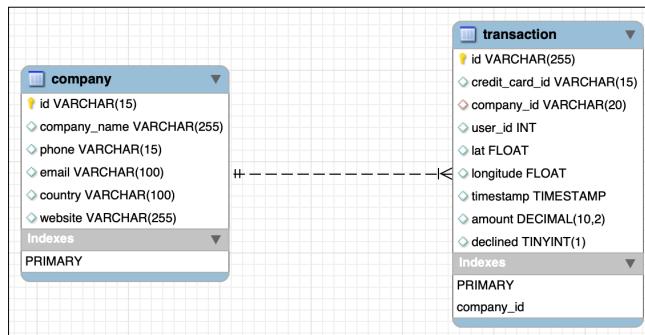


SPRINT 2

NIVELL 1

1.1.- A partir dels documents adjunts (estructura_dades i dades_introduir), importa les dues taules. Mostra les característiques principals de l'esquema creat i explica les diferents taules i variables que existeixen. Assegura't d'incloure un diagrama que il·lustri la relació entre les diferents taules i variables.



La BD está formada por dos tablas; **company** y **transaction**, donde se relacionan las transacciones que efectúan un conjunto de empresas y facilitan información tanto de las empresas como de las mismas transacciones.

La relación entre ambas es de 1:n en el sentido **company->transaction**, es decir, pueden existir varias transacciones realizadas por una única empresa.

La tabla **company** está formada por **100 registros** con **6 campos (o atributos)** de información cada uno de ellos acerca de las empresas candidatas a haber realizado transacciones.

3 • describe company;						
Result Grid Filter Rows: Search Export:						
Field	Type	Null	Key	Default	Extra	
id	varchar(15)	NO	PRI	NULL		
company_name	varchar(255)	YES		NULL		
phone	varchar(15)	YES		NULL		
email	varchar(100)	YES		NULL		
country	varchar(100)	YES		NULL		
website	varchar(255)	YES		NULL		

- **id**: varchar() con capacidad para 15 caracteres. Cada id lo forman dígitos y el carácter especial '-'. Actúa como PK (Primary Key). Es un código asociado a cada empresa.
- **company_name**: varchar() con capacidad para 255 caracteres. Cada company_name se compone básicamente por varias palabras (mayúsculas y minúsculas) y espacios entre ellas. Es el nombre de la empresa.
- **phone**: varchar() con capacidad para 15 caracteres. Están formados por distintas combinaciones numéricas separadas entre ellas por espacios. es el teléfono de cada una de las empresas.
- **email**: varchar() con capacidad para 100 caracteres. Compuestos por distintas combinaciones de letras y los caracteres especiales '.' y '@'. Es la dirección de correo electrónico de cada una de las empresas.
- **country**: varchar() con capacidad para 100 caracteres. Formados tan solo por letras. indica el país al que pertenece la empresa.
- **website**: varchar() con capacidad para 255 caracteres. Compuestos por distintas combinaciones de letras y los caracteres especiales '.', ':' y '/'. Indica la dirección web de cada empresa.

Por otra parte la tabla *transaction* la componen **597 registros** con **9 campos (o atributos)** que facilitan información acerca de las transacciones realizadas.

Result Grid:

Field	Type	Null	Key	Default	Extra
<code>id</code>	<code>varchar(255)</code>	NO	PRI	<code>NULL</code>	
<code>credit_card_id</code>	<code>varchar(15)</code>	YES	MUL	<code>NULL</code>	
<code>company_id</code>	<code>varchar(20)</code>	YES	MUL	<code>NULL</code>	
<code>user_id</code>	<code>int</code>	YES		<code>NULL</code>	
<code>lat</code>	<code>float</code>	YES		<code>NULL</code>	
<code>longitude</code>	<code>float</code>	YES		<code>NULL</code>	
<code>timestamp</code>	<code>timestamp</code>	YES		<code>NULL</code>	
<code>amount</code>	<code>decimal(10,2)</code>	YES		<code>NULL</code>	
<code>declined</code>	<code>tinyint(1)</code>	YES		<code>NULL</code>	

Action Output:

Time	Action	Response
11:43:34	describe transaction	9 row(s) returned

- **`id`**: varchar() con capacidad para 255 caracteres. Cada id lo forman dígitos, letras (mayúsculas) y el carácter especial '-'. Actúa como PK (Primary Key). Es un código asociado a cada transacción.
- **`credit_card_id`**: varchar() con capacidad para 15 caracteres. Cada credit_card_id se compone de letras (mayúsculas y minúsculas) y un guión '-'. Es un código asociado a cada una de la tarjetas de crédito.
- **`company_id`**: varchar() con capacidad para 20 caracteres. Están formados por letras (minúsculas) y un guión '-'. Actúa como FK (Foreignn Key) para relacionarse con la tabla company. Es un código asociado a cada empresa.
- **`user_id`**: int. Compuestos básicamente por la combinación de 3 o 4 dígitos que indican el usuario que realizó la transacción.
- **`lat`**: float. Valor numérico en coma flotante, que indica la latitud en la que se encuentra la empresa que realizó la venta.
- **`longitude`**: float. Valor numérico en coma flotante, que indica la longitud en la que se encuentra la empresa que realizó la venta.
- **`timestamp`**: timestamp. Variable del tipo DATA que actúa como indicador de fecha y hora y debe hacer referencia al momento en el que se realizó cada una de las transacciones.
- **`amount`**: decimal(10,2). Variable del tipo decimal (mayor precisión que el float) con capacidad para 10 dígitos de los que 2 serán decimales (detrás del punto). Hace referencia la importe de la venta.
- **`declined`**: tinyint(1). Variable del tipo 'integer pequeño', utilizado para rangos inferiores al de un int. En este caso el rango es de 1, e indica si la transacción se pudo realizar (1) o no (0).

Ambas se relacionan a través de la Primary Key (**PK**) de la tabla *company* y la Foreign Key (**FK**) de la tabla *transaction*.

La **PK** de la tabla *company* es el campo *id*, único en toda la tabla y que identifica inequívocamente a cada una de las empresas.

La **PK** de la tabla *transaction* también es el campo identificado como *id*, pero esta vez identifica cada una de las transacciones que contiene la tabla.

2.- Utilitzant JOIN realitzaràs les següents consultes:

2.1.- Llistat dels països que estan fent compres.

```

3   -- Llistat dels països que estan fent compres.
4 •   select distinct country as 'PAISES QUE REALIZAN COMPRAS'
5     from company
6     join transaction on company.id=transaction.company_id
7     order by country desc;
8

```

Result Grid Filter Rows: Search Export:

PAISES QUE REALIZAN COMPRAS
United States
United Kingdom
Sweden
Spain
Norway
New Zealand
Netherlands
Italy
Ireland
Germany
France
China
Canada
Belgium
Australia

Result 1

Action Output

Time	Action	Response
15:31:46	select distinct country as 'PAISES QUE REALIZAN COMPRAS' from company join transaction on company.id=transaction.company_id order by country desc	15 row(s) returned

El SELECT **DISTINCT** evita duplicar los datos a consultar.

La JOIN permite relacionar las tablas *company* y *transaction* a través de la **PK** de *company* y la **FK** de *transaction*, de esa forma es posible mostrar el nombre de las empresas.

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

```

9   -- 2.2. Des de quants països es realitzen les compres (utilitzant JOIN):
10 •  select count(distinct country) AS 'Nº PAISES QUE REALIZAN COMPRAS'
11    from company
12    join transaction on company.id=transaction.company_id;
13

```

Result Grid Filter Rows: Search Export:

Nº PAISES QUE REALIZAN COMPRAS
15

Result 31

Action Output

Time	Action	Response
11:30:37	select count(distinct country) AS 'Nº PAISES QUE REALIZAN COMPRAS' fr...	1 row(s) returned

con la **JOIN** tan solo se realiza la consulta sobre registros comunes en ambas tablas. Con la función de agregación **COUNT** en este caso, combinada con un **DISTINCT** (para no duplicar valores) obtenemos el resultado.

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

```

14 -- 2.3. Identifica la companyia amb la mitjana més gran de vendes (utilitzant JOIN):
15 • select company_name,round(avg(amount),2) as 'MEDIA VENTAS'
16   from transaction
17   join company on transaction.company_id=company.id
18   group by company_name
19   order by avg(amount) desc
20   limit 1;
21

```

Result Grid Filter Rows: Search Export:

company_name	MEDIA VENTAS
Eget Ipsum Ltd	473.08

Result 33

Action Output

	Time	Action	Response
<input checked="" type="checkbox"/>	1 11:36:53	select company_name,round(avg(amount),2) as 'MEDIA VENTAS' from transaction	1 row(s) returned

A través de la **JOIN** entre ambas tablas, de company consulto el nombre y de la tabla transaction la media de ventas.

Ordenando la consulta por las medias calculadas de mayor a menor (**DESC**) me permite seleccionar la primera (mediante la instrucción **LIMIT**) y devolver tan solo la empresa con media más alta.

3.- Utilitzant només subconsultes (sense utilitzar JOIN):

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

```

39   -- 3.1. Mostra totes les transaccions realitzades per empreses d'Alemania (sense utilitzar JOIN).
40 • select id                                     -- Sacar todos los id de empresas cuyo país es Alemania
41   from company
42   where country='Germany';
43
44 • select id
45   from transaction
46   where company_id in (select id             -- Filtrar incluyendo la query anterior.
47     from company
48     where country='Germany');
49

```

Result Grid Filter Rows: Search Edit: Export/Import:

id
108B1D1D-5B23-A76C-55EF-C568E49A05DD
EA2C3281-C9C1-A387-44F8-729FB4B51C76
0DD2E608-5C9E-D1B3-4999-B99F43AD735A
AB069F53-965E-A2A8-CE06-CA8C4FD92501
0466A42E-47CF-8D24-FD01-C0B689713128
0A476ED9-0C13-1962-F87B-D3563924B539

transaction 5

Action Output	Time	Action	Response
1	11:50:21	select id from transaction where company_id in (select id -- Filtrar incluyendo la query anterior....	118 row(s) returned

Con la **primera query** devuelvo todos las empresas que residen en Alemania. Esta misma consulta servirá de sub-query para la consulta definitiva.

En la **segunda query**, la búsqueda es sobre la tabla transaction aplicando como filtro la primera query y la combinación de la sentencia **WHERE** y el operador **IN**.

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

```

50    -- 3.2. Llista les empreses que han realitzat transaccions per un amount superior a la mitjana de totes les transaccions (sense utilitzar JOIN).
51 •  select round(avg(amount),2)          -- calculo de la media total de amount de todas las transacciones
52   from transaction;
53
54 •  select company_id
55   from transaction                      -- id de empresas con amount superior a la media total
56   where amount>(select round(avg(amount),2)
57       from transaction);
58
59 •  select company_name
60   from company
61   where id in (select company_id
62     from transaction
63     where amount>(select round(avg(amount),2)
64         from transaction));
65
66
100% 9:50
Result Grid Filter Rows: Search Export:
company_name
Ac Fermentum Incorporated
Magna A Neque Industries
Fusce Corp.
Ante laculis Nec Foundation
Donec Ltd
Sed Nunc Ltd
company 12
Action Output
Time Action Response
1 12:32:10 select company_name from company where id in (select company_id from transaction where amount>(select round(avg(amount),2) from transaction)) 70 row(s) returned

```

En la **primera query** (consulta) sacamos la media de todas las transacciones. Utilizo la función **round** para limitar los decimales de la media devuelta.

En la **segunda query**, devuelvo los id de las empresas residentes en Alemania, utilizando como parte del filtro la primera consulta realizada.

La tercera query (definitiva), devuelve el nombre asociado a los id obtenidos en la consulta anterior. La instrucción **WHERE** combinada con el operador **IN** permite realizar el filtro.

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

```

68 -- 3.3. Eliminaran del sistema les empreses que no tenen transaccions registrades, entrega el llistat d'aquestes empreses (sense utilitzar JOIN).
69 • select company_id,count(id)          -- Tabla resumen que devuelve el nº de transacciones realizadas por cada empresa
70   from transaction
71   group by company_id
72   having count(id)='NULL';
73
74 • select company_name
75   from company
76   where exists (select company_id,count(id)      -- Con exists filtramos por las coincidencias entre id
77     from transaction
78     group by company_id
79     having count(id)='NULL');
80
81
100% 4:68
Result Grid Filter Rows: Search Export:
company_name
company 19
Action Output
Time Action Response
1 12:52:59 select company_name from company where exists (select company_id,count(id) -- Con exists filtramos por las... 0 row(s) returned

```

En este caso todas las empresas han realizado mínimo una transacción.

En la **primera query** saco una tabla que indica el número de transacciones que ha realizado cada empresa. El filtro **having** nos discrimina tan solo aquellas empresas de la nueva tabla que no tienen ningún valor en el campo de las transacciones realizadas (**null**).

En la **segunda query** con el filtro **WHERE** combinado con **EXISTS** busco coincidencias entre la primera consulta y la segunda.

NIVELL 2

2.1.- 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.

```
-- 2.1.- 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.
select cast(timestamp as date) as FECHA, sum(amount) as INGRESOS
from transaction
group by FECHA
order by INGRESOS desc
limit 5;
```

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

Result 25

Action Output

Time	Action	Response
10:23:50	select cast(timestamp as date) as FECHA, sum(amount) as INGRESOS from transaction group by FECHA order by INGRESOS desc limit 5	5 row(s) returned

Convierto el tipo de dato que indica la fecha de la tabla *transaction* de **timestamp a data** para eliminar la parte horaria y poder agrupar por fecha tan solo. Para ello utilizo la función **CAST**.

Una vez agrupado por fecha lo ordeno de mayor a menor y devuelvo tan solo los 5 primeros registros.

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

```
-- Quina és la mitjana de vendes per país? Presenta els resultats ordenats de major a menor mitjà.
• select country,avg(amount) as 'MEDIA DE VENTAS'
  from company
  join transaction on company.id=transaction.company_id
  group by country
  order by 'MEDIA DE VENTAS';
```

country	MEDIA DE VENTAS
Germany	244.203220
Australia	232.052500
United States	309.179412
New Zealand	222.277273

Result 31

Action Output

Time	Action	Response
10:59:12	select country,avg(amount) as 'MEDIA DE VENTAS' from company join transaction on company.id=transaction.com... 15 row(s) returned	

Agrupo por el campo *country* de la tabla *company* y mediante la JOIN devuelvo la media de los ingresos mediante la función de agregación **avg**.

2.3.- 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.

2.3.1.- Mostra el llistat aplicant JOIN i subconsultes.

```

99  -- 2.3.1.- Mostra el llistat aplicant JOIN i subconsultes.
100 • select distinct country
101   from company
102   join transaction on company.id=transaction.company_id
103   where company_name='Non Institute';
104
105 • select transaction.id,company_name,country
106   from company
107   join transaction on company.id=transaction.company_id
108   where country=(select distinct country
109     from company
110     join transaction on company.id=transaction.company_id
111     where company_name='Non Institute') and company_name!= 'Non Institute'; -- No devuelvo las de la empresa Non Institute
112

```

Result Grid Filter Rows: Search Export:

id	company_name	country
2B928E1C-EC14-A760-0A75-871477649D6A	Sed Nunc Ltd	United Kingdom
ACD2011A-A2B1-C365-41E1-2AB00C65147A	Sed Nunc Ltd	United Kingdom
4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	Non Magna LLC	United Kingdom
BC2B9A38-77B4-28CD-1FE8-14DED63E773	Non Magna LLC	United Kingdom
1479B3D2-B7BA-C7BB-4CE3-8D7C2DE85ABB	Enim Condimentum Ltd	United Kingdom
152598C2-029D-D684-4B66-91EDF393EBFF	Enim Condimentum Ltd	United Kingdom

Result 23

Action Output Time Action Response

1	12:59:13	select transaction.id,company_name,country from company join transaction on company.id=transaction.company_id where country=(select distinct country...	70 row(s) returned
---	----------	---	--------------------

En la **primera query** obtengo el país al que pertenece la empresa Non Institute.

En la **segunda query**, mediante la JOIN uno ambas tablas de la base de datos y mediante la primera consulta a modo de **sub-query** filtro (**where**) la consulta por el país de la empresa Non Institute.

Decido no mostrar las transacciones realizadas por la empresa Non Institute con el **AND** en el filtro WHERE.

2.3.2.- Mostra el llistat aplicant solament subconsultes.

```

113  -- 2.3.2.- Mostra el llistat aplicant solament subconsultes.
114 •  select country                                -- Devuelve el país al que pertenece la empresa Non Institute
115   from company
116   where company_name='Non Institute';
117
118 •  select distinct transaction.id,country,company_name
119   from company,transaction                         -- UNION entre ambas tablas
120   where (company.id=transaction.company_id) and (country=(select country
121                                         from company
122                                         where company_name='Non Institute')) and company_name!='Non Institute';
123
100%  4:113

```

Result Grid Filter Rows: Search Export:

id	country	company_name
2B92BE1C-EC14-A760-0A75-B71477649D6A	United Kingdom	Sed Nunc Ltd
ACD2011A-A2B1-C365-41E1-2AB00C65147A	United Kingdom	Sed Nunc Ltd
4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	United Kingdom	Non Magna LLC
BC2B9A38-77B4-28CD-1FEB-14DED863E773	United Kingdom	Non Magna LLC
1479B3D2-B7BA-C7BB-4CE3-8D7C2DE85ABB	United Kingdom	Enim Condimentum Ltd
152598C2-029D-D684-4B66-91EDF393EBFF	United Kingdom	Enim Condimentum Ltd

Result 25

Action Output

Time	Action	Response
13:01:58	select distinct transaction.id,country,company_name from company,transaction	-- UNION entre ambas tablas where (company.id=transa... 70 row(s) returned

En la **primera query** obtengo el país al que pertenece la empresa Non Institute y que utilizo como **subquery** de la consulta definitiva para filtrar las empresas que pertenecen al mismo país.
La segunda query está formada por la **COMBINACIÓN** de tablas (que no JOIN ni UNION) para devolver las transacciones realizadas desde el mismo país que Non Institute.

NIVELL 3

3.1.- 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. Ordена els resultats de major a menor quantitat.

```

120  -- 3.1.- Presenta el nom, telèfon, país, data i amount, d'aquelles empreses que van realitzar transaccions amb un valor comprès entre 100 i 200 euros
121  -- 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.
122
123 • select company_name, phone, country, cast(timestamp as date), amount
124   from company
125   join transaction on company.id=transaction.company_id
126   where (amount between 100 and 200) and (timestamp like ('%2021-04-29%') or timestamp like ('%2021-07-20%') or timestamp like ('%2022-03-13%'))
127   order by amount desc;

```

Result Grid | Filter Rows: | Search | Export: |

company_name	phone	country	cast(timestamp as da...)	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

Result 83

Action Output | Time | Action | Response

1 12:56:20 select company_name, phone, country, cast(timestamp as date), amount from company join transaction on company.id=transaction.company_id where (amount between 100 and 200) 5 row(s) returned

En esta query con la JOIN accedo a datos de las dos tablas que forman la base de datos.

Mediante el filtro where selecciono los datos solicitados, por un lado con el operador BETWEEN para acotar el margen respecto a los valores de las transacciones y por otro lado con el operador LIKE combinado con la wildcard (o comodín) % para filtrar por las fechas indicadas.

3.2.- 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 específiquis si tenen més de 4 transaccions o menys.

Mediante la sentencia **CASE** introduzco un nuevo campo a la consulta, en el que en función de la condición **WHEN** tiene un valor u otro.

```

192 -- Opción case
193 • select company_name, count(amount) as 'NUMERO DE TRANSACCIONES',
194 case
195     when count(amount)>4 then 'Mas de 4 transacciones'
196     when count(amount)<4 then 'Menos de 4 transacciones'
197     else '4 transacciones'
198 end as 'TRANSACCIONES'
199 from company
200 join transaction on company.id=transaction.company_id
201 group by company_name
202 order by company_name asc;

```

Result Grid Filter Rows: Search Export:

company_name	NUMERO DE TRANSACCIONES	TRANSACCIONES
A Institute	2	Menos de 4 transacciones
Ac Fermentum Incorporated	2	Menos de 4 transacciones
Ac Industries	2	Menos de 4 transacciones
Ac Libero Inc	2	Menos de 4 transacciones

Result 31

Action Output

	Time	Action	Response
	1 13:20:24	select company_name, count(amount) as 'NUMERO DE TRANSACCIONES', case when count(amount)>4 then 'Mas de 4 t...' 100 row(s) returned	

Otra opción para resolver la consulta, pero menos eficiente de la anterior sería la siguiente.

Paso 1: **2 consultas resumen** en las que devuelvo el número de transacciones realizadas por cada empresa. En una filtrando las que tienen **más de cuatro** y en la otra las que tienen **igual o menos**. La suma de los registros devueltos por ambas queries suma **100**, que son el total de empresas de la tabla company.

```

129 -- 3.2.- Necessitem optimitzar l'assignació dels recursos i depèndrà de la capacitat operativa que es requereixi,
130 -- per la qual cosa et demanen la informació sobre la quantitat de transaccions que realitzen les empreses,
131 -- però el departament de recursos humans és exigent i vol un llistat de les empreses on especificuis si tenen més de 4 transaccions o menys.
132
133 • select company_name, count(transaction.id) as 'NUMERO DE TRANSACCIONES'          -- Tabla resumen de empresas con más de 4 transacciones (QueryMasCuatro)
134   from company
135   join transaction on company.id=transaction.company_id
136   group by company_name
137   having count(transaction.id)>4
138   order by count(transaction.id) desc;
139
140 • select company_name, count(transaction.id) as 'NUMERO DE TRANSACCIONES'          -- Tabla resumen de empresas con 4 o menos transacciones (QueryMenosCuatro)
141   from company
142   join transaction on company.id=transaction.company_id
143   group by company_name
144   having count(transaction.id)<=4
145   order by count(transaction.id) desc;

```

Result Grid		Filter Rows: <input type="text"/> Search	Export:
company_name	NUMERO DE TRANSACCIONES		
Nunc Interdum Incorporated	105		
Ut Semper Foundation	59		
Enim Condimentum Ltd	57		
Arcu LLP	56		
Lorem Eu Incorporated	54		
Malesuada PC	52		
Non Institute	30		
		Result 6	
Action Output	⌚	Time	Action
⌚ 1	15:51:09	select company_name, count(transaction.id... 7 row(s) returned	Response

Result Grid		Filter Rows: <input type="text"/> Search	Export:
company_name	NUMERO DE TRANSACCIONES		
Ac Fermentum Incorporated	2		
Magna A Neque Industries	2		
Fusce Corp.	2		
Convallis In Incorporated	2		
Ante Iaculis Nec Foundation	2		
Donec Ltd	2		
Sed Nunc Ltd	2		
Amet Nulla Donec Corporation	2		
Nascetur Ridiculus Mus Inc.	2		
		Result 8	
Action Output	⌚	Time	Action
⌚ 1	15:52:02	select company_name, count(transaction.id... 93 row(s) returned	Response

Paso 2: en las siguientes queries creo dos nuevas tablas a partir de las consultas del Paso 1 mediante las sentencias CREATE TABLE, es decir, convierto las 2 consultas en tablas.

```

147 -- Conversión de las consultas anteriores en tablas
148 • create table masCuatro as
149   select company_name
150   from (select company_name, count(transaction.id) as 'NUMERO DE TRANSACCIONES'          -- QueryMasCuatro
151     from company
152     join transaction on company.id=transaction.company_id
153     group by company_name
154     having count(transaction.id)>4
155     order by count(transaction.id) desc) as QueryMasCuatro;
156
157 • create table menosCuatro as
158   select company_name
159   from (select company_name, count(transaction.id) as 'NUMERO DE TRANSACCIONES'          -- QueryMenosCuatro
160     from company
161     join transaction on company.id=transaction.company_id
162     group by company_name
163     having count(transaction.id)<=4
164     order by count(transaction.id) desc) as QueryMenosCuatro;

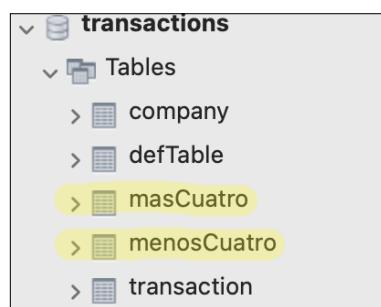
```

Paso 3: en las dos siguientes queries **añado una nueva columna** a cada una de las nuevas tablas creadas, para indicar con SI o con NO para indicar si tienen más de 4 transacciones realizadas.

Para ello utilizo las sentencias **ALTER TABLE** combinadas con los **CONSTRAINT** (o restricciones) **NOT NULL** (para forzar a que no adquieran nunca el valor NULL) y **DEFAULT** para arrancar el valor a SI o NO.

```
166 -- Añadir una nueva columna a las nuevas tablas creadas
167 • alter table masCuatro add MAS_DE CUATRO_TRANSACCIONES char(2) not null default "SI"; -- No puede adoptar el valor NULL y por defecto es SI
168
169 • alter table menosCuatro add MAS_DE CUATRO_TRANSACCIONES char(2) not null default "NO"; -- No puede adoptar el valor NULL y por defecto es NO
```

Las tablas resultantes son:



Result Grid		Filter Rows:	Search	Export:
company_name	MAS_DE_CUATRO_TRANSACCIONES			
Ac Fermentum Incorporated	NO			
Magna A Neque Industries	NO			
Fusce Corp.	NO			
Convallis In Incorporated	NO			
Ante Iaculis Nec Foundation	NO			
Donec Ltd	NO			
Sed Nunc Ltd	NO			
Amet Nulla Donec Corporation	NO			
Nascetur Ridiculus Mus Inc.	NO			
menosCuatro 1				

Paso 4: creo una tabla con la sentencia **CREATE TABLE** como definitiva formada por la UNION de las dos anteriores.

Con la sentencia **UNION**, asegurando que las dos tablas tienen el mismo número de campos las concateno en la nueva tabla creada.

```

171 -- Crear una nueva tabla en la que unir las dos nuevas creadas con la nueva columna añadida
172 • create table defTable(
173     company_id varchar(20),
174     MAS_DE_CUATRO_TRANSACCIONES char(2));
175
176 • select *
177   from defTable
178 union
179   select *
180   from masCuatro
181 union
182   select *
183   from menosCuatro;

```

100% 4:171

Result Grid		Filter Rows: <input type="text"/> Search	Export:
company_id	MAS_DE_CUATRO_TRANSACCIONES		
Nunc Interdum Incorporated	SI		
Ut Semper Foundation	SI		
Enim Condimentum Ltd	SI		
Arcu LLP	SI		
Lorem Eu Incorporated	SI		
Malesuada PC	SI		
Non Institute	SI		
Ac Fermentum Incorporated	NO		
Magna A Neque Industries	NO		
-	-		

Result 10

Action Output	Time	Action	Response
7	16:11:20	select * from defTable union select * from masCuatro union select * from menosCuatro	100 row(s) returned