

[Ver scripts SQL](#)

[Ver Sprint en PDF](#)



Sprint 2

Introducción

Este documento presenta el esquema de base de datos y las consultas SQL desarrolladas como parte del ejercicio de la Sprint 2. Se detallan las tablas involucradas, sus relaciones, y se proporcionan ejemplos de consultas con sus respectivas explicaciones.

Ejercicio 1: Esquema de Base de Datos

Tablas

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:

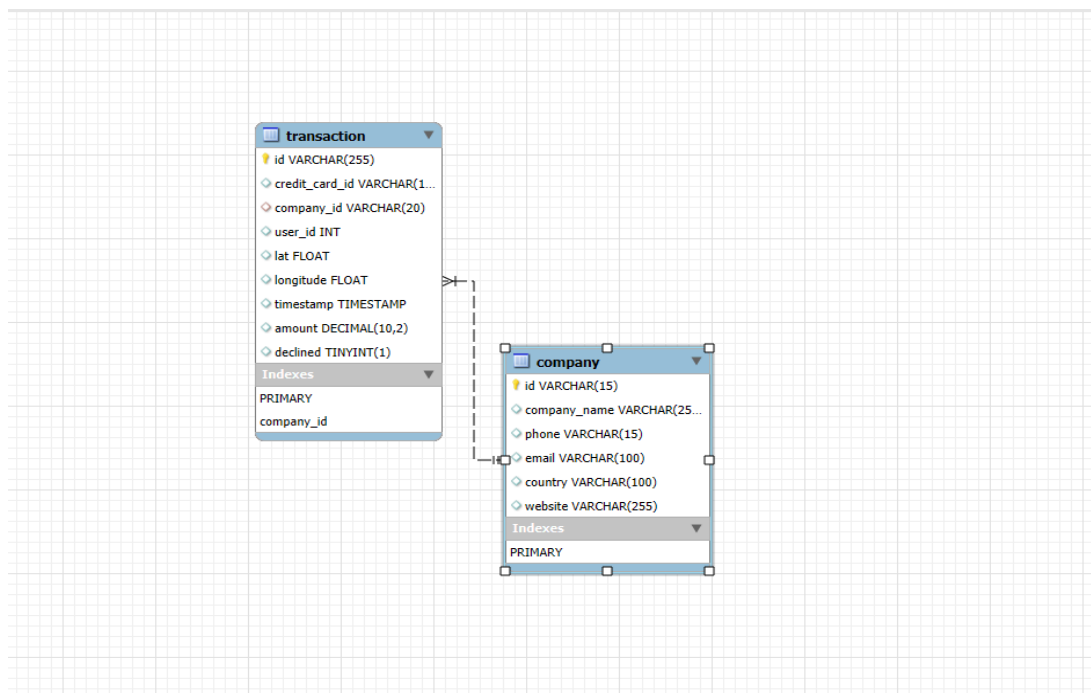
- **id**: Identificador único de la transacción (VARCHAR(255)).
- **credit_card_id**: Identificador de la tarjeta de crédito utilizada en la transacción (VARCHAR(15)).
- **company_id**: Identificador de la empresa involucrada en la transacción (VARCHAR(20)), referencia a **company(id)**.
- **user_id**: Identificador del usuario que realiza la transacción (INT), referencia a **user(id)**.
- **lat**: Latitud de la ubicación donde se realizó la transacción (FLOAT).
- **longitude**: 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)**.

Diagrama del Esquema



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

Consulta 1: Listado de los países que están realizando compras

Código:

sql

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

Explicación:

- **SELECT DISTINCT:** Elimina filas duplicadas del resultado.
- **company.país:** Selecciona la columna country de la tabla company.
- **FROM transaction:** Recupera los datos de la tabla transaction.
- **JOIN company ON transaction.company_id = company.id:** Combina filas de las tablas transaction y company basadas en company_id.

The screenshot shows a SQL IDE interface with a query editor at the top containing the following SQL code:

```
1 • USE transactions;
2 • SELECT DISTINCT company.country
3 • FROM transaction
4 • JOIN company ON transaction.company_id = company.id;
```

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

country
Germany
Australia
United States
New Zealand
Norway
United Kingdom
Italy
Belgium
Sweden
Ireland
China
Canada
France
Netherlands
Spain

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

#	Time	Action	Message	Duration / Fetch
1	12:08:06	USE transactions	0 row(s) affected	0.000 sec
2	12:08:06	SELECT DISTINCT company.country FROM transaction JOIN company ON transaction.company_id = comp...	15 row(s) returned	0.000 sec / 0.000 sec

Consulta 2: ¿Cuántos países realizan compras?

Código:

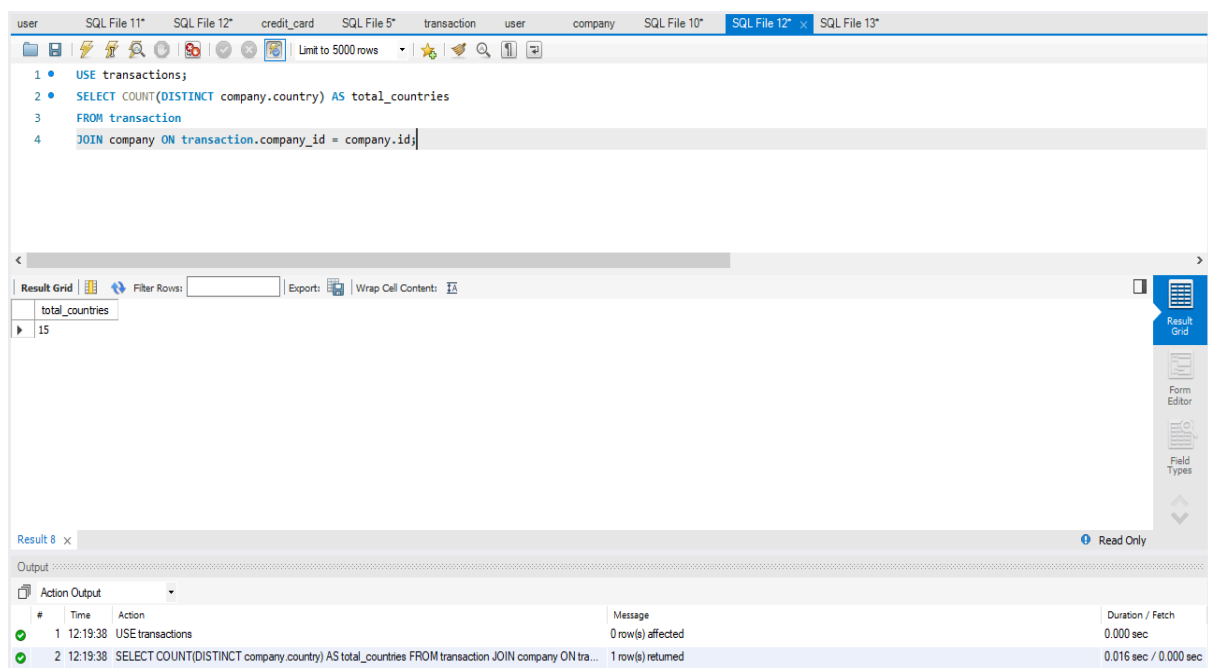
sql

```
SELECT COUNT(DISTINCT company.país) AS total_países
```

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

Explicación:

- `COUNT(DISTINCT company.país)`: Cuenta el número de países únicos.
- `AS total_países`: Alias para el resultado.
- `FROM transaction`: Recupera los datos de la tabla transaction.
- `JOIN company ON transaction.company_id = company.id`: Une las tablas transaction y company.



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

Código:

sql

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

Explicación:

- JOIN: Une las filas de las tablas transaction y company.
- AVG(transaction.importe): Calcula el promedio del importe de las transacciones.
- ROUND(. . . , 2): Redondea el promedio a 2 decimales.
- GROUP BY company.nombre_empresa: Agrupa los resultados por nombre de la empresa.
- ORDER BY avg_sales DESC: Ordena los resultados de mayor a menor promedio de ventas.
- LIMIT 1: Limita el resultado a la primera fila.

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

Ejercicio 3: SubConsultas SQL sin utilizar JOIN

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

Código:

```
SELECT * FROM transaction WHERE company_id IN ( SELECT id FROM company WHERE país = 'Alemania' );
```

Explicación:

- Subconsulta: Selecciona los id de empresas en Alemania.
- Consulta principal: Selecciona todas las transacciones donde company_id coincide con los id obtenidos.

SQL File 11* SQL File 12* credit_card SQL File 5* transaction user company SQL File 10* SQL File 12* x SQL File 13*

Limit to 5000 rows

```

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

```

Result Grid

	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

transaction 13 x

Output

Action Output

#	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

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

Código:

```

SELECT * FROM company WHERE id IN ( SELECT company_id FROM transaction
WHERE importe > ( SELECT AVG(importe) FROM transaction ) );

```

Explicación:

- Subconsulta anidada: Calcula el promedio del importe de las transacciones.
- Subconsulta principal: Selecciona los company_id con importe superior al promedio.
- Consulta principal: Selecciona todas las columnas de company donde id coincide con los company_id obtenidos.

SQL File 4* SQL File 3* SQL File 4* x

Limit to 1000 rows

```

2 FROM company
3 WHERE id NOT IN (
4     SELECT DISTINCT company_id
5     FROM transaction
6 )

```

Result Grid

	id	company_name	phone	email	country	website
▶	b-2124	No tute	06 77 15 31 14	amus@protonmail.couk	United Kingdom	https://gua.co.uk/settings

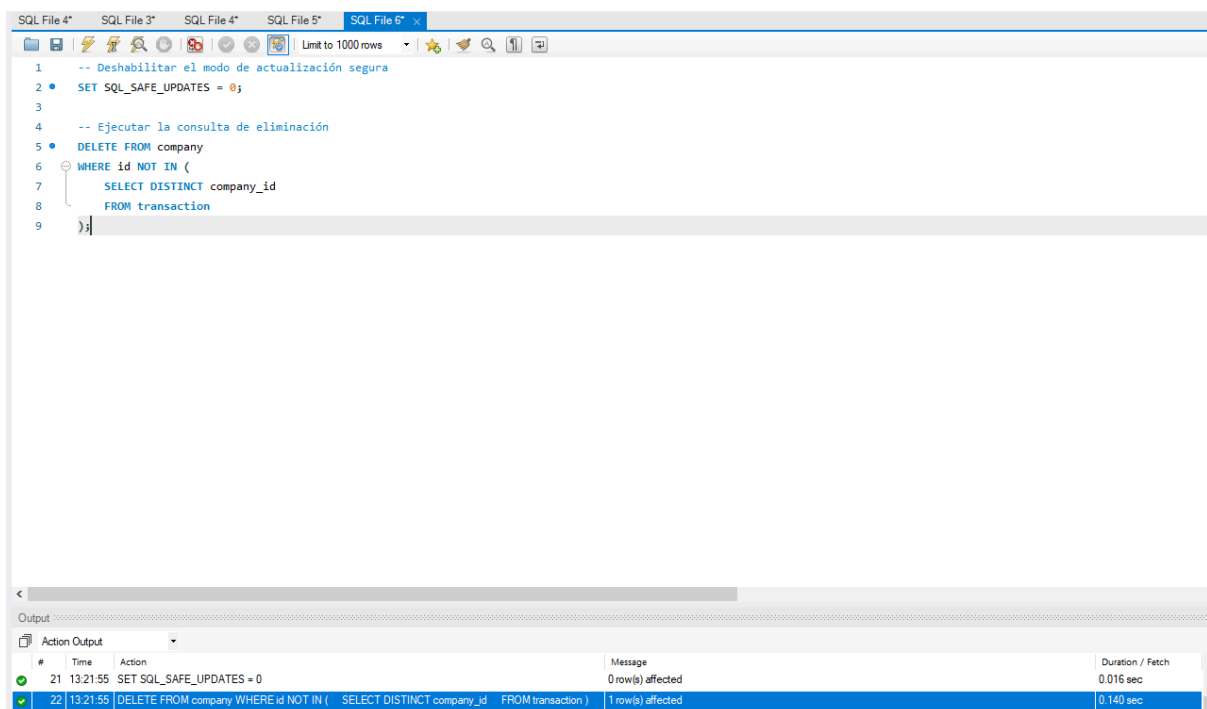
Consulta 3: Eliminar del sistema las empresas que carecen de transacciones registradas, entrega el listado de estas empresas

Código:

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

Explicación:

- Subconsulta: Selecciona los company_id únicos de la tabla transaction.
- Consulta principal: Selecciona todas las columnas de company donde id no coincide con los company_id obtenidos.



The screenshot shows a SQL IDE with a script editor and an output window. The script in the editor is as follows:

```
1 -- Deshabilitar el modo de actualización segura
2 SET SQL_SAFE_UPDATES = 0;
3
4 -- Ejecutar la consulta de eliminación
5 DELETE FROM company
6 WHERE id NOT IN (
7     SELECT DISTINCT company_id
8     FROM transaction
9 );
```

The output window at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
21	13:21:55	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.016 sec
22	13:21:55	DELETE FROM company WHERE id NOT IN (SELECT DISTINCT company_id FROM transaction)	1 row(s) affected	0.140 sec

Conclusión

Este documento ha presentado el esquema de base de datos y varias consultas SQL desarrolladas para la Sprint 2. Cada consulta incluye una explicación detallada para facilitar su comprensión y aplicación.