

# Sprint2 BBDD relacionales e introducción a SQL

## Tarea S2.01. Nociones básicas SQL

### - Nivel 1

#### - Ejercicio 1

A partir de los documentos adjuntos (*estructura\_dades* y *dades\_introduir*), importa las dos tablas. Muestra las características principales del esquema creado y explica las diferentes tablas y variables que existen. Asegurate de incluir un diagrama que ilustre la relación entre las diferentes tablas y variables.

Nuestra base de datos está formada por las tablas: *company* y *transaction*.

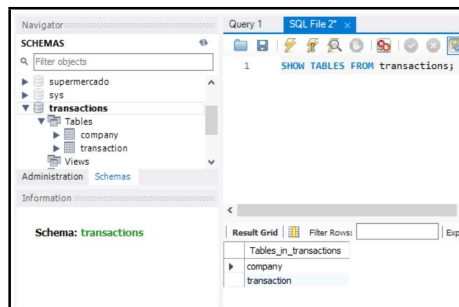


Imagen 1. Imagen de las tablas contenidas en mi DDBB *transactions*.

La tabla *company* contiene los siguientes campos:

- la clave primaria (o ID)
- nombre de la compañía
- teléfono, email
- país
- dirección web de la empresa
- .

En este caso, excepto la clave primaria, el resto son variables de texto.

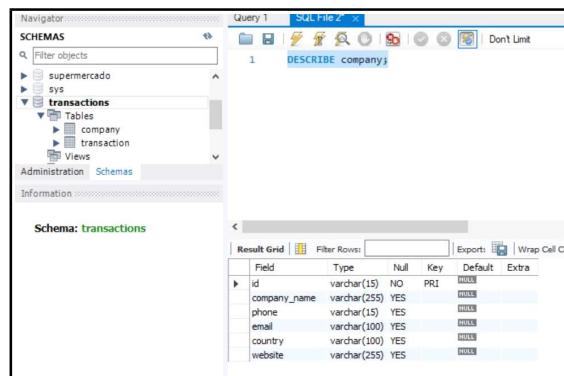
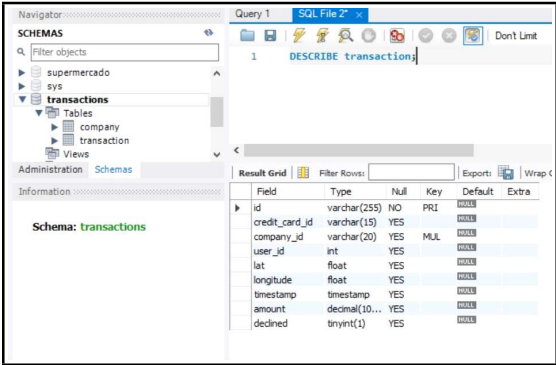


Imagen 2. Imagen de los campos que contenidos en la tabla *company*.

La tabla transaction contiene los siguientes campos:

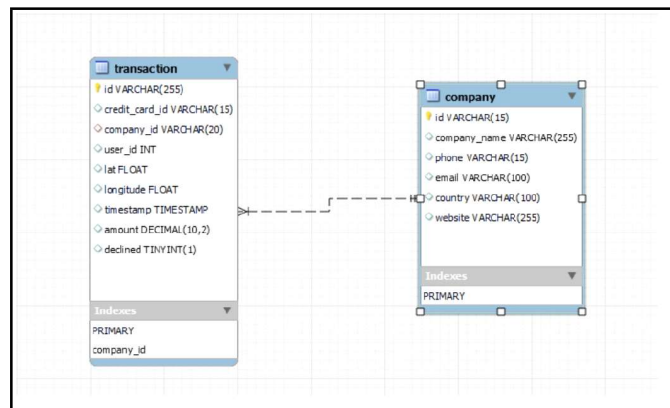
- id
- tarjeta de crédito
- id company (corresponde a la clave foranea)
- código de usuario
- Latitud
- Longitud
- día y hora de la transacción
- cantidad,



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

**Imagen 3.** Imagen de los campos que contenidos en la tabla *transaction*.

La relación entre las tablas de nuestro modelo es de 1:N. Una company tiene N registros en la tabla transacciones.



**Imagen 4.** Imagen de Diagrama Entidad Relación.

- **Ejercicio 2.** Utilizando JOIN realizarás las siguientes consultas.

### **2.1. Listado de los países que están haciendo compras.**

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

Primero se selecciona los country con una función DISTINCT que nos elimina los valores repetidos. Realizamos un JOIN para buscar los distintos países de la tabla *company* en la tabla *transaction*.

El resultado de la consulta es el siguiente:

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

### **2.2. Desde cuantos países se realizan las compras.**

```
SELECT count(DISTINCT country) as paises
FROM company
INNER JOIN transaction
ON company.id = transaction.company_id;
```

Resultado: Se realizan las compras desde 15 países diferentes

En este ejercicio, se repite la consulta del ejercicio anterior. Y se añade la función count para contar el número total de países de la consulta anterior.

### **2.3. Identifica la compañía con la media más gran de ventas.**

```
SELECT    company.company_name as empresa,
          round(avg(amount),2) as media_ventas
FROM company
INNER JOIN transaction
ON company.id = transaction.company_id
WHERE declined = 0
GROUP BY empresa
ORDER BY media_ventas DESC
LIMIT 1;
```

La compañía con la media más grande de ventas es: Eget Ipsum Ltd = 481.86

En esta consulta, Se busca listas los campos de nombre de la compañía y el average del campo amount. Este ultimo, como se trata de un campo de moneda se redondea a dos decimales con la función Round.

Hacemos una JOIN para relacionar las dos tablas que contienen la información a relacionar, y filtramos con un WHERE con la condición que el declined sea igual a cero. Esto quiere decir que las transacciones realizadas se hayan hecho con éxito.

Al contener un campo de agrupación en la SELECT se debe agrupar por el campo descriptivo company\_name.

Por último, ordenamos (ORDER BY) por la media\_ventas en orden descendente y seleccionamos el primer registro con un LIMIT 1.

### **- Ejercicio 3. Utilizando sólo subconsultas (sin utilizar JOIN)**

#### **3.1. Muestra todas las transacciones realizadas por empresas de Alemania.**

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

El resultado es de 118 transacciones.

En esta consulta, se realiza una subquery donde se selecciona los id de la tabla company, donde se cumpla la condición que el country sea igual a Germany. Esta lista se utiliza para filtrar através de un IN, todos los campos de cada registro de la tabla transaction.

#### **3.2. Lista las empresas que han realizado transacciones por un amount superior a la media de todas las transacciones.**

```
SELECT    company_id,  
          max(amount) as maximo  
FROM transaction  
GROUP BY company_id  
HAVING maximo > (SELECT avg(amount) as mediaTotal  
                FROM transaction);
```

-- El resultado es una lista formada por 70 países.

En este caso, se seleccionan los campos company\_id, y se busca el máximo del campo amount para cada company, asignándole un alias. Esto quiere decir que se agrupa por company\_id, al tener en la SELECT una función de agregación, es necesario añadir el GROUP BY.

Para obtener las empresas que tengan alguna transacción mayor a la media, se compara con un 'mayor que' (>), el valor máximo del amount por empresa con el average de todas las transacciones que hay en la tabla transaction.

Por último, en este caso, se calcula el máximo del amount por empresa, ya que si existe un valor que sea mayor a la media del total, ya cumpliría con la condición solicitada.

### 3.3. Eliminar del sistema las empresas que no tienen transacciones registradas, entrega el llistat d'aquestes empreses.

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

En esta consulta, se listarán los nombres de las compañías (`company_name`) de la tabla `company` que cumplen la condición (WHERE) que el campo `id` (de la tabla `company`) no se encuentre dentro de la siguiente subconsulta, con la función `NOT IN`. Esta es una lista con los distintos `company_id` de la tabla `transaction`. El resultado es una tabla vacía.

La conclusión es que todas las empresas de la tabla `company` tienen transacciones en la tabla `transcription`. NO es necesario eliminar ningún registro.

## - Nivel 2

### Ejercicio 1

*Identifica los cinco días que se generó la mayor cantidad de ingresos en la empresa por ventas. Muestra la fecha de cada transacción junto con el total de las ventas.*

```
SELECT    date(transaction.timestamp) as date,
          sum(transaction.amount) as total
FROM company
JOIN transaction
ON company.id = transaction.company_id
WHERE declined = 0
GROUP BY date
ORDER BY total DESC
LIMIT 5;
```

En esta consulta, se busca listar la fecha de la transacción (`transaction.timestamp`), a la cual se ha añadido una función `date` para cambiar el formato de la fecha para quedarse sólo con la fecha, y la `suma` (función de agregación) del `amount` agrupada por esa misma fecha. Al estar los valores de búsqueda en tablas diferentes se realiza una `JOIN`.

En este caso, al tratarse de ventas, se tiene en cuenta el `declined`, considerando solo los valores cero, que son los que tienen transacciones efectivas.

Para acabar, se ordena de manera descendiente por la suma del `amount`, y se limita a 5 los resultados. Así se seleccionan los 5 valores máximos.

### Ejercicio 2

*¿Cuál es el promedio de ventas por país? Presenta los resultados ordenados de mayor a menor.*

```
SELECT    country,
          round(avg(amount),2) as media
FROM company
INNER JOIN transaction
ON company.id = transaction.company_id
WHERE declined = 0
GROUP BY country
ORDER BY media DESC;
```

Para el cálculo de la media, se utiliza la función de agregación average sobre el campo amount. A ese valor se le aplica la función Round para limitar el valor a dos decimales, ya que es una moneda. Se lista el país y la media del amount que se agrupa por País. Ya que los datos requeridos se encuentran en tablas diferentes, se aplica una JOIN.

Por otro lado, como se está hablando de ventas aplicaremos el filtro para cumplan la condición de declined igual a cero que se ha explicado anteriormente,

Por último, se ordenan por la media del amount, de manera descendiente (ORDER BY DESC)

### **Ejercicio 3**

***En tu empresa, se plantea un nuevo proyecto para lanzar algunas campañas publicitarias para hacer competencia a la compañía "Non Institute". Para ello, te piden la lista de todas las transacciones realizadas por empresas que están situadas en el mismo país que esta compañía.***

#### **Muestra el listado aplicando JOIN y subconsultas.**

```
SELECT    company.company_name,
          transaction.id,
          transaction.timestamp,
          transaction.amount
FROM      company
INNER JOIN transaction
ON        company.id = transaction.company_id
WHERE     country = (SELECT country
                    FROM company
                    WHERE company_name = 'Non Institute')
AND       company_name <> 'Non Institute'
```

En la SELECT se detallan los campos: nombre de la compañía, id de la transacción, fecha de la transacción y el valor de la venta. Como los datos a buscar están en tablas distintas, se aplica una JOIN.

Se listan los datos que cumplan dos condiciones (las dos a la vez). La primera que el país sea el mismo que el de Non Institute que se obtiene de la subquery.

La segunda condición es que no sea la empresa Non Institute, ya que pide comparar el resto de empresas ubicadas en el mismo país que la empresa que pide el estudio.

#### **Muestra el listado aplicando solamente subconsultas.**

```
SELECT    company.company_name,
          transaction.id,
          transaction.timestamp,
          transaction.amount
FROM      company, transaction
WHERE     country = (SELECT country
                    FROM company
                    WHERE company_name = 'Non Institute')
AND       company_name <> 'Non Institute'
AND       company.id = transaction.company_id;
```

En el caso de sólo utilizar subconsultas, la diferencia con la anterior es que no se utiliza una JOIN, y se añade en el WHERE una condición más que es:

company.id = transaction.company\_id

Al utilizar AND, se deben cumplir las tres condiciones a la vez.

## - Nivel 3

### Ejercicio 1

*Presenta el nombre, teléfono, país, fecha y amount, de aquellas empresas que realizaron transacciones con un valor comprendido entre 100 y 200 euros y en alguna de estas fechas: 29 de abril de 2021, 20 de julio de 2021 y 13 de marzo de 2022. Ordena los resultados de mayor a menor cantidad.*

```
SELECT    company.company_name,
          company.phone, company.country,
          date(transaction.timestamp) as date,
          transaction.amount
FROM      company
INNER JOIN transaction
ON        company.id = transaction.company_id
WHERE     transaction.amount BETWEEN 100 AND 200
          AND
          (date(transaction.timestamp) IN ('2021-04-29', '2021-07-20', '2022-03-13'))
ORDER BY  amount DESC;
```

En esta consulta, se seleccionan los datos requeridos. Se aplica la función date para transformar el formato en solo fecha. Como los datos a listar se encuentran en diferentes tablas, se aplica una JOIN. Se aplica un WHERE (un filtro) con dos condiciones que se deben cumplir ambas. La primera es un BETWEEN del campo amount definiendo un rango, y la segunda es un IN donde la fecha debe coincidir con una de esas tres fechas. Para acabar, ordenamos por el campo cuantitativo amount de manera descendente.

### Ejercicio 2

*Necesitamos optimizar la asignación de los recursos y dependerá de la capacidad operativa que se requiera, por lo que te piden la información sobre la cantidad de transacciones que realizan las empresas, pero el departamento de recursos humanos es exigente y quiere un listado de las empresas donde especifiques si tienen más de 4 transacciones o menos.*

```
SELECT    company.company_name AS empresa,
          CASE      WHEN count(transaction.id) >= 4 THEN 'Mayor o igual a 4'
                    ELSE 'Menor a 4'
          END AS Cantidad
FROM      company
INNER JOIN transaction
ON        company.id = transaction.company_id
GROUP BY  company.id;
```

En esta última consulta, se listan el nombre de la empresa, y una nueva columna que recibe el nombre de Cantidad. Esta columna se define con la función CASE WHEN. Ésta compara el valor del conteo del número de las transacciones con un valor determinado. Si cumple la primera condición escribe, en este caso: 'Mayor o igual a 4', sino se escribe: 'Menor a 4'.