

[Ver scripts SQL](#)[Ver Sprint en PDF](#)**Nivel 1**

Ejercicio 1: Esquema de Base de Datos: Empresa y Transacción

El esquema creado incluye dos tablas principales: `company` y `transaction`. A continuación, se describen estas tablas y las variables que contienen, así como las relaciones entre ellas.

Tabla `company`

Esta tabla almacena información sobre las empresas involucradas en las transacciones. Contiene los siguientes campos:

- `id` : Identificador único de la empresa (`VARCHAR(15)`).
- `nombre_empresa` : Nombre de la empresa (`VARCHAR(255)`).
- `teléfono` : Número de teléfono de la empresa (`VARCHAR(15)`).
- `email` : Dirección de correo electrónico de la empresa (`VARCHAR(100)`).
- `país` : País de la empresa (`VARCHAR(100)`).
- `sitio web` : Sitio web de la empresa (`VARCHAR(255)`).

Tabla `transaction`

Esta tabla registra las transacciones realizadas y contiene los siguientes campos:

- `id` : Identificador único de la transacción (`VARCHAR(255)`).

- `credit_card_id` : Identificador de la tarjeta de crédito utilizada en la transacción (VARCHAR(15)).
- `company_id` : Identificador de la empresa involucrada en la transacción (VARCHAR(20)), referencia a `company(id)`.
- `user_id` : Identificador del usuario que realiza la transacción (INT), referencia a `user(id)`.
- `lat` : Latitud de la ubicación donde se realizó la transacción (FLOAT).
- `longitud` : Longitud de la ubicación donde se realizó la transacción (FLOAT).
- `timestamp` : Marca de tiempo de la transacción (TIMESTAMP).
- `importe` : Monto de la transacción (DECIMAL(10, 2)).
- `declinado` : Indica si la transacción fue rechazada (BOOLEAN).

Relaciones entre las tablas

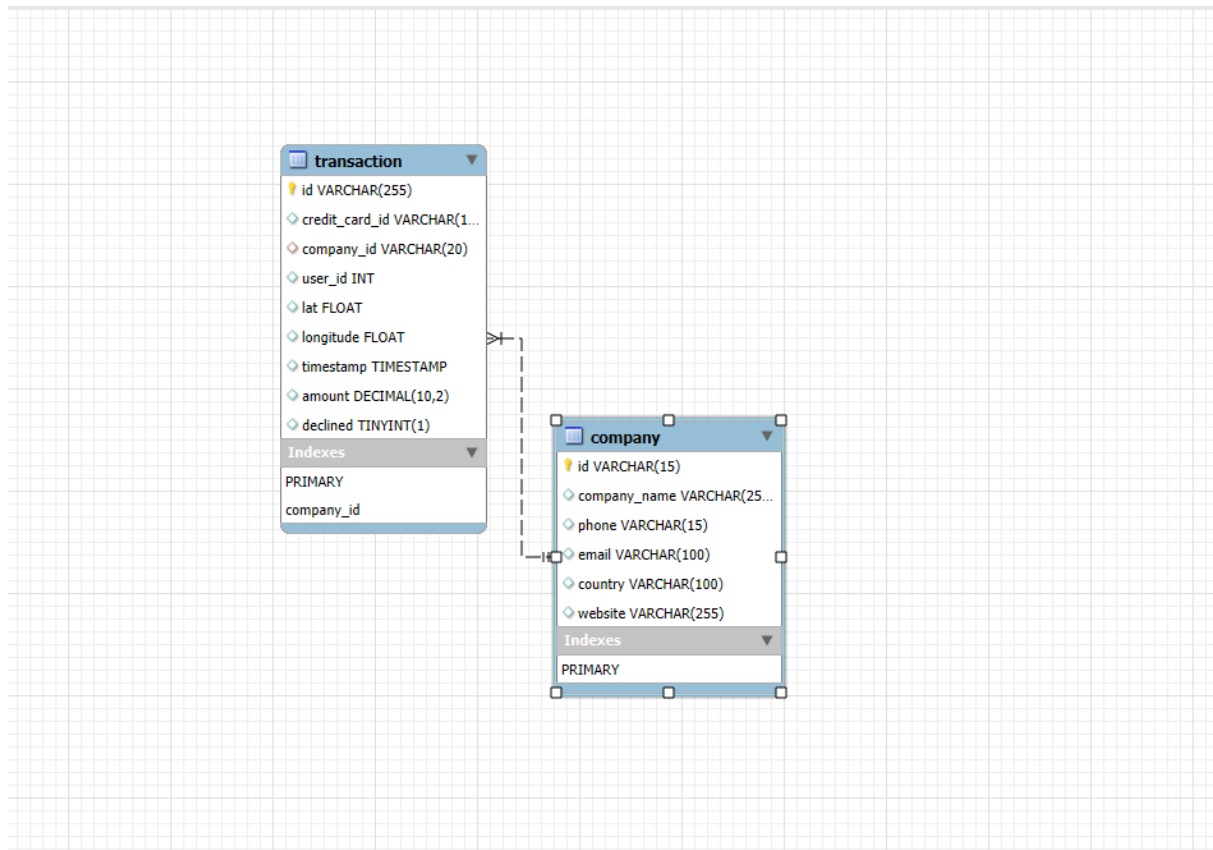
Las tablas están relacionadas de la siguiente manera:

- La tabla `transaction` tiene una clave foránea `company_id` que hace referencia a `company(id)`.
- La tabla `transaction` también tiene una clave foránea `user_id` que hace referencia a `user(id)`, aunque no se proporcionarán detalles de la tabla `user` en el esquema actual.

Diagrama del esquema

A continuación se presenta un diagrama que ilustra la relación entre las tablas `company`

A continuación se presenta un diagrama que ilustra la relación entre las tablas `company` y `transaction`:

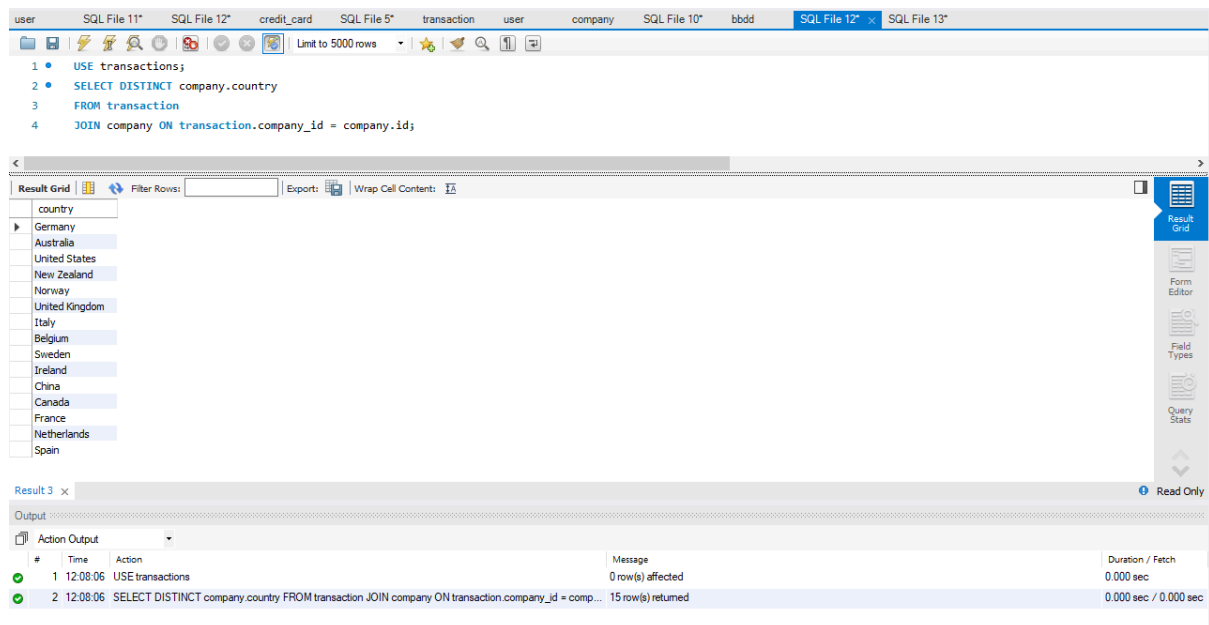


Ejercicio 2: Consultas SQL para el Esquema de Base de Datos

Consulta 1: Listado de los países que están realizando compras

```
SELECT DISTINCT company.country  
FROM transaction  
JOIN company ON transaction.company_id = company.id;
```

Imagen de Listado de los países que están realizando compras:



Explicación del Código

SELECT: Para especificar qué columnas queremos recuperar de la base de datos.

DISTINCT: Para eliminar filas duplicadas del resultado. En este caso, asegurará que solo se devuelvan países únicos.

empresa.país: Para seleccionar la columna `country` de la tabla `company`.

FROM transacción: Para recuperar los datos de la tabla `transaction`.

ÚNETE a la empresa ON transaction.company_id = company.id:

- **JOIN:** Para combinar filas de dos o más tablas basadas en una condición relacionada entre ellas.
- **compañía:** Es la tabla que estamos uniendo con la tabla `transaction`.
- **ON transaction.company_id = company.id:** especifica la condición de combinar las filas de ambas tablas donde el valor de `company_id` de la tabla `transaction` coincide con el valor de `id` de la tabla `company`.

Lo que hace la consulta:

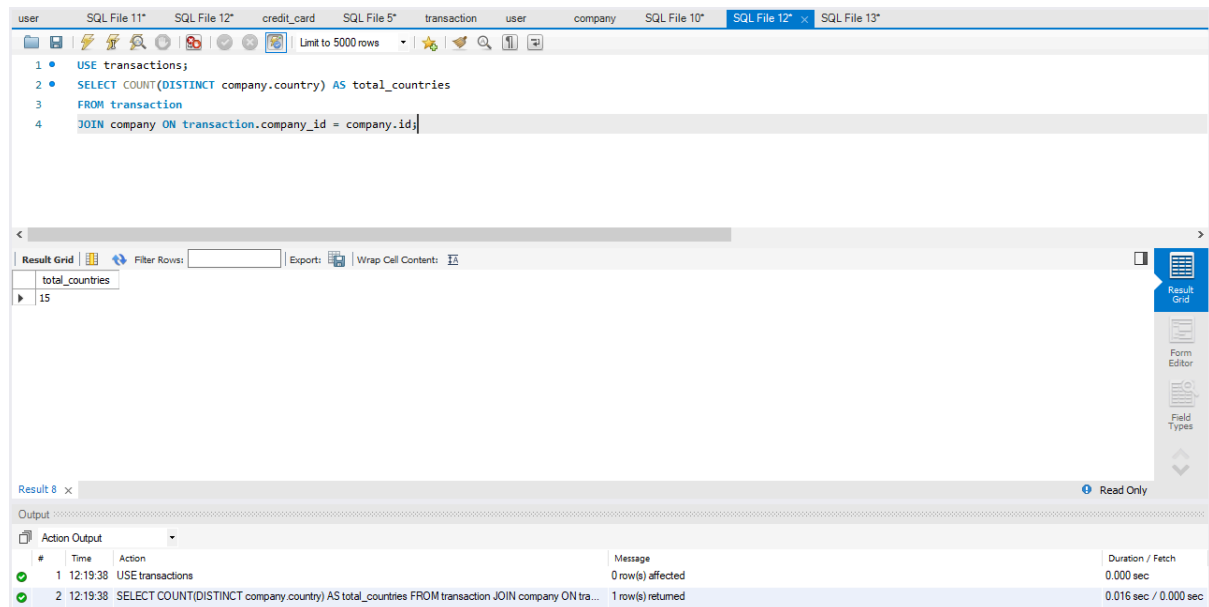
- **Unión de tablas:** unir la tabla `transaction` con la tabla `company` utilizando el campo `company_id` en `transaction` y el campo `id` en `company` para acceder a los datos de ambas tablas en una sola consulta.
- **Selección de países únicos:** Después de unir las tablas, la consulta selecciona los valores únicos de la columna `country` de la tabla `company`.

Consulta 2: ¿Cuántos países se realizan las compras?

```
SELECT COUNT(DISTINCT company.country) AS total_countries
```

FROM transaction
JOIN company ON transaction.company_id = company.id;

Imagen de Listado de cuántos países se realizan las compras:



Explicación del Código

SELECCIONAR CONTAR(DISTINCT empresa.país) COMO total_países:

- **SELECT:** Para especificar qué columnas o expresiones recuperar de la base de datos.
- **COUNT(DISTINCT company.country):** La función COUNT para contar el número de filas. DISTINCT dentro de COUNT, para contar el número de valores únicos en la columna company.country.
- **AS total_countries:** La cláusula AS para dar un alias al resultado total_countries.

FROM transacción: Para recuperar los datos de la tabla transaction.

ÚNETE a la empresa ON transaction.company_id = company.id:

- **ÚNETE:** Para combinar la tabla company con la tabla transaction.
- **ON transaction.company_id = company.id:** condición que indica para combinar las filas de ambas tablas donde el valor de company_id de la tabla transaction coincide con el valor de id de la tabla company.

Lo que hace la consulta:

- **Unión de tablas:** La consulta une la tabla transaction con la tabla company utilizando el campo company_id en transaction y el campo id en company.

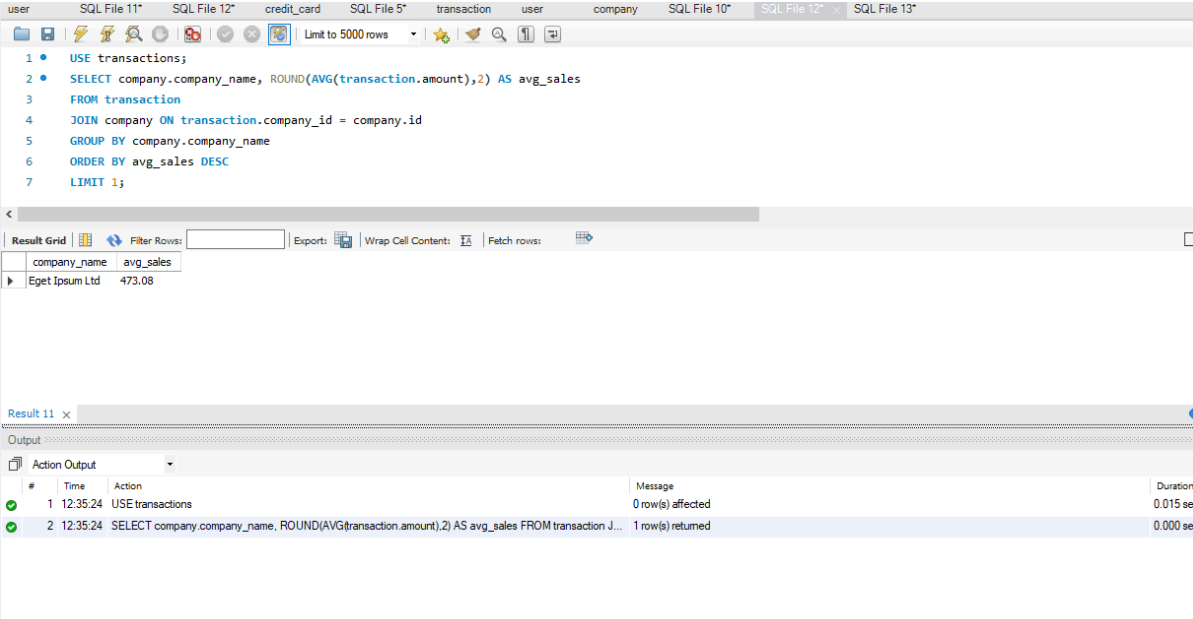
company. Esto permite acceder a los datos de ambas tablas en una sola consulta.

- Conteo de países únicos: Después de unir las tablas, la consulta cuenta el número de valores únicos en la columna country de la tabla company y le asigna el alias total_countries.

Consulta 3: Identificar a la compañía con la mayor media de ventas

```
SELECT company.company_name, ROUND(AVG(transaction.amount),2) AS avg_sales
FROM transaction
JOIN company ON transaction.company_id = company.id
GROUP BY company.company_name
ORDER BY avg_sales DESC
LIMIT 1;
```

Imagen de consulta Identifica a la compañía con la mayor media de ventas:



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

```
1 • USE transactions;
2 • SELECT company.company_name, ROUND(AVG(transaction.amount),2) AS avg_sales
3 FROM transaction
4 JOIN company ON transaction.company_id = company.id
5 GROUP BY company.company_name
6 ORDER BY avg_sales DESC
7 LIMIT 1;
```

The results pane displays a single row of data:

company_name	avg_sales
Eget Ipsum Ltd	473.08

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

#	Time	Action	Message	Duration /
1	12:35:24	USE transactions	0 row(s) affected	0.015 sec
2	12:35:24	SELECT company.company_name, ROUND(AVG(transaction.amount),2) AS avg_sales FROM transaction J...	1 row(s) returned	0.000 sec

Explicación del código:

- Unión de tablas: La consulta combina las filas de la tabla transaction con las filas de la tabla company donde transaction.company_id es igual a company.id. Esto permite acceder a los datos de ambas tablas en una sola consulta.
- Cálculo del promedio de ventas: Después de unir las tablas, la consulta calcula el promedio de amount para cada compañía (company_name).
- Redondeo del promedio: El promedio calculado se redondea a 2 decimales.

- Agrupación por compañía: La consulta agrupa los resultados por el nombre de la compañía (company_name).
- Ordenación y selección del mejor resultado: Los resultados se ordenan en orden descendente según el promedio de ventas (avg_sales), y se limita el resultado a la primera fila, que es la compañía con el promedio de ventas más alto.

Ejercicio 3: SubConsultas SQL sin utilizar JOIN

Consulta 1: Muestra todas las transacciones realizadas por empresas de Alemania.

```
SELECT *
FROM transaction
WHERE company_id IN (
    SELECT id
    FROM company
    WHERE country = 'Germany'
);
```

Imagen de consulta de todas las transacciones realizadas por empresas de Alemania.:

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, a search icon, and a 'Limit to 5000 rows' dropdown. The query editor contains the following SQL code:

```
2 • SELECT *
3 FROM transaction
4 WHERE company_id IN (
5     SELECT id
6     FROM company
7     WHERE country = 'Germany'
8 );
```

Below the query editor is the 'Result Grid' tab, which displays the results of the query. The results are shown in a table with the following columns: id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, and declined. The table contains 10 rows of data.

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
10881D1D-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-B99F43AD735A	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
0A476ED9-0C13-1962-F87B-D3563924B539	CcU-4359	b-2302	221	-56.4901	114.801	2022-02-26 20:33:54	430.49	0
122DC333-E19F-D629-DCD8-9C54CF1EBB9A	CcU-4366	b-2302	221	29.6372	-166.173	2021-06-09 06:04:14	172.01	0
135267BA-2E7D-957C-C42C-6450A2B3ED54	CcU-4520	b-2302	210	20.6724	14.9732	2021-12-29 20:38:23	17.97	0

Below the result grid is the 'Output' tab, which shows the execution log. It contains two entries:

#	Time	Action	Message
1	12:37:54	USE transactions	0 row(s) affected
2	12:37:54	SELECT * FROM transaction WHERE company_id IN (SELECT id FROM company WHERE countr...	118 row(s) returned

Explicación del código:

- Subconsulta: `SELECT id FROM company WHERE country = 'Alemania'`
Seleccione todos los id de la tabla company donde se encuentra el país Germany para obtener la lista de id de las compañías que están en Alemania.
- Consulta principal: `SELECT * FROM transacción WHERE company_id IN`
Luego, la consulta principal selecciona todas las columnas (*) de la tabla transacción donde el company_id está en la lista de id obtenidos de la subconsulta.

Consulta 2: Lista las empresas que han realizado transacciones por un monto superior a la media de todas las transacciones.

USE transactions;
`select * from company where id in (select company_id from transaction where amount > (select avg(amount) from transaction));`

Imagen de consulta empresas que han realizado transacciones por un monto superior a la media de todas las transacciones:

The screenshot shows a database management tool interface. At the top, there are tabs for various SQL files. The main editor displays the following SQL query:

```
USE transactions;  
select * from company where id in ( select company_id from transaction where amount > (select avg(amount) from transaction));
```

Below the editor, the 'Result Grid' shows the results of the query. The results are displayed in a table with the following columns: id, company_name, phone, email, country, and website. The table contains 6 rows of data.

id	company_name	phone	email	country	website
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.tellus@yahoo.net	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-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/settings
b-2242	Donec Ltd	01 25 51 37 37	at.iaculis@hotmail.co.uk	Norway	https://hytimes.com/user/110

At the bottom of the interface, the 'Output' section shows the execution log. It indicates that the query was executed successfully, returning 70 rows.

Explicación del código:

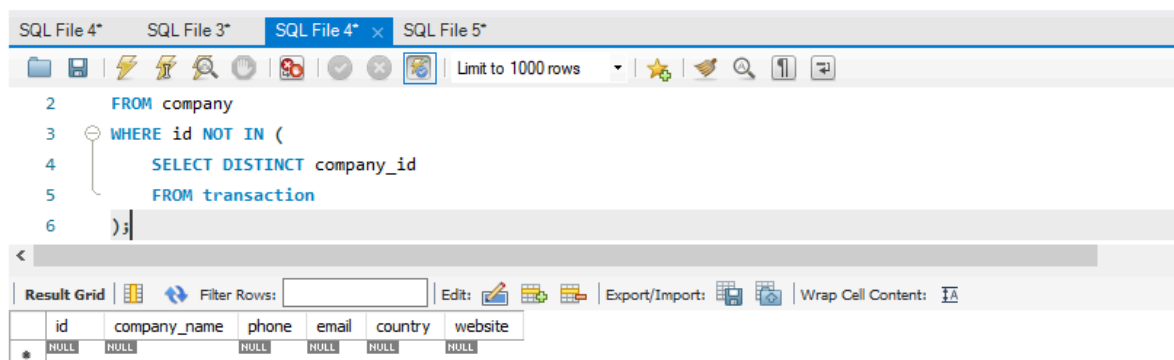
- Seleccionar la base de datos: `USE transacciones` La consulta empieza seleccionando la base de datos transactions.

- Subconsulta anidada SELECT AVG(amount) FROM transaction: La subconsulta más interna SELECT AVG(amount) FROM transaction calcula el promedio del campo amount en la tabla transaction.
- Subconsulta principal SELECT company_id FROM transaction WHERE importe > (...) Selecciona los company_id de la tabla transacción donde el importe es mayor que el promedio calculado por la subconsulta anidada.
- Consulta principal SELECT * FROM company WHERE id IN (...) Selecciona todas las columnas de la tabla company donde el id está en la lista de company_id obtenidos de la subconsulta principal.

Consulta 3: Eliminar del sistema las empresas que carecen de transacciones registradas, entrega el listado de estas empresas.

```
SELECT *
FROM company
WHERE id NOT IN (
    SELECT DISTINCT company_id
    FROM transaction
);
```

Imagen de consulta empresas que carecen de transacciones registradas:



Explicación del código:

- SELECCIONAR * DE empresa: Selecciona todas las columnas de la tabla company
- WHERE id NOT IN (SELECT DISTINCT company_id FROM transaction): Filtra las filas donde el id de la empresa no está en la lista de company_id de la tabla transaction.

Para probarlo inserte un registro en la tabla company

USE transactions;

```
INSERT INTO company (id, company_name, phone, email, country, website) VALUES ('b-2124', 'No tute', '06 77 15 31 14', 'amus@protonmail.couk', 'United Kingdom', 'https://gua.co.uk/settings');
```

Imagen de sql inserte un registro en la tabla empresa:



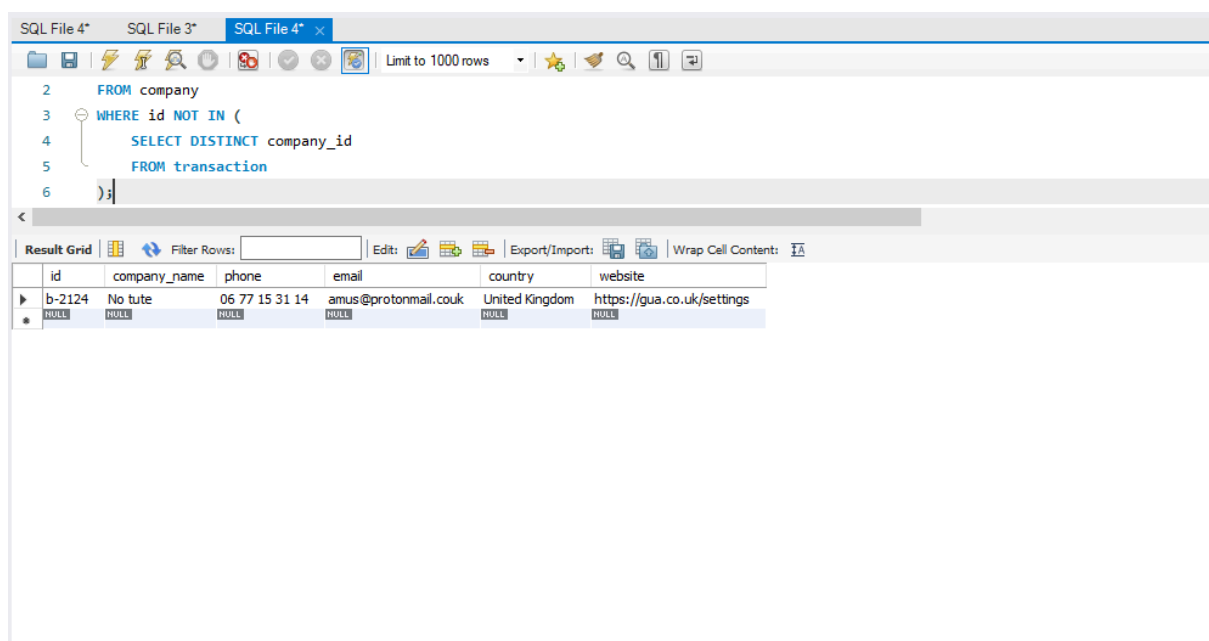
Explicación del código:

- SELECCIONAR * DE la empresa:INSERTAR EN la empresa (...) VALORES (..); Insertar un nuevo registro en la tabla 'company' para poder probar la consulta de eliminación.

Consulta empresas que carecen de transacciones registradas:

```
SELECT *  
FROM company  
WHERE id NOT IN (  
    SELECT DISTINCT company_id  
    FROM transaction  
);
```

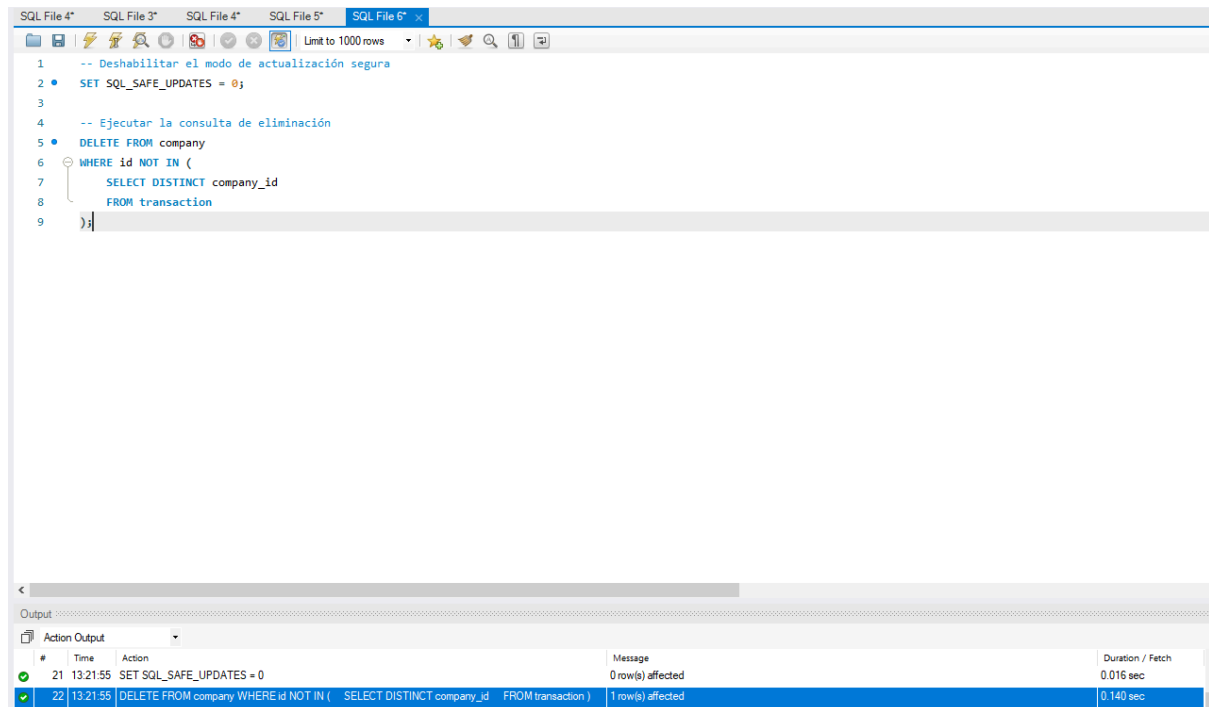
Imagen de aplicación de sql con el registro insertado:



Ejecutar la consulta de eliminación

```
DELETE FROM company
WHERE id NOT IN (
    SELECT DISTINCT company_id
    FROM transaction
);
```

Imagen de aplicación de sql con el registro insertado:



Explicación del código:

- ELIMINAR DE la empresa DONDE NO ESTÁ la identificación (SELECCIONE DISTINTO id_empresa DE la transacción); Eliminación de empresas que no tienen transacciones registradas.