Co... 70 row(s) returned

Administration Schemas

Information

No object selected