



Recibe una cálida:

# ¡Bienvenida!

---

Te estábamos esperando 😊 +



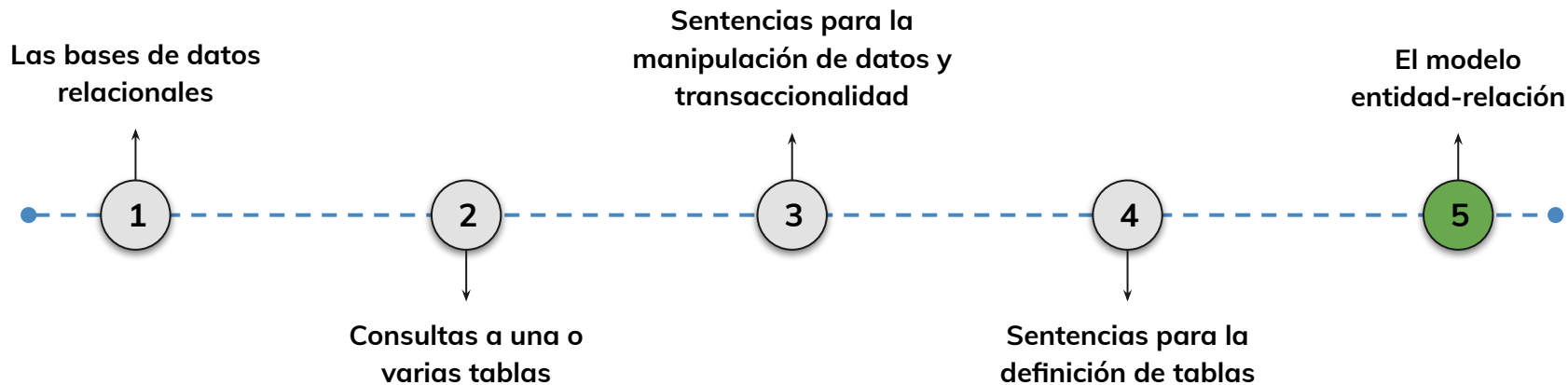
# › El modelo entidad-relación

## - Parte 1

**AE5:** Elaborar un modelo de datos de acuerdo a los estándares de modelamiento para resolver un problema de baja complejidad.

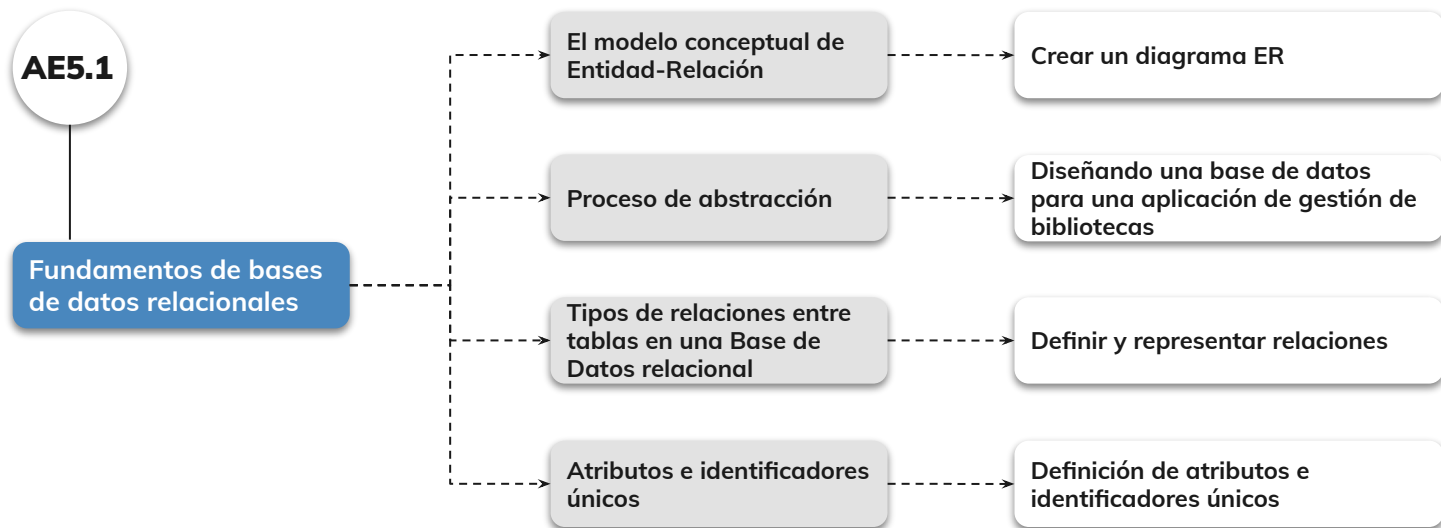
# Roadmap de lecciones

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



# Learning Path

¿Cuáles temas trabajaremos hoy?



# Objetivos de aprendizaje

¿Qué aprenderás?

- Conocer el proceso de abstracción en bases de datos
- Conocer el modelo conceptual de Entidad-Relación
- Crear diagramas de flujo ER
- Definir los tipos de relaciones entre tablas en una base de datos
- Diferenciar los tipos de relaciones entre tablas
- Entender qué es un atributo
- Aprender qué es un identificador único

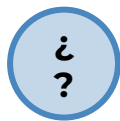
# Repaso clase anterior

¿Quedó alguna duda?

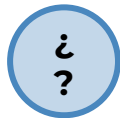
En la clase anterior trabajamos :

- Modificar el nombre de una columna en una tabla
- Modificar el tipo de dato de un campo en una tabla
- Modificar una restricción de nulidad en una tabla
- Eliminar una tabla utilizando la sentencia `DROP TABLE`.
- Truncar una tabla utilizando la sentencia `TRUNCATE`.

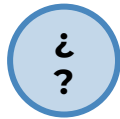
# #Momentode Preguntas...



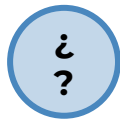
*¿Qué es el modelo conceptual Entidad-Relación?*



*¿Qué es el proceso de abstracción?*



*¿Qué Tipos de relaciones existen entre tablas en una Base de Datos relacional?*



*¿Qué es un atributo en bases de datos?*

# El modelo conceptual de Entidad-Relación





# El modelo conceptual de Entidad-Relación

## **¿Qué es el modelo conceptual Entidad-Relación?**

El modelo conceptual de Entidad-Relación (ER) es un enfoque utilizado en el diseño de bases de datos para representar y describir de manera abstracta y visual la estructura de datos y las relaciones entre entidades en un sistema de información.

El modelo ER proporciona una representación gráfica que facilita la comunicación entre los diseñadores de bases de datos y los usuarios finales.

# El modelo conceptual de Entidad-Relación

**El modelo ER se basa en tres componentes principales:**

**Entidades** : Las entidades representan objetos o conceptos del mundo real que son significativos para el sistema que se está modelando.

*Por ejemplo, en una base de datos de una biblioteca, las entidades podrían incluir "Libro", "Autor", "Lector" y "Préstamo". Cada entidad se representa en el diagrama ER como un rectángulo con un nombre descriptivo.*



# El modelo conceptual de Entidad-Relación

## El modelo ER se basa en tres componentes principales:

1. **Atributos** : Los atributos son las propiedades o características de las entidades que se desean registrar en la base de datos.
  - Por ejemplo, para la entidad "Libro", los atributos podrían ser "Título", "ISBN", "Año de Publicación" y "Editorial". Los atributos se representan como elipses u óvalos y se conectan a la entidad correspondiente.

# El modelo conceptual de Entidad-Relación

2. **Relaciones** : Las relaciones representan las asociaciones entre las entidades. Estas asociaciones pueden ser uno a uno, uno a muchos o muchos a muchos, y reflejan cómo las entidades se relacionan entre sí en el sistema.

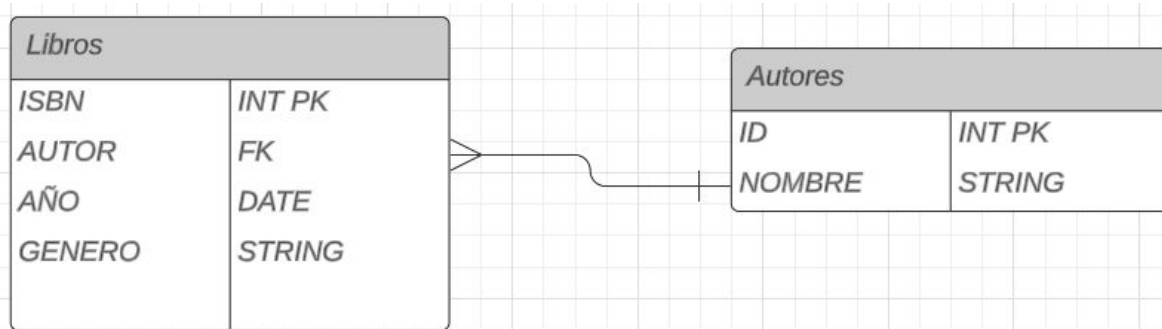
3. Las **relaciones** se representan como líneas que conectan las entidades participantes y se etiquetan con verbos descriptivos, como "posee", "autor de" o "prestado a".

# El modelo conceptual de Entidad-Relación

## Diagramas y ER:

El modelo ER utiliza diagramas ER para visualizar estas entidades, atributos y relaciones. Estos diagramas ayudan a los diseñadores de bases de datos a comprender la estructura de datos, identificar las restricciones y requerimientos del sistema, y comunicarse eficazmente con los usuarios y otros miembros del equipo de desarrollo.

## Ejemplo de un diagrama simple:



# LIVE CODING

Ejemplo en vivo

## Crear un diagrama ER | Parte 1

El objetivo es crear un diagrama ER que refleje las relaciones entre estas entidades: Cursos, Estudiantes y Profesores.

- **Estudiante:** ID, nombre, fecha de nacimiento y dirección.
- **Profesor:** ID, nombre y título.
- **Curso:** ID, nombre y descripción

**Tiempo:** 20 minutos



# LIVE CODING

Ejemplo en vivo

## Crear un diagrama ER| Parte 2:

### Relaciones

- **Inscripción:** Los estudiantes se inscriben en cursos. Cada inscripción se registra con una fecha de inscripción.
- **Enseñanza:** Los profesores enseñan cursos. Cada enseñanza se registra con una fecha de inicio.

**Tiempo:** 15 minutos

# LIVE CODING

Ejemplo en vivo

## Crear un diagrama ER | Parte 3:

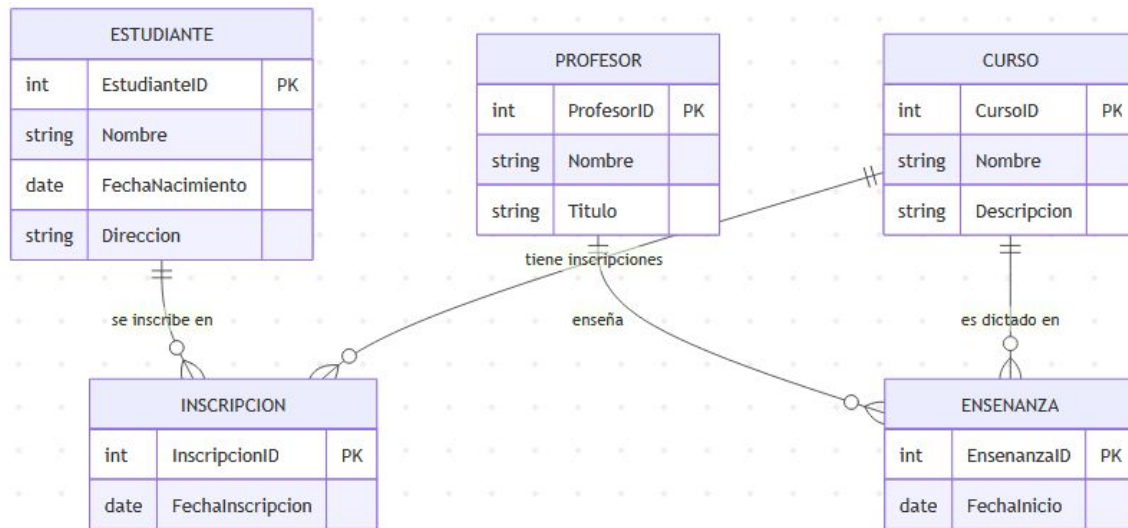
1. Dibujar un diagrama entidad-relación (ER) que incluya las tres entidades mencionadas anteriormente (Estudiante, Profesor y Curso).
2. Agregar las relaciones "Inscripción" y "Enseñanza" entre las entidades apropiadas. Identificar los atributos clave en cada entidad.
3. Agregar atributos adicionales según lo mencionado en la descripción de las entidades (ej: fecha de nacimiento, título, etc.).

**Tiempo:** 15 minutos



# LIVE CODING

Recursos mentor



# Proceso de abstracción



# Proceso de abstracción



## **¿Qué es el proceso de abstracción?:**

El proceso de abstracción de datos en bases de datos relacionales se refiere a la idea de ocultar ciertos detalles de la estructura y organización de los datos, a fin de facilitar la interacción con la base de datos y garantizar la integridad de los datos.

Esto se logra a través de varios conceptos y técnicas en las bases de datos relacionales, como tablas, vistas, esquemas y consultas.



# Proceso de abstracción

**Aquí tienes una descripción más detallada de cómo funciona este proceso:**

**Tablas :** En una base de datos relacional, los datos se almacenan en tablas. Cada tabla está compuesta por filas y columnas. Cada fila representa una entidad o registro único, mientras que las columnas representan los atributos o características de esa entidad.

**Esquemas:** Los esquemas en una base de datos definen la estructura de las tablas, especificando qué columnas contiene cada tabla, así como los tipos de datos y las restricciones que se aplican a esas columnas.

# Proceso de abstracción

**Consultas** : Los usuarios interactúan con la base de datos a través de consultas SQL (Structured Query Language). Estas consultas permiten a los usuarios recuperar, modificar o eliminar datos de la base de datos sin necesidad de conocer la ubicación física de los datos o cómo se almacenan internamente.

**Vistas** : Las vistas son consultas predefinidas que ofrecen una abstracción adicional al ocultar ciertos detalles de las tablas subyacentes. Las vistas pueden unir múltiples tablas, seleccionar columnas específicas o aplicar filtros, lo que permite a los usuarios acceder a los datos de manera más conveniente sin necesidad de comprender la estructura completa de la base de datos.



Ejercicio N° 1

# **Diseñando una base de datos para una aplicación de gestión de bibliotecas**

# Diseñando una base de datos para una aplicación de gestión de bibliotecas

## Contexto: 🙌

Tu tarea es aplicar el proceso de abstracción de datos para crear un modelo ER (Entidad-Relación) que capture información sobre libros, autores y préstamos de libros a los miembros de la biblioteca de manera abstracta y clara.

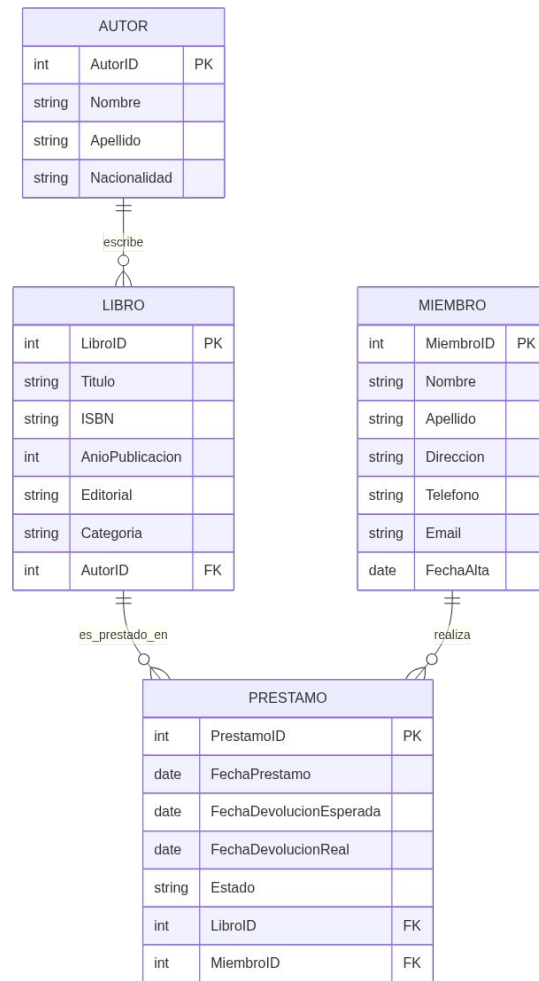
## Paso a paso: ⚙️

1. Identificar las entidades y atributos claves.
2. Establecer relaciones entre las entidades.
3. Generar un diagrama ER que represente estas relaciones.

Tiempo 🕒: 15 minutos

# Diseñando una base de datos para una aplicación de gestión de bibliotecas

Recursos para el mentor







Momento:

# Time-out!



5 -10 min.



# Tipos de relaciones entre tablas en una Base de Datos relacional



# Tipos de relaciones

*Tablas en una Base de Datos relacional*



## **Relaciones entre tablas:**

En el modelo entidad-relación (ER) en bases de datos, existen varios tipos de relaciones que pueden definirse entre las entidades para representar cómo se relacionan los datos. Cada tipo de relación tiene implicaciones en la estructura de la base de datos y cómo se gestionan los datos.

## **Los tipos de relaciones más comunes son:**

Relación Uno a Uno, relación Uno a Muchos, relación Muchos a uno, relación Muchos a Muchos

# Tipos de relaciones

Tablas en una Base de Datos relacional

## Relación Uno a Uno (One-To-One)

En esta relación, una entidad A está relacionada con una única entidad B, y viceversa.

**Por ejemplo,** si tienes una entidad "Persona" y otra entidad "Pasaporte", una persona tiene un único pasaporte y un pasaporte está vinculado a una sola persona.



# Tipos de relaciones

*Tablas en una Base de Datos relacional*

## **Relación Uno a Muchos (One-to-Many):**

En esta relación, una entidad A está relacionada con muchas entidades B, pero una entidad B solo está relacionada con una entidad A.



**Por ejemplo,** en una base de datos de estudiantes y clases, un estudiante puede estar inscrito en varias clases, pero cada clase tiene varios estudiantes inscritos.

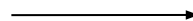


# Tipos de relaciones

Tablas en una Base de Datos relacional

## Relación Muchos a Uno (Many-To-One)

Es el opuesto de la relación uno a muchos. Una entidad A está relacionada con una sola entidad B, pero muchas entidades B pueden estar relacionadas con la entidad A.



**Por ejemplo,** en una base de datos de pedidos y clientes, cada pedido está relacionado con un solo cliente, pero cada cliente puede tener varios pedidos.

# Tipos de relaciones

Tablas en una Base de Datos relacional

## Relación Muchos a Muchos (Many-To-Many)

En esta relación, muchas entidades A pueden estar relacionadas con muchas entidades B.

**Por ejemplo,** en una base de datos de estudiantes y asignaturas, varios estudiantes pueden estar inscritos en varias asignaturas.

### En resumen:

Estos son los tipos de relaciones más comunes en el modelo entidad-relación. La elección del tipo de relación depende de cómo los datos se relacionen en el contexto específico de la base de datos que estás diseñando



Ejercicio N° 2

# Definir y representar relaciones



# Diseñando una base de datos para una aplicación de gestión de bibliotecas

## Contexto: 🙌

Supongamos que estás diseñando una base de datos para una aplicación de gestión de proyectos y necesitas modelar las relaciones entre empleados, proyectos y tareas dentro de esos proyectos.

## Consigna: 📝

Tu tarea es definir y representar estas relaciones especificadas anteriormente

**Tiempo** 🕒: 15 minutos

## Paso a paso: ⚙️

1. Definir de Entidades y Atributos claves
2. Establecer relaciones entre las entidades (por ej: one-to-many, many-to-one, etc) según corresponda
3. Dibujar un diagrama entidad-relación (ER) que represente visualmente las entidades, atributos y relaciones que has identificado.

# Atributos e identificadores únicos



# Atributos e identificadores únicos



## ¿Qué es un atributo en bases de datos?

En una base de datos, un atributo es una característica o propiedad que describe una entidad o un objeto dentro de un conjunto de datos. Los atributos son elementos básicos que permiten almacenar información específica sobre los elementos que se están registrando en la base de datos



# Atributos e identificadores únicos



Algunos puntos clave sobre los atributos en bases de datos:

## **Descripción de Datos:**

Los atributos se utilizan para describir los datos en una base de datos.

Cada entidad en la base de datos tendrá atributos que capturan información relevante sobre ese elemento.

## **Tipos de Datos:**

Los atributos pueden tener diferentes tipos de datos, como texto, números, fechas, booleanos, etc. Por ejemplo, un atributo "Nombre" de una entidad "Cliente" puede contener datos de tipo texto para almacenar el nombre del cliente.

# Atributos e identificadores únicos

Algunos puntos clave sobre los atributos en bases de datos:

## **Identificación Única:**

En muchos casos, un atributo se utiliza como clave primaria para identificar de manera única cada registro en una tabla.

Por ejemplo, en una tabla de "Empleados", el atributo "ID de Empleado" puede ser una clave primaria única para cada empleado.

**Un identificador único,** también conocido como *clave primaria*, es un atributo especial o una combinación de atributos dentro de una tabla de base de datos que identifica de forma única cada registro (fila) de esa tabla.

# LIVE CODING

Ejemplo en vivo

## Definición de atributos e identificadores únicos:

Tu tarea es definir los atributos y los identificadores únicos para las entidades "Producto" y "Pedido".

1. Definir los atributos
2. Definir los identificadores únicos
3. Realiza una consulta para obtener el nombre de productos sin stock

**Tiempo:** 15 minutos



Ejercicio N° 3

# Diseñando la base de datos de una tienda online

# Diseñando una base de datos para una aplicación de gestión de bibliotecas

**Contexto:** 🙌

Imagina que estás creando el backend de una plataforma de comercio electrónico.

Necesitas modelar los datos de Usuarios, Productos, Órdenes y Pagos, de modo que el sistema permita: gestionar inventario, registrar compras con múltiples productos y llevar el historial de cada transacción.

**Consigna:** ✍️

Tu tarea es definir y representar las entidades clave y sus relaciones, asegurando que cada registro sea único y consistente.

**Tiempo** 🕒: 15 minutos

**Paso a paso:** ⚙️

1. Identifica Entidades y Atributos esenciales  
Usuario (nombre, email, contraseña, dirección, ...)  
Producto (SKU, nombre, descripción, precio, stock, ...)  
Orden (n.º orden, fecha, usuario, total, ...)  
Pago (n.º pago, método, estado, monto, ...)
2. Establece Claves primarias y foráneas
3. Relaciona Orden–Producto con una tabla intermedia (OrderItems).
4. Dibuja un diagrama entidad-relación (ER)
5. Refleja cardinalidades (uno-a-muchos entre Usuario→Orden, muchos-a-muchos entre Orden↔Producto, uno-a-uno entre Orden→Pago).
6. Define reglas de integridad  
ON DELETE / ON UPDATE  
Restricciones de unicidad (email único, SKU único, etc.).



# Para el mentor

## Entidades y Atributos

### 1. Producto

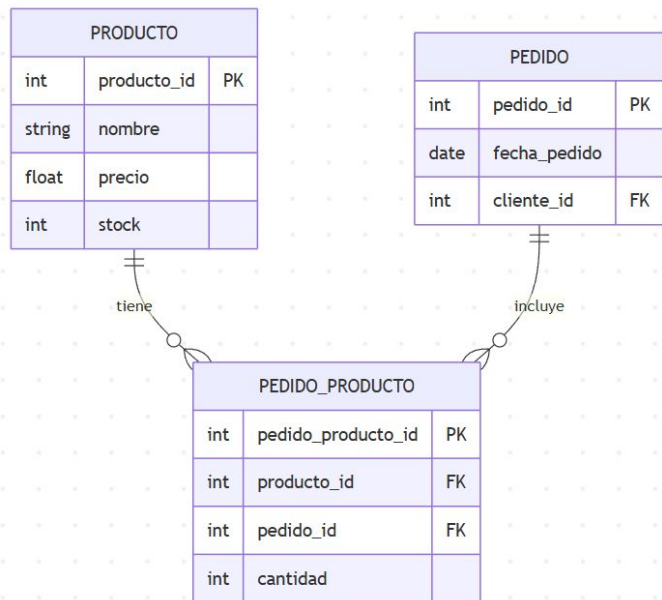
- `producto_id` (PK, Identificador único)
- `nombre` (Nombre del producto)
- `precio` (Costo del producto)
- `stock` (Cantidad disponible)

### 2. Pedido

- `pedido_id` (PK, Identificador único)
- `fecha_pedido` (Fecha en la que se realizó el pedido)
- `cliente_id` (FK, Cliente que realizó el pedido)

### 3. Relación Producto - Pedido

- `pedido_producto_id` (PK, Identificador único)
- `producto_id` (FK, Hace referencia a Producto)
- `pedido_id` (FK, Hace referencia a Pedido)
- `cantidad` (Cantidad del producto en el pedido)



```
SELECT nombre
FROM Producto
WHERE stock = 0;
```

¿Alguna consulta?





# Resumen

¿Qué logramos en esta clase?



- ✓ Conocer qué es el proceso de *abstracción* en bases de datos
- ✓ Conocer el modelo conceptual de *Entidad-Relación*
- ✓ Crear diagramas de flujo ER utilizando software de diagramas
- ✓ Definir los tipos de relaciones entre tablas en una base de datos
- ✓ Diferenciar los tipos de relaciones entre tablas
- ✓ Crear un atributo en una tabla
- ✓ Crear un identificador único en una tabla



## ¡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!** >

