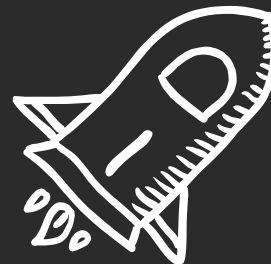




Semana 7

¡Bienvenidos!



Temas de hoy

1

Diseño de Bases de
Datos

2

Normalización y
desnormalización

3

Diagramas
entidad-relación (ERD)

1

Diseño de Bases de datos

Así como en un supermercado, donde todo tiene su lugar y orden, en una base de datos también necesitamos organización para almacenar nuestra información.

Vamos a aprender a estructurar la información, para que cada dato esté bien guardado y listo para ser servido cuando lo necesitemos.

¡Vamos a ver como diseñar una base de datos!








Diseño de la base de datos

¿Qué es el diseño de la base de datos?

Diseñar una BD es como **planificar** un asado desde cero: decidir cuánto carbón o leña vas a necesitar, cuántos kilos de carne vas a comprar, y cómo organizar todo para que el asado sea un éxito.

Un buen diseño evita problemas a futuro, como que te falten ingredientes o que se mezclen los sabores. De la misma manera, un buen diseño de base de datos **evita inconsistencias** y **facilita la administración** de la información.

Antes de empezar a diseñar, tenés que saber qué necesitas. En esta etapa, **se recopila** toda la información relevante sobre lo que la base de datos debe soportar: tipos de datos, relaciones, y volumen esperado.

Entrevistas de usuario		
PREGUNTAS		RESPUESTAS
		  

Diseño de la base de datos

¿Te imaginás cómo gestiona la información un restaurante que tiene muchos pedidos o un banco que tiene muuucha información?

Si, con una BD, pero previo a poner en funcionamiento la misma, tuvieron que hacer un trabajo de investigación en cuanto a: Qué van a necesitar almacenar en su base de datos y que tipos de datos va a recibir cada campo dentro de una “tabla”.

Veníamos hablando en clases de una biblioteca, vamos a tomarlo como ejemplo y poder pensar juntos...

¿Qué datos necesita una biblioteca que almacena en su depósito libros que pueden ser rentados a sus socios?

¿Alcanza con solo saber el título y el autor de cada libro? Definitivamente no.

Además de estos datos, vas a necesitar saber quienes son sus socios, sus datos de contacto ante alguna eventualidad, sus datos personales para asociarlos al alquiler de un libro, etc.

Todo esto que pensamos forma parte de la definición de la estructura que va a tener nuestra BD.

Necesitamos entender la lógica detrás de cada proyecto para que la base de datos esté diseñada de la mejor manera posible para evitar futuras eventualidades.

Tomarse el tiempo de hacer todas las consultas previas, o leer bien los requerimientos y analizar la información brindada es esencial en este proceso.

A su vez, podemos definir que tipo de “relaciones” tendrán entre sí estos datos, para poder reflejar esta relación dentro de la BD.

Por ejemplo, para poder saber que **libro** alquiló un **socio** tendríamos que tener una *relación* entre *libro* y *socio* para almacenar en algún lugar los libros alquilados dentro del socio o el socio que lo rentó dentro de libro. Cómo se decide esto lo veremos en breve ... 😊

2

Normalización

La normalización es cómo asegurarte de que no comprás más carne de la que podés cocinar. Implica dividir las tablas de la base de datos para eliminar redundancias y asegurar que cada pieza de información esté en su lugar adecuado.

Normalización

Organizando los datos de manera efectiva

La normalización es un proceso fundamental en el diseño de BD que tiene como objetivo organizar los datos de manera eficiente, eliminando redundancias y asegurando la integridad de la información. Este proceso se lleva a cabo dividiendo una BD en tablas más pequeñas y estableciendo relaciones entre ellas según ciertas reglas llamadas "formas normales".

Pero...

¿Qué son las **Formas Normales**?

La normalización se realiza en varias etapas, llamadas formas normales. Cada forma normal tiene un conjunto de reglas específicas.

- La **normalización** es un balance entre mantener la integridad de los datos y la eficiencia en el acceso a ellos. Comprender cuándo y cómo aplicar estas reglas es clave para un diseño de base de datos robusto y escalable.

Ventajas de la Normalización

- **Consistencia de Datos:** Minimiza las inconsistencias en la base de datos, lo que garantiza que los datos sean fiables y precisos.
- **Ahorro de Espacio:** Al eliminar redundancias, la normalización reduce el espacio de almacenamiento necesario.
- **Facilidad de Mantenimiento:** Los cambios en la base de datos se vuelven más sencillos, ya que la información se almacena en un solo lugar.

Normalización: ejemplo

Imaginá que tenés una tabla en una base de datos para almacenar información sobre clientes y sus pedidos. La tabla se ve así:

El nombre y la dirección del cliente están repetidos para cada pedido que hace. Si “Juan Pérez” cambia su dirección, habría que actualizarla en varias filas.



ID_pedido	Nombre_cliente	Direccion_cliente	Producto	Cantidad

Para eliminar la redundancia y simplificar la actualización, dividimos la información en dos tablas:

- **Menos Redundancia:** La información del cliente se almacena en una sola tabla.
- **Facilidad de Actualización:** Si “Juan Pérez” cambia su dirección, solo vas a necesitar actualizarla en una fila de la tabla de clientes.



ID_cliente	Nombre_cliente	Direccion_cliente

ID_pedido	ID_cliente	Producto	Cantidad

Desnormalización

Reorganizando los datos de otra forma

La **desnormalización** es el proceso de revertir parcialmente la normalización en una base de datos para mejorar el rendimiento en ciertas situaciones.

Aunque la normalización es excelente para eliminar redundancias y mantener la integridad de los datos, puede, *en algunos casos*, hacer que las consultas sean más complejas y lentas, especialmente en sistemas donde se requieren respuestas rápidas a gran volumen de datos.

¿Cuándo y por qué se utiliza la Desnormalización?

→ Mejora del Rendimiento:

En bases de datos muy grandes, con millones de registros, las consultas que requieren unir varias tablas

pueden volverse lentas. La desnormalización reduce la cantidad de uniones necesarias al almacenar algunos datos redundantes en una sola tabla, lo que acelera el acceso a la información.

→ Optimización de Consultas Específicas:

Si una consulta se realiza con mucha frecuencia y necesita acceder a datos de múltiples tablas, puede ser beneficioso desnormalizar para que esos datos se encuentren en una sola tabla. Esto minimiza el tiempo de ejecución de la consulta.

→ Simplicidad de Consultas:

En algunos casos, la desnormalización simplifica el diseño de la BD desde la perspectiva del usuario, permitiendo que las consultas sean menos complejas, especialmente en sistemas donde la BD se consulta con herramientas que no soportan fácilmente uniones complejas.

Desnormalización: ejemplo

- **Copiar Datos de una Tabla a Otra:**

Supongamos que tenés una tabla de *clientes* y una tabla de *pedidos*. Normalmente, cada *pedido* tendría una referencia a un *cliente* a través de una *clave foránea*. Pero si constantemente necesitás consultar el nombre del *cliente* junto con los detalles del *pedido*, podrías desnormalizar y agregar una columna *nombre del cliente* directamente en la tabla de *pedidos*. Esto evita la necesidad de unir las dos tablas en cada consulta.

- **Agrupación de Datos Calculados:**

Otro ejemplo es cuando se utilizan datos agregados o calculados con frecuencia. Por ejemplo, si constantemente necesitás el total de ventas de un *cliente*, en lugar de calcularlo cada vez que se consulta, podés desnormalizar y almacenar este total directamente en la tabla del *cliente*, actualizándolo cada vez que se realiza una nueva venta.

❑ **Ventajas de la Desnormalización**

- **Mejora del Rendimiento:** Al reducir el número de uniones y simplificar las consultas, se mejora la velocidad de acceso a los datos.
- **Consultas más Simples:** Las consultas que antes requerían unir varias tablas ahora pueden realizarse en una sola tabla, lo que simplifica el código y reduce la complejidad.
- **Reducción de Carga en el Sistema:** Menos operaciones de unión significan menos carga en el servidor de bases de datos, lo que puede ser crucial en sistemas de alta demanda.

❑ **Desventajas de la Desnormalización**

- **Aumento de Redundancia:** Al duplicar datos en múltiples tablas, se corre el riesgo de inconsistencias, ya que los datos redundantes deben mantenerse sincronizados manualmente o mediante triggers.
- **Mantenimiento más Complejo:** Las actualizaciones y eliminaciones en la base de datos se vuelven más complicadas, ya que los cambios deben reflejarse en varios lugares.
- **Mayor Uso de Espacio:** Al almacenar datos redundantes, el tamaño total de la base de datos puede aumentar significativamente.

3

Diagrama Entidad-Relación (ERD)

Y ahora hablemos de la herramienta que tenemos para diagramar esto que estamos pensando...

Si, como lees en el título, Diagrama Entidad-Relación es una de las herramientas que podemos usar, ¡previas a la construcción de la base de datos!

Vamos a ver de que se trata.



Diagrama ERD

Diagramando de forma visual

El **Diagrama Entidad-Relación (ERD)** es una herramienta visual utilizada en el diseño de BD para representar la estructura y las relaciones entre los datos.

Es fundamental para comprender cómo se organiza la información y cómo se interrelacionan las diferentes partes de una BD.

Componentes Clave del Diagrama Entidad-Relación

- **Entidades**

Una entidad **representa un objeto** o concepto en el sistema que tiene una existencia independiente y sobre el cual se desea almacenar información.

→ Ejemplos: Cliente, Pedido, Producto.

→ Representación en el Diagrama: Se representan con **rectángulos**. El nombre de la entidad se escribe dentro del rectángulo.

- **Atributos**

Los atributos son las **características o propiedades** de una entidad. Cada atributo describe una pieza específica de información sobre una entidad.

→ Ejemplos: Un Cliente puede tener atributos como Nombre, Dirección, Teléfono.

→ Representación en el Diagrama: Los atributos se representan con **óvalos** conectados a sus respectivas entidades. Los atributos clave (como el ID del Cliente) se **subrayan** para destacarlos.

- **Relaciones**

→ Definición: Una relación describe cómo las entidades están relacionadas entre sí. Indica cómo los datos en una entidad se conectan con los datos en otra entidad.

Diagrama ERD

- Ejemplos: Un Cliente realiza un Pedido, un Pedido contiene Productos.
- Representación en el Diagrama: Las relaciones se representan con rombos conectados por líneas a las entidades involucradas. El nombre de la relación se escribe dentro del rombo.
- **Cardinalidad**
La cardinalidad especifica el **número de instancias** de una **entidad** que pueden o deben estar **asociadas con instancias** de **otra entidad** en una relación.
- Ejemplos: Un Cliente puede hacer muchos Pedidos (*relación uno a muchos*), pero cada Pedido pertenece a un solo Cliente.
- **Representación en el Diagrama**: Se indica mediante anotaciones cerca de las líneas que conectan las entidades. Las notaciones comunes incluyen **1** (uno) y **N** (muchos).
- **Tipos de Relaciones**
- **Uno a Uno (1:1)**: Una instancia de una entidad está asociada con una instancia de otra entidad.
Ejemplo: Cada pasaporte pertenece a una sola persona.
- **Uno a Muchos (1:M)**: Una instancia de una entidad está asociada con muchas instancias de otra entidad.
Ejemplo: Un Cliente puede hacer muchos Pedidos.
- **Muchos a Muchos (M:N)**: Muchas instancias de una entidad están asociadas con muchas instancias de otra entidad.
Ejemplo: Un Pedido puede contener muchos Productos y un Producto puede estar en muchos Pedidos.

ERD: ejemplo

Ejemplo de un Diagrama Entidad-Relación

Imaginemos una base de datos para una tienda en línea. Queremos modelar las entidades Cliente, Pedido y Producto, y las relaciones entre ellas.

Entidades y Atributos:

Entidad: **Cliente**

Atributos: ID Cliente (clave primaria), Nombre, Dirección, Teléfono.

Entidad: **Pedido**

Atributos: ID Pedido (clave primaria), Fecha, ID Cliente (clave foránea).

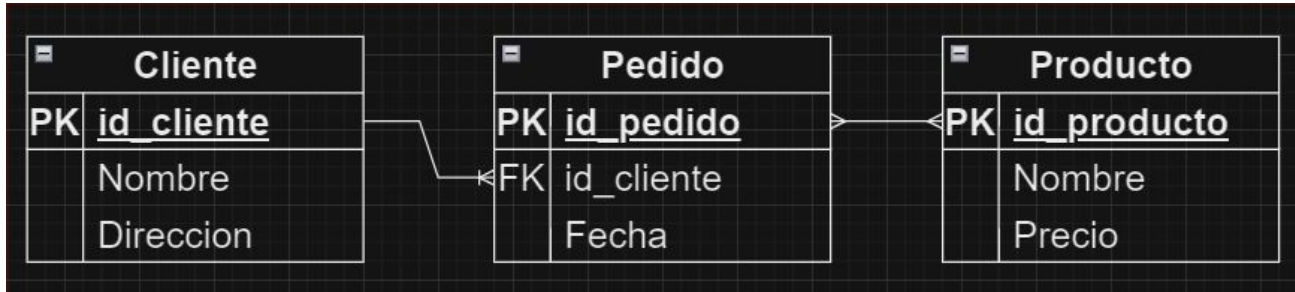
Entidad: **Producto**

Atributos: ID Producto (clave primaria), Nombre, Precio.

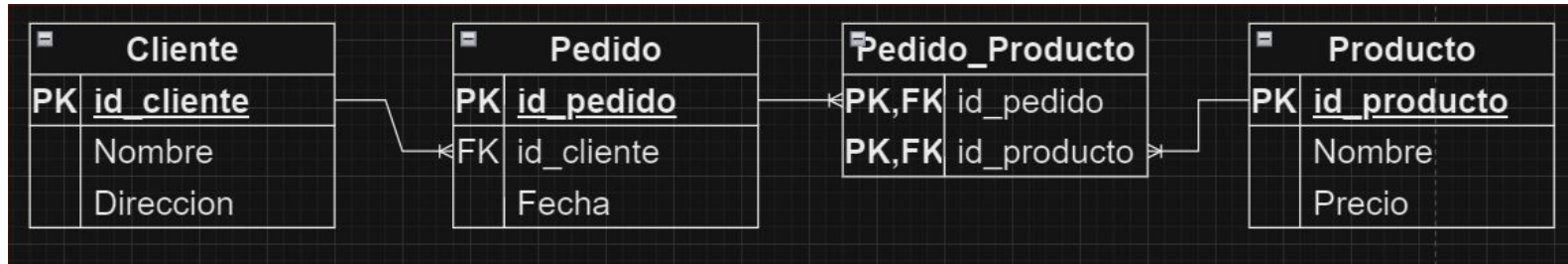
Beneficios del Diagrama Entidad-Relación

- **Visualización Clara:** Proporciona una representación gráfica clara y comprensible de la estructura de la base de datos.
- **Facilita el Diseño:** Ayuda a planificar y diseñar la base de datos antes de la implementación, identificando posibles problemas y redundancias.
- **Comunicación Eficaz:** Facilita la comunicación entre diseñadores de bases de datos y partes interesadas, asegurando que todos comprendan la estructura y las relaciones de los datos.

ERD: ejemplo



O bien :



ERD

Conclusión

El Diagrama Entidad-Relación es una herramienta esencial en el diseño de bases de datos, ya que proporciona una visión clara de cómo se organiza la información y cómo interactúan las distintas partes del sistema. Con un ERD bien diseñado, es más fácil construir una base de datos eficiente, mantener la integridad de los datos y satisfacer los requisitos del sistema.



Entonces...

El Diagrama Entidad-Relación es una herramienta esencial en el diseño de bases de datos, ya que proporciona una visión clara de cómo se organiza la información y cómo interactúan las distintas partes del sistema. Con un ERD bien diseñado, es más fácil construir una base de datos eficiente, mantener la integridad de los datos y satisfacer los requisitos del sistema.



Lo que vimos hoy 🤔

Hasta acá pudimos ver el paso previo a la construcción de una base de datos que asegura una estructura bien definida para evitar inconvenientes futuros y, estamos listos para ver en clase siguiente el paso de un **MER** a **Tablas** e introducimos al conocimiento del lenguaje de consultas y sus sentencias para las distintas operaciones en una base de datos.



**¡Nos vemos
En la próxima
clase!**

