

Consultas SQL - Multitabla

Bases de Datos II

Tecnología en Desarrollo de Software - Ingeniería de Sistemas y
Telecomunicaciones

David Ricardo Rivera-Arbeláez

Un futuro con historias



Consultas multitabla

Permiten combinar información de dos o más tablas en una sola consulta. SQL proporciona varias formas de hacerlo, siendo las más comunes el uso de **JOINS** y **subconsultas**.

Aplicaciones de las consultas multitable

Las consultas multitable permiten responder preguntas complejas como:

- ¿Qué cliente hizo determinada compra?
- ¿Qué productos están incluidos en cada orden?
- ¿Cuál es el total gastado por cada cliente?

Sintaxis de una consulta multitabla

```
SELECT columnas  
FROM tabla1  
JOIN tabla2 ON condición  
[JOIN tabla3 ON condición]  
WHERE filtros  
GROUP BY agrupamiento  
ORDER BY ordenamiento;
```

Consultas multitabla

Uso de subconsultas: permiten anidar una consulta dentro de otra.

¿Cómo se relacionan las tablas?

Las relaciones entre tablas se construyen mediante:

- **Claves primarias (PRIMARY KEY):** identifican de forma única cada fila en una tabla.
- **Claves foráneas (FOREIGN KEY):** apuntan a claves primarias de otras tablas para establecer relaciones.

Tipos de JOINS en SQL

- INNER JOIN
- LEFT JOIN (LEFT OUTER JOIN)
- RIGHT JOIN (RIGHT OUTER JOIN)
- FULL JOIN (FULL OUTER JOIN)
- CROSS JOIN

Tipos de JOINS en SQL – INNER JOIN

Une registros **solo cuando hay coincidencias** en ambas tablas.

```
SELECT c.nombre, o.fecha  
FROM clientes c  
INNER JOIN ordenes o ON c.id = o.cliente_id;
```


Tipos de JOINS en SQL – LEFT OUTER JOIN

Muestra **todos los registros de la tabla izquierda**, y los de la derecha solo si coinciden.

```
SELECT c.nombre, o.fecha  
FROM clientes c  
LEFT JOIN ordenes o ON c.id = o.cliente_id;
```

Tipos de JOINS en SQL – RIGHT OUTER JOIN

Muestra **todos los registros de la tabla derecha**, y los de la izquierda solo si coinciden.

Tipos de JOINS en SQL – FULL OUTER JOIN

Muestra **todos los registros de ambas tablas**, aunque no haya coincidencia.

Tipos de JOINS en SQL – CROSS JOIN

Genera el **producto cartesiano** de dos tablas: **combina cada fila de la primera tabla con todas las filas de la segunda.**

- **No requiere una condición de unión (ON)**
- Se utiliza en casos específicos como:
 - Generar combinaciones posibles
 - Crear conjuntos base para reportes
 - Pruebas o análisis de cobertura

```
SELECT *  
FROM tabla1  
CROSS JOIN tabla2;
```

```
SELECT c.nombre AS cliente, p.nombre AS producto  
FROM clientes c  
CROSS JOIN productos p;
```

Tipos de JOINS en SQL – CROSS JOIN: Ejemplo de uso

Suponer que se quiere saber cuántas veces *podría* comprarse cada producto por cada cliente, para un análisis de proyección o simulación.

```
SELECT c.nombre AS cliente, p.nombre AS producto, 0 AS  
cantidad_inicial  
FROM clientes c  
CROSS JOIN productos p;
```

Ejercicio práctico

Consultas multitable en SQL desde VSCode con SQLite.

Requisitos previos

- Tener instalado [VSCode](#)
- Instalar la extensión: **SQLite** (autor: Alex Covizzi)

Creación del proyecto

1. Crear el proyecto

1. Abrir VSCode.
2. Crear una nueva carpeta para el proyecto (ej: consultas_multitabla_sql)
3. Dentro de la carpeta, crea un archivo llamado: **base_datos.sql**

Creación de las tablas

Copiar y pegar lo siguiente en **base_datos.sql**:

```
-- Crear tablas
DROP TABLE IF EXISTS clientes;
DROP TABLE IF EXISTS productos;
DROP TABLE IF EXISTS ordenes;
DROP TABLE IF EXISTS orden_detalle;
```

```
CREATE TABLE clientes (
  id INTEGER PRIMARY KEY,
  nombre TEXT
);
```

```
CREATE TABLE productos (
  id INTEGER PRIMARY KEY,
  nombre TEXT,
  precio REAL
);
```

Creación de las tablas - Continuación

Copiar y pegar lo siguiente en
base_datos.sql:

```
CREATE TABLE ordenes (  
  id INTEGER PRIMARY KEY,  
  cliente_id INTEGER,  
  fecha TEXT,  
  FOREIGN KEY (cliente_id) REFERENCES clientes(id)  
);
```

```
CREATE TABLE orden_detalle (  
  id INTEGER PRIMARY KEY,  
  orden_id INTEGER,  
  producto_id INTEGER,  
  cantidad INTEGER,  
  FOREIGN KEY (orden_id) REFERENCES ordenes(id),  
  FOREIGN KEY (producto_id) REFERENCES productos(id)  
);
```


Insertar datos

Debajo de la creación de tablas, pegar esto:

-- Insertar clientes

```
INSERT INTO clientes VALUES (1, 'Ana'), (2, 'Luis');
```

-- Insertar productos

```
INSERT INTO productos VALUES (1, 'Laptop', 1200.00), (2, 'Mouse', 25.00), (3, 'Teclado', 45.00);
```

-- Insertar órdenes

```
INSERT INTO ordenes VALUES (1, 1, '2025-03-01'), (2, 2, '2025-03-02');
```

-- Insertar detalles de órdenes

```
INSERT INTO orden_detalle VALUES  
(1, 1, 1, 1),  
(2, 1, 2, 2),  
(3, 2, 2, 1),  
(4, 2, 3, 1);
```

Crear y abrir la base de datos SQLite

1. Crear un archivo vacío en la misma carpeta llamado **tienda.db**
2. Abrir la **barra lateral de SQLite** (ícono con forma de base de datos).
3. Hacer clic en "**Open Database**" y seleccionar **tienda.db**.

Ejecutar el script SQL para crear tablas y poblar datos

1. Abrir la vista de comandos (Ctrl + Shift + P)
2. Escribir y seleccionar: SQLite: Run Query
3. Ejecutar el contenido de base_datos.sql
4. Verificar que las tablas se han creado: ir a la barra lateral de SQLite y revisar que aparecen clientes, productos, ordenes, orden_detalle

Crear archivo de consultas

1. Crear un nuevo archivo llamado consultas.sql
2. Copiar y pegar consultas prácticas. Ver: [consultas.sql](#)
3. Seleccionar cada bloque de consulta y ejecutarlo con **SQLite:**
Run Selected Query

Sugerencias para exportar resultados

- Es posible hacer clic derecho sobre una tabla o resultado y elegir “Export as CSV” para análisis posterior.
- Se puede conectar esta base de datos a Python o Node.js en siguientes niveles.

Actividad práctica en clase

Consultas Multitabla con Base de Datos Abierta (Kaggle)

Objetivo de la actividad

Aplicar conocimientos sobre consultas multitabla (JOIN, INNER, LEFT, CROSS, GROUP BY, subconsultas) utilizando una base de datos real de Kaggle, para responder preguntas clave de análisis de datos y extraer insights desde múltiples tablas relacionadas.

Actividad práctica en clase: Enunciado

Seleccionar un conjunto de datos relacional (es decir, que incluya **dos o más tablas relacionadas**) desde una fuente abierta como Kaggle, y realizar una exploración mediante **consultas multitabla en SQL**. Se deberá diseñar y ejecutar al menos **cinco consultas significativas**, incluyendo:

- 1.Consultas con **INNER JOIN** y **LEFT JOIN**.
- 2.Consultas con **subconsultas** (SELECT anidados).
- 3.Uso de **funciones agregadas** (SUM, COUNT, AVG, etc.) con GROUP BY.
- 4.Una consulta creativa con **CROSS JOIN** (opcional, si aplica).
- 5.Una visualización de los resultados de alguna de las consultas (puede ser una tabla o gráfico generado con algún software o script).

Datasets recomendados en Kaggle

- IMDb Movies Dataset
- Sales Database
- Northwind Database
- Chinook Database

Entregables

Los estudiantes deben entregar:

1. Informe PDF o presentación, que contenga:

- Breve descripción del dataset elegido.
- Diagrama de las tablas utilizadas (puede ser dibujado o generado).
- Enunciado de las preguntas que se plantearon.
- Consultas SQL ejecutadas con su resultado resumido.
- Visualización opcional de resultados (por ejemplo, gráfica de total de ventas por país).
- Conclusiones o hallazgos obtenidos.

2. Archivo .sql con todas las consultas escritas y comentadas.

3. (Opcional) Script o notebook (Python, R, etc.) si decidieron hacer consultas desde código o conectarse a SQLite/MySQL para visualizar.

Recursos de apoyo

- [Quick Ref](#)

Un futuro con historias

Referencias

- C. M. Ricardo, *Bases de datos*. México: McGraw Hill/Interamericana Editores, 2010.
- L. F. Wanumen Silva, E. Rivas Trujillo, y D. J. Mosquera Palacios, *Bases de datos en SQL server*, Primera edición. Bogotá, Colombia: Ecoe Ediciones, 2018.
- T. Boronczyk, M. Glass, y E. Naramore, *Desarrollo web con PHP 6, Apache y MySQL*. Madrid: Anaya Multimedia, 2010.
- M. L. Gillenson, *Fundamentals of database management systems*, Internat. ed. Hoboken, NJ: Wiley, 2005.