



Recibe una cálida:

# ¡Bienvenida!

---

Te estábamos esperando 😊 +

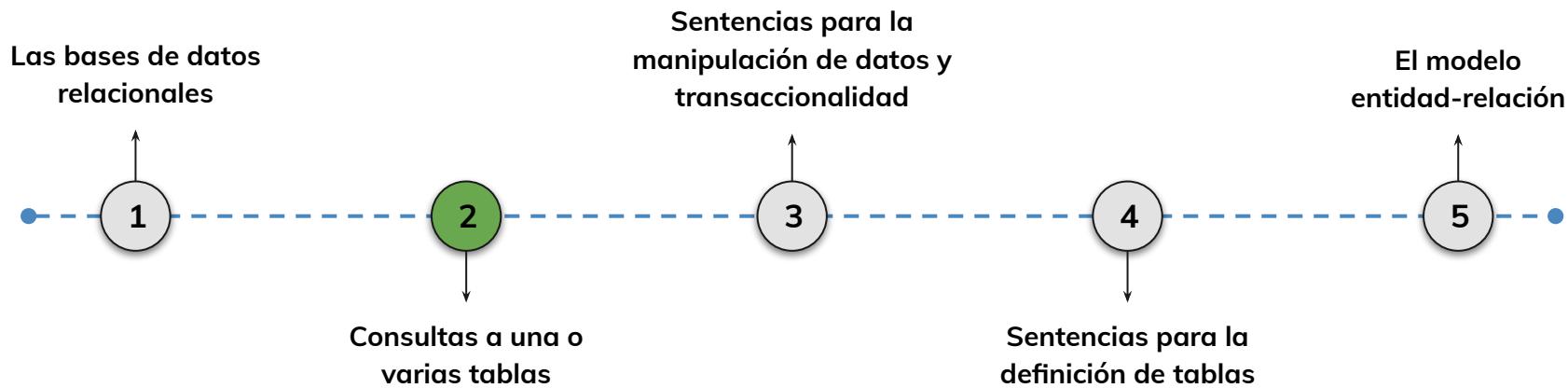


# › Consultas a una o varias tablas - Parte 1

**AE2:** Utilizar lenguaje estructurado de consultas sql para la obtención de información que satisface los requerimientos planteados a partir de un modelo de datos dado.

# Roadmap de lecciones

¿Cuáles **lecciones** estaremos estudiando en este módulo?



# Learning Path

¿Cuáles temas trabajaremos hoy?

**AE2.1**

## Fundamentos de bases de datos relacionales

El objetivo de este tema es brindar una descripción básica de los conceptos asociados a las bases de datos relacionales que te permitirán desde tu ordenador instalar las herramientas necesarias para establecer una conexión a BD y comenzar a manipular dichos datos.

Lenguaje Estructurado de Consultas SQL

Recuperar información de una tabla utilizando consultas SQL

¿Qué es un modelo de datos?

# Objetivos de aprendizaje

¿Qué aprenderás?

- Definir el Lenguaje Estructurado de Consultas SQL
- Conocer los diferentes tipos de cláusulas SQL para recuperar datos de una tabla
- Recuperar información de una tabla utilizando consultas SQL
- Conocer el concepto de Modelo de Datos

# Repaso clase anterior

¿Quedó alguna duda?

En la clase anterior trabajamos :

- Qué es un RDBMS
- Los principales objetos de una base de datos

# #Momentode Preguntas...



¿Qué es SQL?



¿Qué es un modelo de datos?



¿Qué es una clave primaria?



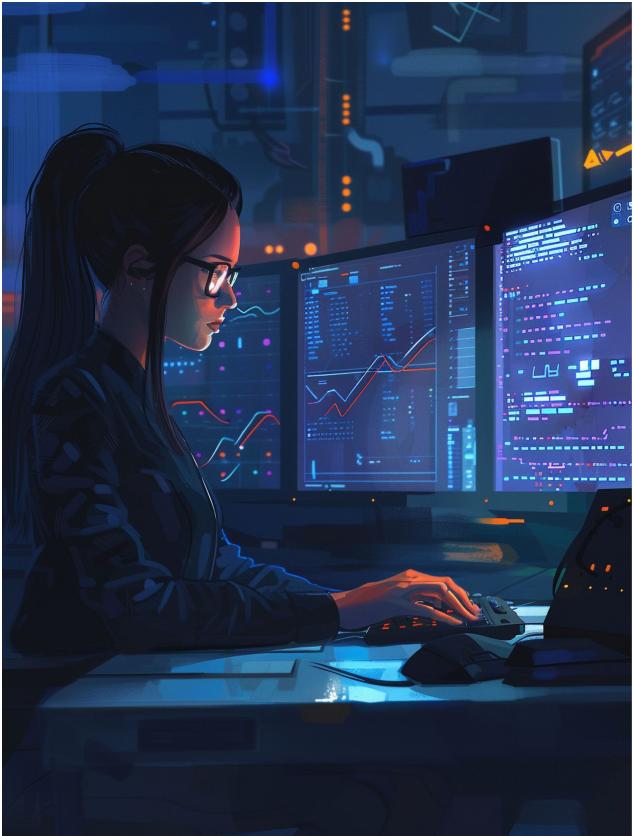
¿Qué es una clave externa en SQL?

# El Lenguaje Estructurado de Consultas (SQL)

# El Lenguaje Estructurado de Consultas (SQL)

## ¿Qué es SQL?

El Lenguaje Estructurado de Consultas, conocido comúnmente como SQL (por sus siglas en inglés, Structured Query Language), es un lenguaje de programación utilizado para gestionar y manipular bases de datos relacionales. Fue desarrollado originalmente en la década de 1970 por IBM, y desde entonces ha sido adoptado como un estándar en la industria de la gestión de bases de datos.



# El Lenguaje Estructurado de Consultas (SQL)

SQL se utiliza para realizar diversas operaciones en bases de datos, como la creación, modificación y eliminación de tablas y registros, así como para recuperar y manipular datos almacenados en esas tablas. Algunas de las principales tareas que se pueden realizar con SQL incluyen:



## Manipulación de transacciones

Gestionar transacciones en sistemas de ventas, operaciones bancarias o cualquier proceso crítico que necesite ser tratado de manera atómica.

# El Lenguaje Estructurado de Consultas (SQL)



## Consulta de datos

Realizar consultas para reportes, crear dashboards de métricas, verificar datos de clientes, obtener estadísticas de ventas, entre otros.



## Inserción de datos

Agregar clientes a una base de datos CRM, registrar ventas, agregar entradas en un sistema de soporte, entre otros.



## Actualización de datos

Actualizar el estado de una orden de compra, modificar la dirección de un cliente, corregir un registro de ventas, entre otros.

# El Lenguaje Estructurado de Consultas (SQL)



## Eliminación de datos

Eliminar registros de clientes inactivos, borrar datos antiguos de transacciones, eliminar productos descontinuados, entre otros.



## Definición de Esquemas

Definir nuevas tablas, relaciones entre tablas, reglas de negocio en la base de datos, y definir el esquema completo de una aplicación.



## Gestión de permisos

Asignar roles y permisos a los usuarios de la base de datos, gestionar la seguridad para evitar accesos no autorizados, crear reglas de acceso a datos confidenciales.

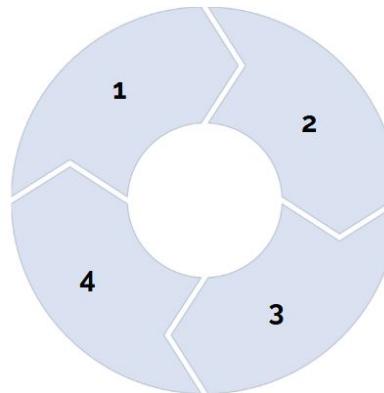
# El Lenguaje Estructurado de Consultas (SQL)

Las sentencias SQL se agrupan en diferentes sub-sets, de acuerdo a su propósito. Los principales son los siguientes.

1. DDL (Data Definition Language)

2. DML (Data Manipulation Language)

4. TCL (Transaction Control Language o TCL)



3. DCL (Data Control Language)

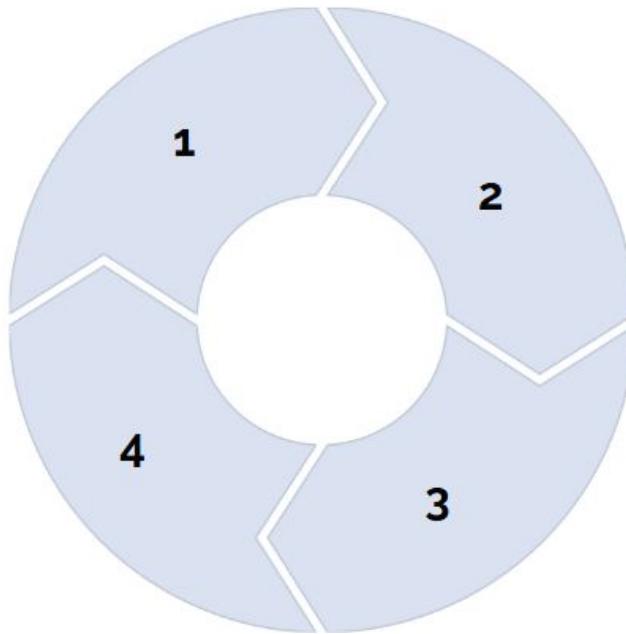
# El Lenguaje Estructurado de Consultas (SQL)

## 1. DDL (Data Definition Language o DDL)

Es el conjunto de sentencias de SQL que se usa para crear y modificar objetos de base de datos.

## 4. TCL (Transaction Control Language o TCL)

Es el conjunto de sentencias SQL que se utilizan para gestionar transacciones, permitiendo confirmar o deshacer cambios en la base de datos.



## 2. DML (Data Manipulation Language o DML)

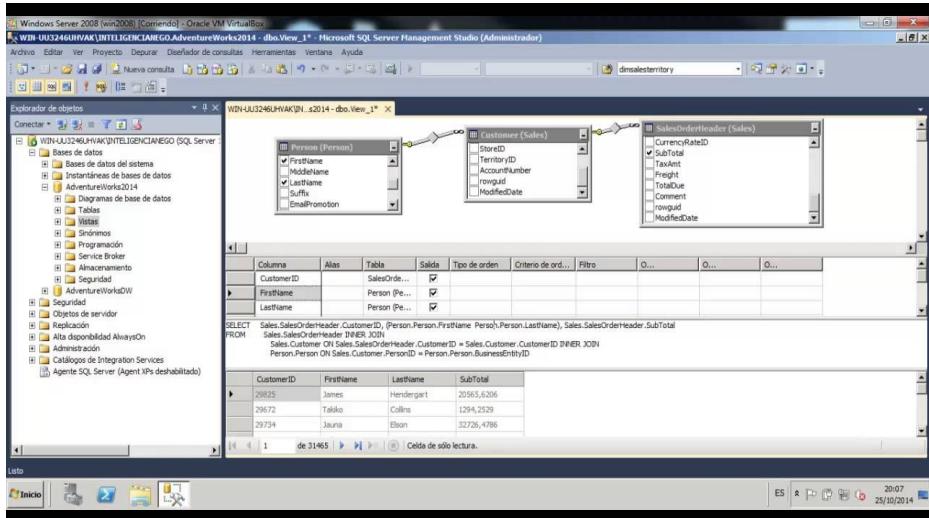
Es el conjunto de sentencias SQL que se utilizan para consultar los datos dentro de una base de datos.

## 3. DCL (Data Control Language o DCL)

Es el conjunto de sentencias SQL que permiten a un administrador de bases de datos, gestionar y controlar el acceso a los objetos de la misma.

# El Lenguaje Estructurado de Consultas (SQL)

SQL no se usa solamente para la manipulación y consulta de datos, sino que también puede utilizarse para crear y modificar objetos de una base de datos, como lo son tablas y vistas.



Fuente de imagen: [Creando vistas en SQL SERVER 2014](#)

# Recuperar información de una tabla

## ¿Qué significa recuperar datos de una tabla utilizando SQL?

Se refiere al proceso de solicitar y obtener información específica almacenada en una base de datos relacional mediante la ejecución de una consulta SQL.

En otras palabras, se trata de obtener registros o filas de datos de una tabla, que cumplan con ciertos criterios establecidos en la consulta.

La capacidad de recuperar datos de manera selectiva y personalizada es una de las características poderosas de SQL y es fundamental para aprovechar al máximo una base de datos relacional.



# Recuperar información de una tabla

Para recuperar datos de una tabla utilizando SQL, generalmente se sigue el siguiente proceso:

- 1. Escribir una consulta SQL:** Debes redactar una consulta SQL que especifique qué datos deseas recuperar y cómo deseas que se filtren o se presenten. La consulta se redacta en el lenguaje SQL y se envía al sistema de gestión de bases de datos (DBMS) para su procesamiento.
- 2. Usar la cláusula SELECT:** La cláusula SELECT es fundamental para recuperar datos de una tabla. Define las columnas específicas que deseas recuperar de la tabla. Puedes seleccionar una o varias columnas, o incluso todas las columnas en la tabla.
- 3. Especificar la tabla:** Debes indicar la tabla de la cual deseas recuperar los datos. Esto se hace usando la cláusula FROM, seguida del nombre de la tabla.



# Recuperar información de una tabla

4. **Aplicar condiciones de filtro:** Si deseas recuperar solo ciertos registros que cumplan con ciertas condiciones, puedes usar la cláusula **WHERE**. Esto permite filtrar los registros basados en los valores en ciertas columnas.
5. **Ordenar los resultados:** Si deseas que los resultados se muestren en un orden específico, puedes utilizar la cláusula ORDER BY. Esto te permite ordenar los resultados en función de los valores en una o más columnas, ya sea de forma ascendente o descendente.

# Recuperar información de una tabla

**¿Cuáles son los diferentes tipos de cláusulas SQL para recuperar información de la tabla?**

SQL ofrece varias cláusulas que permiten recuperar información de una tabla de manera específica y personalizada.

**Estas cláusulas se utilizan en combinación con la instrucción SELECT** para definir las condiciones, el orden y otros aspectos de la recuperación de datos.

**SELECT**: Esta es la instrucción básica para recuperar datos de una tabla. Junto con ella, se especifican las columnas que se desean recuperar de la tabla.

**FROM**: Esta cláusula especifica la tabla de la cual se van a recuperar los datos. Es esencial en todas las consultas SQL.

**WHERE**: Se utiliza para filtrar los registros basados en una o varias condiciones. Solo se recuperarán los registros que cumplan con las condiciones especificadas en esta cláusula.

# Recuperar información de una tabla

**ORDER BY:** Permite ordenar los resultados en función de una o más columnas, ya sea en orden ascendente (ASC) o descendente (DESC).

```
sql
SELECT nombre, edad
FROM usuarios
WHERE ciudad = 'Nueva York'
ORDER BY edad DESC;
```

**LIMIT** (o TOP en algunas bases de datos): Limita el número de filas recuperadas en el resultado. Puede ser útil para mostrar solo una cantidad específica de registros.

**EXISTS:** Se utiliza para verificar la existencia de registros en una subconsulta. Puede ser útil para verificar si se cumple una condición en otra tabla antes de incluir un registro en el resultado.

# Utilización de funciones en las consultas

Las funciones permiten modificar o analizar los valores de las columnas dentro de una consulta SQL. Pueden aplicarse sobre texto, números o fechas y devuelven un valor transformado sin alterar los datos originales en la tabla.

- ◆ Ejemplos comunes:

- UPPER() → convierte texto a mayúsculas.
- LOWER() → convierte texto a minúsculas.
- LENGTH() → devuelve la cantidad de caracteres.
- ROUND() → redondea valores numéricos.

```
SELECT  
    UPPER(nombre) AS NombreMayuscula,  
    ROUND(salario, 2) AS SalarioRedondeado  
FROM Empleados;
```

💡 Estas funciones se aplican dentro del SELECT para mostrar datos calculados o transformados al momento de la consulta

# Consultas de selección con funciones de agregación

Las funciones de agregación permiten resumir información de una tabla, operando sobre conjuntos de filas en lugar de datos individuales.

- ◆ Principales funciones:

- COUNT() → cuenta registros.
- SUM() → suma valores.
- AVG() → calcula promedios.
- MIN() / MAX() → obtiene valores mínimos o máximos.

```
SELECT  
    departamento,  
    COUNT(*) AS TotalEmpleados,  
    AVG(salario) AS PromedioSalarial  
FROM Empleados  
GROUP BY departamento;
```

💡 Las funciones de agregación se combinan con GROUP BY para obtener resultados resumidos por categoría o grupo.

# LIVE CODING

## Introducción al caso de estudio

**Título:** Caso de estudio: Tienda en línea

- Dataset: <https://www.kaggle.com/datasets/carrie1/ecommerce-data>

## Contenido:

- Tablas principales: **Clientes, Productos, Órdenes, DetalleOrden**
- Relaciones:
  - Órdenes.id\_cliente → Clientes.id\_cliente
  - DetalleOrden.id\_orden → Órdenes.id\_orden
  - DetalleOrden.id\_producto → Productos.id\_producto
- Objetivo: aplicar consultas SQL a un modelo realista



# LIVE CODING

Realizar las siguientes tareas (parte 2):

1. Recupera todos los registros de la tabla Clientes.
2. Recupera el nombre y precio de los productos con precio mayor a 100.
3. Recupera los nombres de los clientes que realizaron órdenes en el año 2024.
4. Recupera el producto más caro de la tabla Productos.
5. Recupera el número total de órdenes realizadas en la base.
6. Recupera el nombre del producto y el total vendido de la categoría "Electrónica", ordenados de mayor a menor venta.

Tiempo: 45 minutos



# ¿Qué es un modelo de datos?



## **Algunos de los modelos de datos más comunes**

Un modelo de datos en SQL se refiere a la estructura y organización de los datos en una base de datos. Define cómo se almacenan, acceden y relacionan los datos dentro del sistema de gestión de bases de datos.

Un modelo de datos especifica la manera en que se crean las tablas, las columnas, las relaciones y las restricciones en la base de datos.

Existen varios tipos de modelos de datos en SQL, y cada uno de ellos define cómo se representan los datos y cómo se relacionan entre sí.

En la próxima diapositiva veremos algunos de los modelos de datos más comunes.

# ¿Qué es un modelo de datos?

Algunos de los modelos de datos más comunes:

**Modelo Relacional:** Es el modelo más utilizado en SQL. Los datos se almacenan en tablas con filas y columnas, y las relaciones entre los datos se establecen mediante claves primarias y foráneas.

**Modelo de Entidad-Relación (ER):** Este modelo utiliza entidades para representar objetos del mundo real y relaciones para conectar esas entidades.

**Modelo Orientado a Documentos:** Almacena datos en documentos en formatos como JSON o XML, permitiendo una mayor flexibilidad en la estructura de los datos.

## ¡Recapitulemos!

En SQL, el modelo de datos se utiliza para definir la estructura de la base de datos, incluyendo la creación de tablas, la especificación de tipos de datos, las claves primarias y foráneas, las restricciones de integridad y las relaciones entre tablas.



Momento:

# Time-out!

⌚ 5 - 10 min.



# Clave primaria y clave externa

# Clave primaria y clave externa

## ¿Qué es una clave primaria?

En SQL, una llave primaria (primary key en inglés) es un campo o conjunto de campos en una tabla que sirve para identificar de manera única cada registro dentro de esa tabla.

La llave primaria garantiza que no puede haber duplicados en este campo o conjunto de campos, lo que ayuda a mantener la integridad y la consistencia de los datos en la base de datos.

La función principal de una llave primaria es proporcionar un método eficiente para buscar y acceder a registros específicos dentro de una tabla.



# ¿Qué es un modelo de datos?

Algunos aspectos importantes sobre las llaves primarias en SQL son:

**Unicidad:** Cada valor en la columna de la llave primaria debe ser único en toda la tabla.

**No nulidad:** Los valores en la columna de la llave primaria no pueden ser nulos (NULL).

**Identificación:** La llave primaria se utiliza para identificar de manera única cada registro en la tabla

Ejemplo de declaración de una llave primaria en SQL:

```
sql
CREATE TABLE Empleados (
    ID INT PRIMARY KEY,
    Nombre VARCHAR(50),
    Departamento VARCHAR(50)
);
```

# Clave primaria (PK) y clave externa (FK)

## ¿Qué es una clave externa en SQL?:

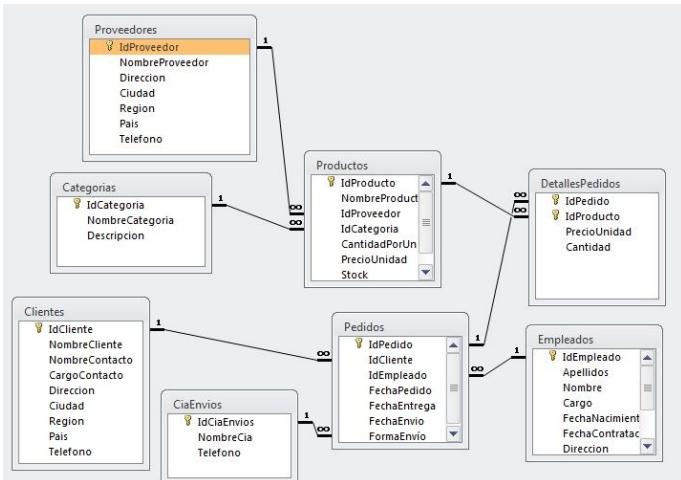
Una clave externa (foreign key en inglés) es un campo o conjunto de campos en una tabla que se utiliza para establecer una relación entre dos tablas.

Representa una referencia a la clave primaria de otra tabla, creando una relación entre las dos tablas.



# Clave primaria y clave externa

Las claves primarias y claves foráneas permiten representar relaciones entre los datos y mantener la integridad referencial en la base de datos.



## ¿Para qué pueden usarse las claves foráneas?:

Las claves foráneas pueden utilizarse para realizar uniones entre tablas, recuperar datos relacionados y aplicar restricciones de integridad de datos.

Son un concepto fundamental en el diseño de bases de datos y se utilizan ampliamente en bases de datos relacionales.

# Consultas SQL utilizando claves primarias y foráneas

# Consultas SQL utilizando claves primarias y foránea

## Repasemos:

- ★ **Clave Primaria (Primary Key):** Una clave primaria es un atributo o conjunto de atributos en una tabla que identifica de manera única cada registro en esa tabla.
- ★ **Clave Foránea (Foreign Key):** Una clave foránea es un atributo o conjunto de atributos en una tabla que establece una relación entre esa tabla y otra tabla. La clave foránea suele ser una referencia a la clave primaria de la tabla relacionada.



# Consultas SQL utilizando claves primarias y foránea

## ★ **Introducción:**

- ★ Hacer consultas SQL utilizando claves primarias y foráneas implica aprovechar las **relaciones entre tablas** en una base de datos relacional para recuperar información específica de manera más precisa y coherente.
- ★ Tanto las claves primarias como las foráneas son elementos esenciales en el diseño de bases de datos y juegan un papel crucial en la organización de los datos y la definición de las relaciones entre las tablas.
- ★ **¡Veamos un ejemplo en la siguiente diapositiva!**



# Consultas SQL utilizando claves primarias y foráneas

Supongamos que tienes dos tablas en una base de datos: Una tabla de "Clientes" y una tabla de "Órdenes". La tabla de "Órdenes" tiene una columna llamada "id\_cliente" que es una clave foránea que se relaciona con la clave primaria "id" en la tabla de "Clientes".

```
-- Consulta para obtener información de un cliente específico y sus órdenes
SELECT *
FROM Clientes
WHERE id = 123;

-- Consulta para obtener las órdenes realizadas por un cliente específico
SELECT *
FROM Órdenes
WHERE id_cliente = 123;
```

# Clave primaria y clave externa

## Analicemos el ejemplo anterior:

En el primer ejemplo, estamos utilizando la clave primaria para obtener información detallada sobre un cliente específico.

En el segundo ejemplo, estamos utilizando la clave foránea para recuperar las órdenes realizadas por ese mismo cliente.

## Recapitulaciones:

Las claves primarias y foráneas son fundamentales para la efectividad de las consultas SQL eficientes y para garantizar la integridad de los datos en bases de datos relacionales.

Permiten establecer conexiones entre los datos almacenados en diferentes tablas y facilitan la recuperación y manipulación coherente de la información.



# LIVE CODING

## Escribir consultas SQL para obtener la siguiente información:

Tienes una base de datos que registra información de clientes, productos y órdenes en una **tienda en línea**, con las siguientes tablas principales:

- **Clientes:** id\_cliente, nombre, email, país
- **Productos:** id\_producto, nombre, categoría, precio
- **Órdenes:** id\_orden, fecha, id\_cliente
- **DetalleOrden:** id\_orden, id\_producto, cantidad

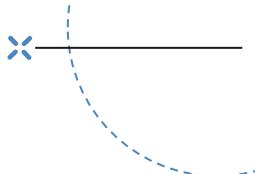
Consignas:

- Obtener el nombre y email de todos los clientes registrados.
- Obtener el nombre y categoría de los productos con precio mayor a 100.
- Obtener el nombre de los clientes y el ID de las órdenes que realizaron en el año 2024.



Ejercicio N° 1

# Consultas SQL con Tablas Relacionadas



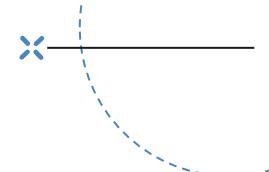
# Consultas SQL con Tablas Relacionadas

**Contexto:** 🙋

Estás administrando una base de datos para una tienda en línea. La base contiene cuatro tablas principales:

- Clientes: id\_cliente (PK), nombre, email, país
- Productos: id\_producto (PK), nombre, categoría, precio
- Órdenes: id\_orden (PK), fecha, id\_cliente (FK)
- DetalleOrden: id\_orden (FK), id\_producto (FK), cantidad

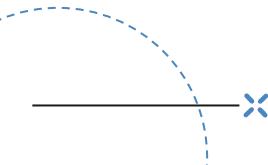
# Consultas SQL con Tablas Relacionadas



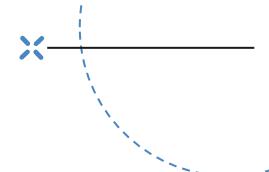
**Consigna:** 📝

1. Recuperar el nombre y país de los clientes que hayan realizado al menos una orden en 2024.
2. Obtener los nombres de los productos y la suma total vendida (cantidad × precio).
3. Listar el nombre del cliente y el monto total comprado, ordenados de mayor a menor.
4. Obtener el nombre del producto más caro de la categoría "Electrónica".
5. Contar cuántos clientes distintos hicieron compras en cada país.

**Tiempo** ⏰: 45 minutos

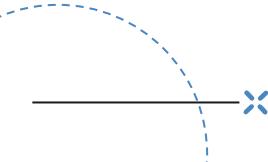


# Consultas SQL con Tablas Relacionadas



Paso a paso: 

1. Define las relaciones entre las tablas con las claves primaria y foránea.
2. Escribe las consultas paso a paso, utilizando las cláusulas **SELECT, FROM, JOIN, WHERE, GROUP BY** y **ORDER BY**.
3. Asegúrate de probar las consultas en un entorno SQL para verificar los resultados.
4. Reflexiona sobre el uso de las claves primarias y foráneas para realizar uniones entre tablas.





# ¿Alguna consulta?



# Resumen

¿Qué logramos en esta clase?

- ✓ Definir el concepto SQL
- ✓ Recuperar información de una tabla utilizando SQL
- ✓ Diferenciar los distintos tipos de cláusulas SQL que se utilizan para recuperar datos de una tabla
- ✓ Comprender qué es un modelo de datos
- ✓ Aprender la definición de una llave primaria
- ✓ Aprender la definición de una llave llave foránea/externa
- ✓ Realizar consultas SQL utilizando claves primarias y foráneas



# ¡Ponte a prueba!

Momento de ejercitación

Te invitamos a aprovechar esta última sección del espacio sincrónico para realizar de manera individual las **actividades disponibles en la plataforma**. Estas propuestas son clave para afianzar lo trabajado y **forman parte obligatoria del recorrido de aprendizaje**.

-  [Análisis de caso](#) —————  [Selección Múltiple](#)
-  [Comprensión lectora](#)

Si al resolverlas surge alguna duda, compártela o tráela al próximo encuentro sincrónico.



# #Checkout

¿Qué les pareció la clase de hoy?



< ¡Muchas gracias! >

