

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.

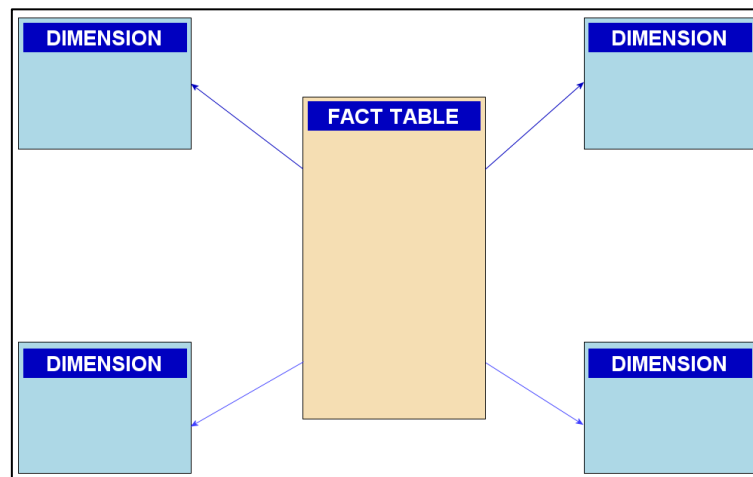
- Para crear la base de datos, primero ejecutamos el archivo 'estructura_dades.sql'. En este archivo se crea la base de datos llamada 'transactions', junto con sus dos tabla, 'company' y 'transaction'.

- 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

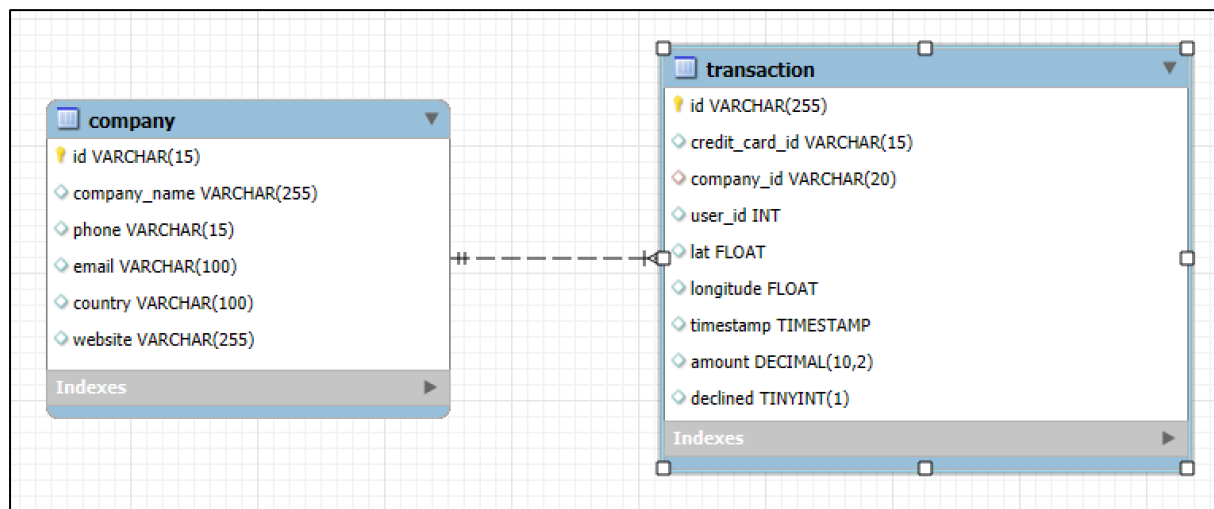


Diagrama entidad relacion

- En el diagrama entidad-relacion, 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 relacion entre 'company' y 'transaction', es de 1:N, porque una compañía puede tener multiples 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 transaccion
declined	BOOLEAN	Valor 1= TRUE -> Transaccion 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 paises sin repeticiones utilizando DISTINCT en la selección de datos.

The screenshot shows a SQL IDE window titled 'tasca2' with a tab for 'Query 1' containing an SQL file. The query is as follows:

```
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;
```

Below the query editor, the 'Result Grid' is displayed, showing a table with one column, 'country', and eight rows of data:

country
Germany
Australia
United States
New Zealand
Norway
United Kingdom
Italy

At the bottom, the 'Output' pane shows the 'Action Output' for the query execution:

#	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 países que están 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.

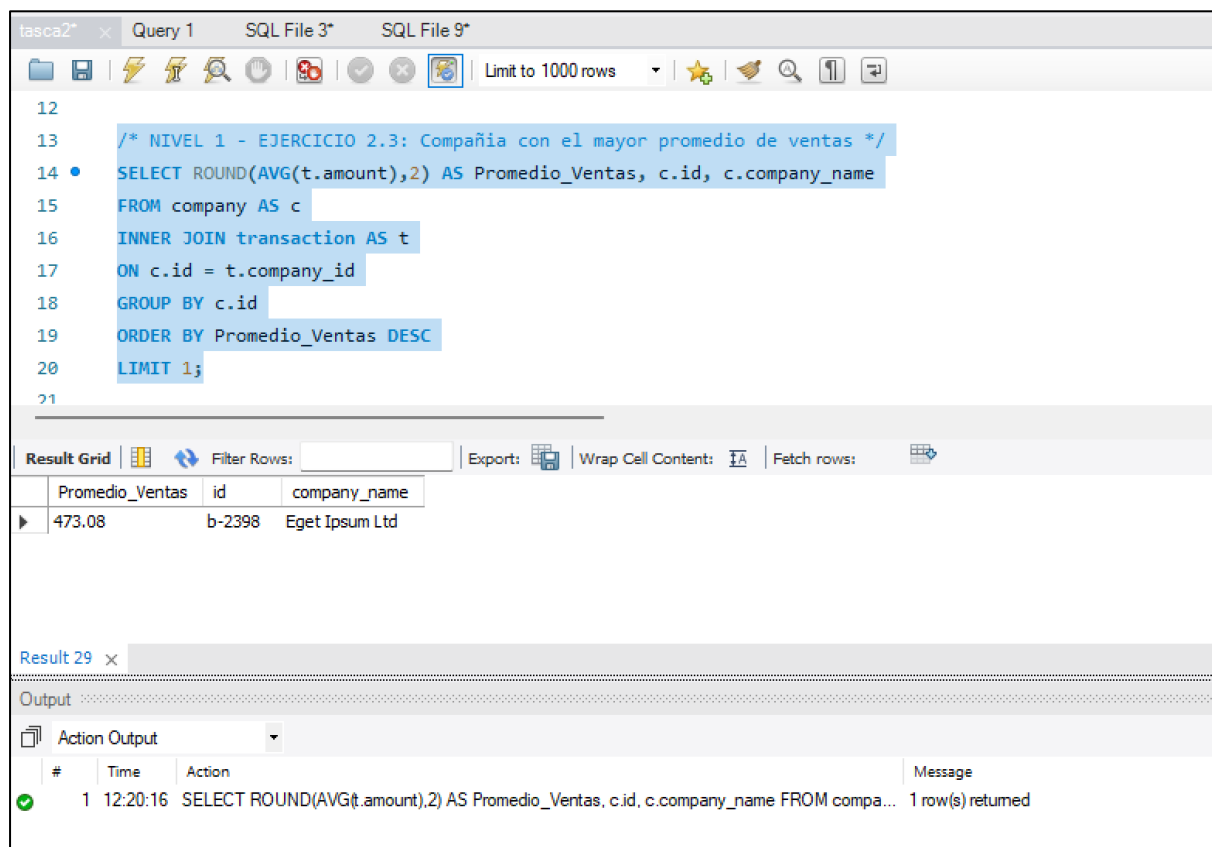
- Para obtener la compañía con la media más grande de ventas, tenemos que extraer el promedio de los montos de las transacciones, con la función AVG sobre la variable 'amount'

- 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 pane. The query editor contains the following SQL code:

```

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 pane displays a single row of data in a table format:

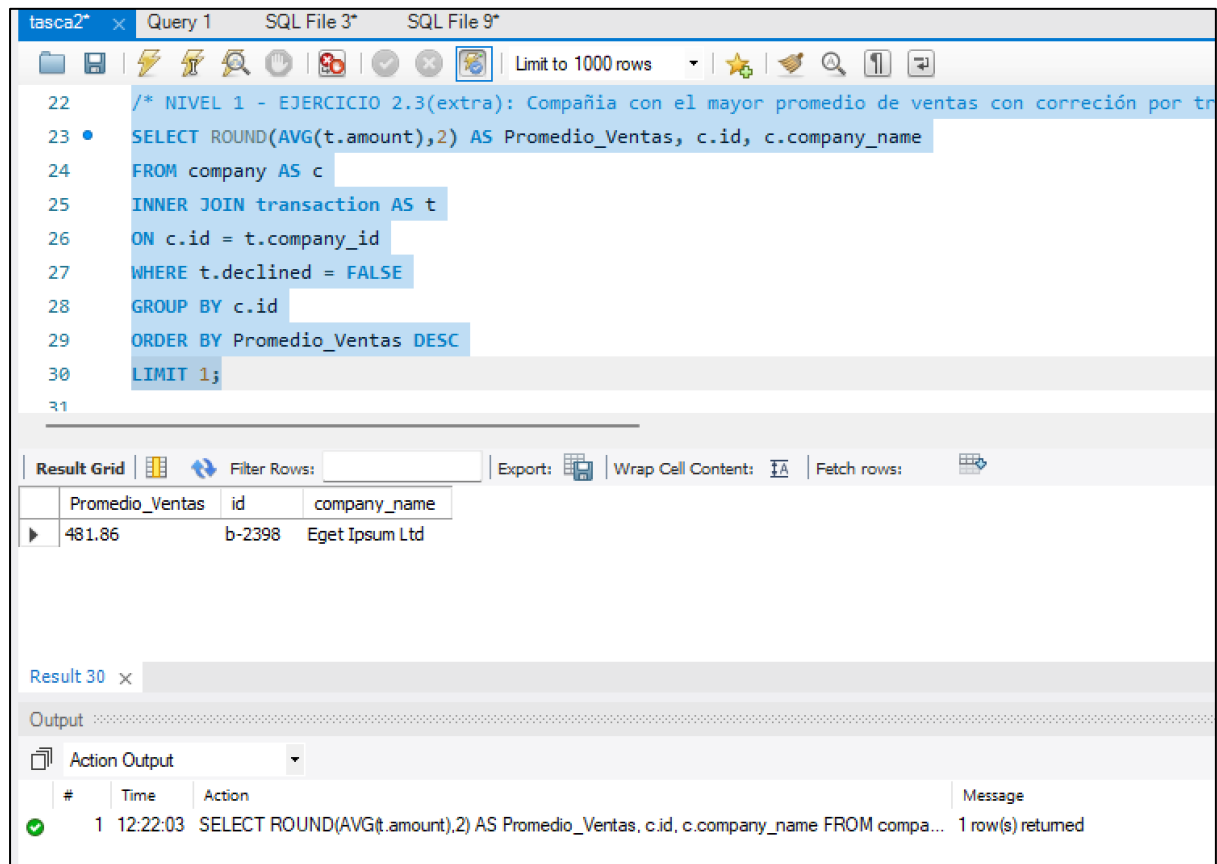
Promedio_Ventas	id	company_name
473.08	b-2398	Eget Ipsum Ltd

Below the results pane, the 'Output' section shows 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 grid. The query is as follows:

```
22 /* NIVEL 1 - EJERCICIO 2.3(extra): Compañia 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 grid shows the following data:

Promedio_Ventas	id	company_name
481.86	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	12:22:03	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 cláusula 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 result 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 result 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 unicamente los identificadores de empresa 'company_id', sin repeticiones (las repeticiones se pueden eliminar tambien 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 clausula 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 declinada, especificando mediante una clausula 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
```

Below the query editor is the 'Result Grid' section. It has a 'Filter Rows' field and an 'Export' button. The results are displayed in 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.

Below the result grid is the 'Output' section. It has a dropdown menu set to 'Action Output'. The output shows a single message:

#	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 clausula 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 the following content:

Query 1

```

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

```

Result Grid

company_name
company 16

Output

company 16 x

Output

Action Output

#	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).

