

# Diseño y Modelado de Bases de Datos

Fundamentos clave sobre el diseño conceptual, lógico y físico de bases de datos, así como el modelado NoSQL y consideraciones de integridad y escalabilidad.

# Diseño Conceptual

- Entender los requerimientos del negocio.
- Representar la estructura lógica de la base de datos.
- Uso de diagramas entidad-relación (ER). ✓  $\rightarrow$  Álgebra Relacional.
- Definición de entidades, atributos y relaciones.

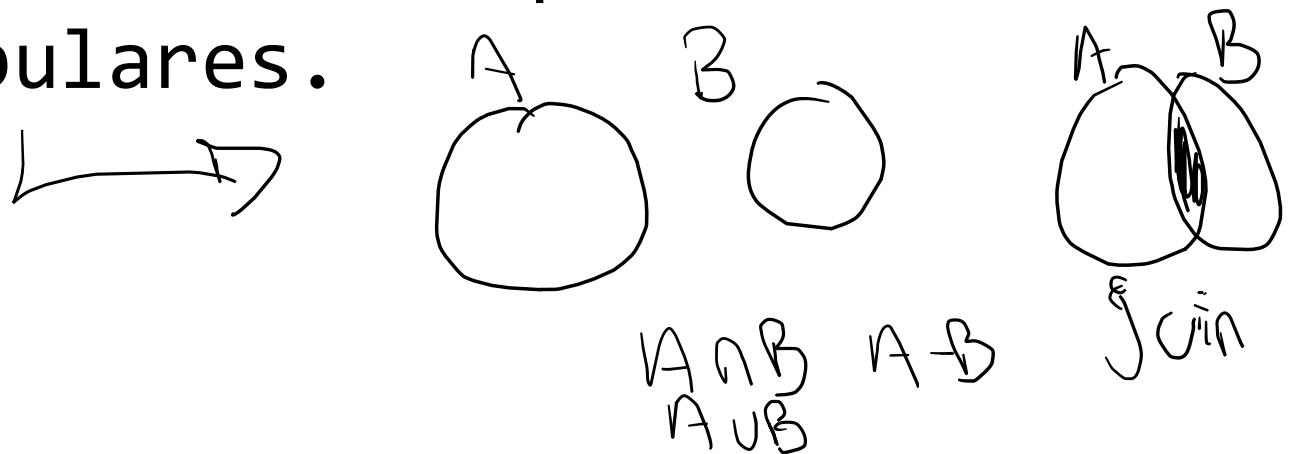
# - Arquitectura SW.

- 1) Requerimientos
- 2) Diseño → BD → Tablas  
↳ B.D Relaciones. 4
- 3) Implementación → workbench. MySQL  
SQLite
- 4) Pruebas
- 5) Implementación

Information

# Diseño Lógico

- Conversión del esquema conceptual a un formato compatible con SGBD.
- Definición de tablas, llaves primarias y foráneas.
- Modelo relacional para bases de datos tabulares.

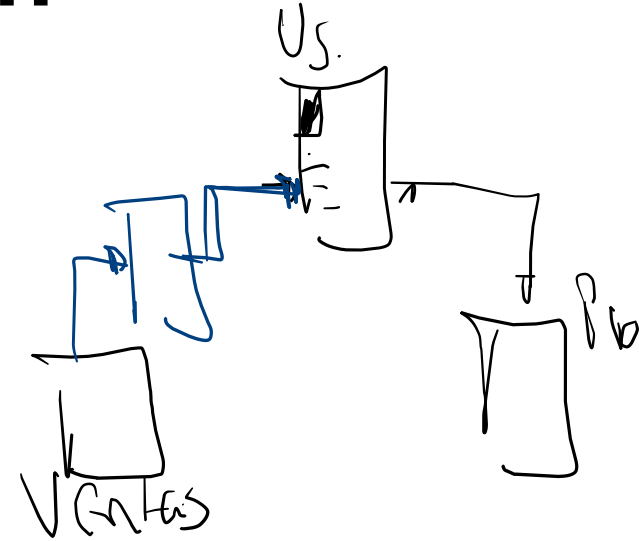


SLDB → Sistema de Gestión de Base de Datos

- Crear \* Workbench. \*
- actualizar
- consultas

# Normalización

- Proceso para minimizar redundancias.
- Evitar problemas de inconsistencia.
- Aplicación de formas normales (hasta 3FN) para distribuir datos correctamente.

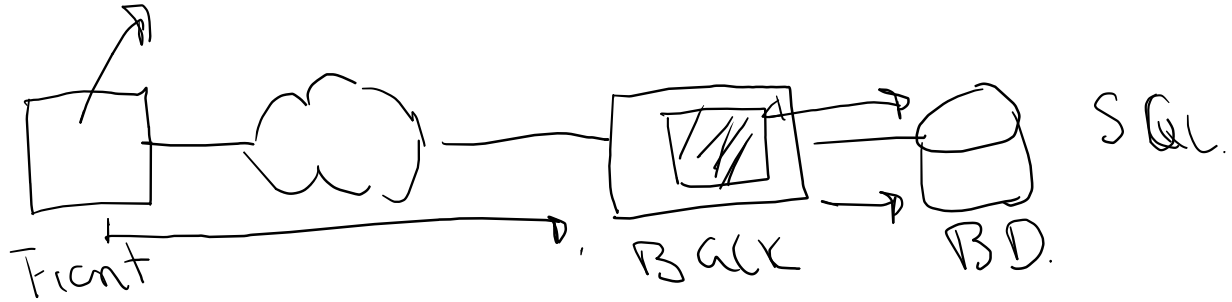


# Diseño Físico

- Implementación del diseño lógico en un SGBD específico (MySQL, PostgreSQL, etc.).
- Decisión sobre índices, almacenamiento y acceso a los datos.

→ Despliegue

S.I. Query → Concerns → tomorrow Decisions

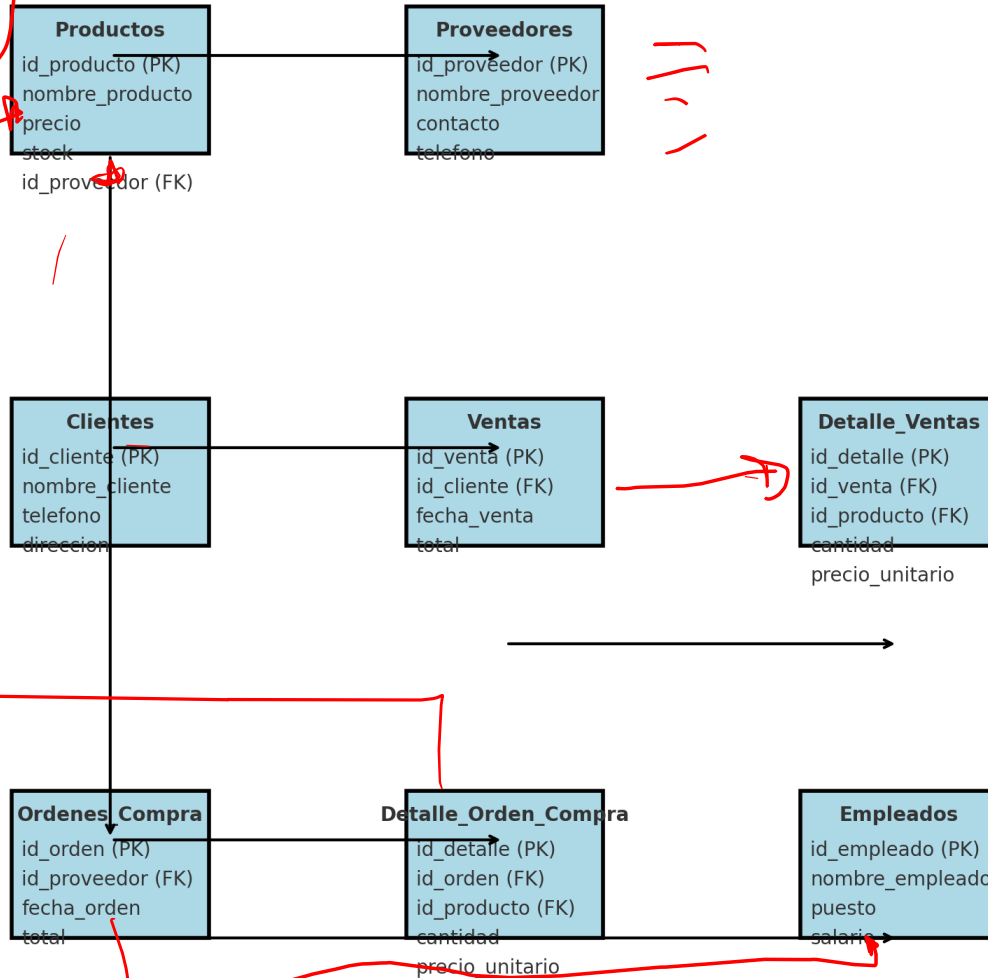


Create  
Read  
Update  
Delete.



# Ejemplo

### Diagrama Relacional de la Base de Datos del Almacén

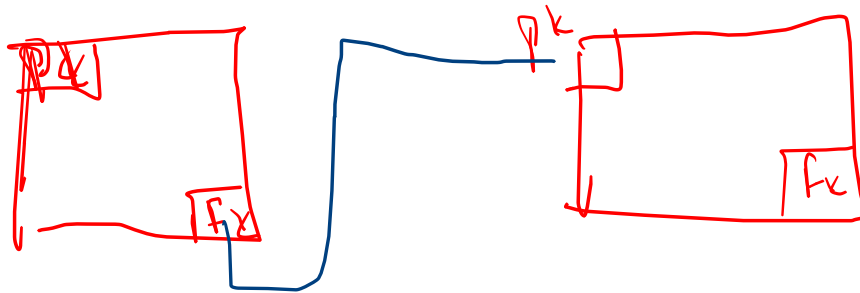


# Modelado NoSQL

- Modelado de estructuras no tabulares (MongoDB, Cassandra).
  - Optimización de consultas específicas en lugar de normalización.
  - Tipos de bases NoSQL: Documentos, grafos, clave-valor, columnas.
- Handwritten notes:*
- Mongo B.D (with an arrow pointing to the 'Documentos' type)
  - Redis (with an arrow pointing to the 'clave-valor' type)
  - diccionario (with an arrow pointing to the 'clave-valor' type)

# Integridad de los Datos

- Definición de reglas para asegurar datos válidos y consistentes.
- Integridad referencial entre tablas mediante claves foráneas.



# Escalabilidad y Rendimiento

- Consideraciones para escalabilidad de la base de datos.
- Uso de índices, particiones y replicación para mejorar consultas y manejo de grandes volúmenes.

