

Índice

Vocablo.....	2
Repasso – Preguntas.....	2
Sobre Archivos.....	2
Sobre BD.....	2
Sobre DER.....	2
Sobre Claves	3
Resumen de Archivos.....	4
Archivo	4
Registro	4
Campo	4
Clasificaciones:	4
Métodos de organización de archivos.	4
Archivo secuencial.....	4
Archivo Indexado.....	4
Archivo Secuencial Indexado.....	4
Archivo Directo.....	5
Comparación entre archivos	5
Resumen de que es una Base de Datos	6
¿Qué es un SBD?	6
Esquemas de diseño (3)	7
Diseño Conceptual	7
Diseño Lógico	7
Diseño Físico.....	7
Ventajas (3)	7
Desventajas (3).....	7
Componentes de los sistemas de base de datos (6)	8
Arquitectura General (3)	8
Nivel Interno.....	8
Funciones del DBA (6)	8
SGBD.....	8
Funciones del SGBD.....	9
Lenguajes de SGBD	10
DER	11
Generalización:.....	11

Teoría de Normalización	11
1FN	12
2FN	12
3FN	12
Claves en una BD	13
Clave Primaria	13
Clave candidata	13
Clave secundaria	13
Clave foránea	13

Vocablo

SBD = Sistemas de Bases de Datos

SGBD = Sistema Gestor de Base de Datos

DBA = Data Base Admin / Administrador de base de datos

Repasso – Preguntas

Sobre Archivos

- ⇒ ¿Qué es un archivo?
- ⇒ ¿Qué es un registro?
- ⇒ ¿Qué es un campo?
- ⇒ ¿Qué es una clave?
- ⇒ Clasificación de archivos
- ⇒ ¿Cuáles son los tipos de archivos que existen?
- ⇒ Compare el uso de memoria y la eficacia de las búsquedas y consultas de cada tipo de archivo.

Sobre BD

- ⇒ ¿Qué es una BD? ¿Qué características tiene?
- ⇒ Defina las 3 etapas del diseño.
- ⇒ Nombre algunas ventajas / desventajas de una BD
- ⇒ ¿Cuáles son los componentes de los sistemas de bases de datos?
- ⇒ Defina los 3 niveles de arquitectura. (Vistas)
- ⇒ ¿Cuáles son las funciones que debe cumplir un DBA?
- ⇒ ¿Qué es un SGBD?
- ⇒ ¿Qué funciones debe proveer un SGBD?
- ⇒ ¿Cuáles son los lenguajes que maneja un SGBD?

Sobre DER

- ⇒ ¿Cuál es el objetivo de la normalización?
- ⇒ ¿Porque es buena la normalización?
- ⇒ Defina la 1FN, la 2FN y la 3FN.

Sobre Claves

- ⇒ ¿Cuántos tipos de claves existen?
- ⇒ Defina cada uno de los tipos de claves que existen.

Resumen de Archivos

Archivo

Es un conjunto de registros homogéneos que contiene datos heterogéneos.

Registro

Conjunto de campos relacionados que contienen datos elementales.

Campo

Unidad atómica del registro.

Clasificaciones:

Según tipo de vida:

- ⇒ Persistente: Como los archivos que conforman una BD.
- ⇒ Temporal.

Según tipo de información:

- ⇒ Programa (almacenan instrucciones para manipular datos).
- ⇒ Datos

Métodos de organización de archivos.

Archivo secuencial

Para acceder a un dato N debo pasar por los N-1 anteriores. No existen posiciones sin usar, se accede a un dato de la forma en la que se fueron escribiendo uno a uno.

Alta: Agrega un registro al final del archivo.

Baja: No existe baja física, lo marca como una baja lógica con un valor testigo.

Modificación: Se puede reescribir en un registro.

Archivo Indexado

Tiene un área de datos y un área de índices, asocia una clave de registro a una dirección relativa en otra la otra tabla (la de índices).

El área de datos esta direccionada por el área de índices.

El área de índices contiene las claves y la dirección donde se ubican.

Alta: Se inserta el registro en el área de datos, y se guarda la clave con su dirección relativa en el área de índices.

Baja: Tiene baja física, se elimina el registro del área de índices y se cambia el puntero. O se hace una baja lógica.

Modificación: no creo que haga falta describir esto verdad?

Archivo Secuencial Indexado

El del tp, tiene 2 tablas como el anterior, una de índices y una de datos. La de datos se divide por bloques teniendo a lo último un área de overflow. La tabla de índices solo almacena índices del primer clave de cada bloque, junto a su dirección relativa.

Archivo Directo

Esto es lo que usábamos en C++ cuando abríamos un archivo.

Nos permite acceder de forma rápida a los registros usando su posición relativa, ya que tiene un acceso directo. Se establece el tamaño del archivo (filesize) y se lo ubica en memoria para usar. Es el más costoso en términos de memoria, pero más rápido en cuanto a consultas.

Comparación entre archivos

ANÁLISIS ENTRE COSTO Y BENEFICIO

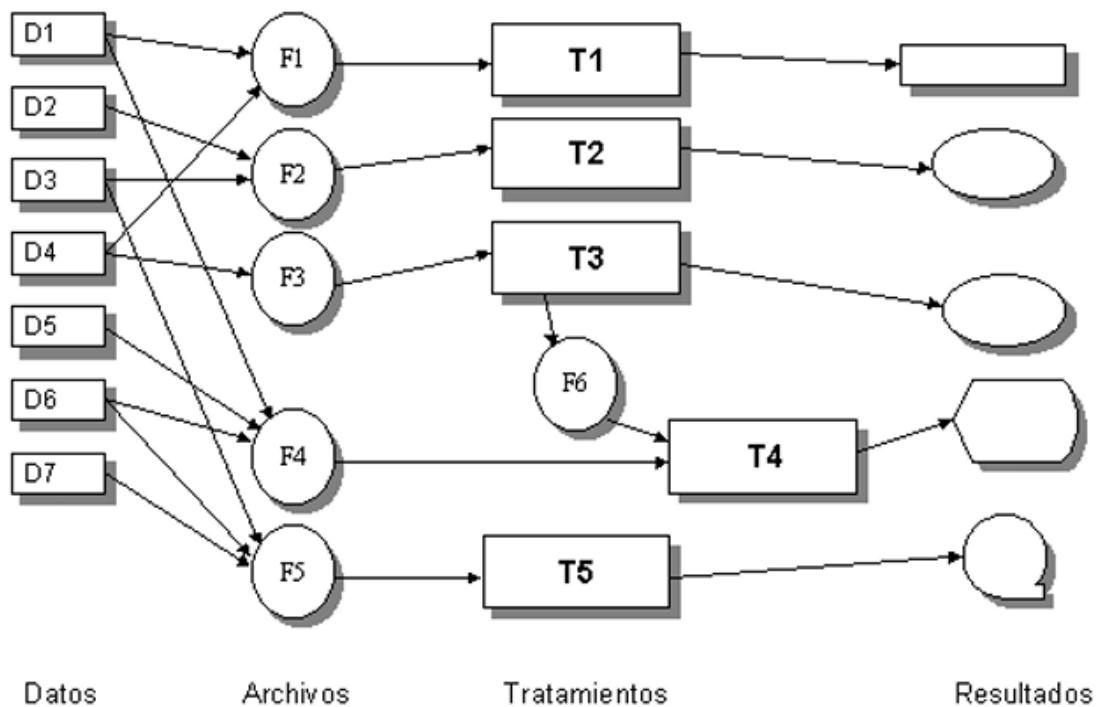
	Archivo Secuencial	Archivo Indexado	Archivo Secuencial Indexado	Archivo Directo
Uso de la memoria	Óptimo.	Utiliza más memoria que un archivo secuencial, pero es justificado ya que tiene un acceso bastante rápido.	Bueno.	Regular, ya que se debe dimensionar el archivo antes de crearlo, con lo que puedo llegar a malgastar memoria si sobran registros, o puede llegar a ser poca la cantidad de registros reservados.
Eficiencia en búsquedas y consultas	Es ineficiente, ya que se debe recorrer uno a uno todos los registros hasta encontrar el que se desea. El costo de las búsquedas y consultas es alto; para disminuirlo se sugiere ordenar los registros (en forma ascendente o descendente) en función de algún campo de los registros.	Son bastante eficientes, más rápido que el secuencial y que el secuencial indexado, pero más lento que el directo.	Buena, es más rápida que la secuencial, aunque más lenta que la directa.	Bastante eficiente, ya que posee costo de acceso 1.

Resumen de que es una Base de Datos

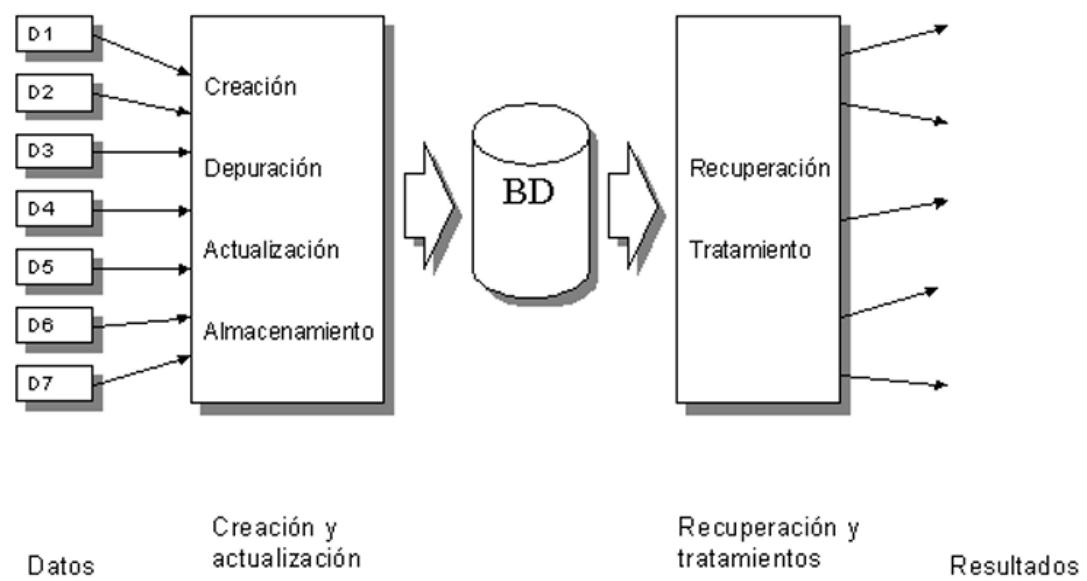
El PPT está bastante bien resumido, pero acá recopilo todo para que me quede y ignore algunas cosas, sin embargo, recomiendo encarecidamente leer el ppt.

¿Qué es un SBD?

Datos => Archivos => Tratamientos => Resultados



Estructura en el manejo de datos a través de bases de datos



Es una colección de datos persistentes e independientes, utilizados por una determinada organización. Esta colección de datos está estructurada en base a las relaciones naturales entre ellos, de modo que provee los caminos necesarios para atender a la necesidad de cada individuo que los necesite. Esta colección de datos a su vez es de redundancia mínima.

Una BD es **Integrada** y **Compartida**, porque esta **integrada** por archivos de forma que se elimina la redundancia total o parcialmente, y **compartida** porque cada pieza de la BD puede ser accedida por un conjunto de usuarios.

Esquemas de diseño (3)

Diseño Conceptual

Representación total y abstracta de los datos que componen la base, acá es cuando juega el DER. Mediante técnicas de relevamiento obtenemos como debería estar armada la estructura.

Diseño Lógico

Implementación del diseño conceptual utilizando un SGBD. Juega SQL pa.

Las entidades se convierten en tablas, las relaciones de 1:N se convierte en propagación de claves (primarias o foráneas), aunque a veces se crean tablas intermedias. Y las relaciones de N:M se convierten en tablas intermedias.

Diseño Físico

Implementar de forma eficiente el diseño lógico, es dependiente del SGBD y del equipo donde se implementa.

Ventajas (3)

Tiene bocha de ventajas, y no creo que las tomen, pero acá van las importantes:

- Los datos:
 - o Tienen más independencia para tratamientos
 - o Mayor disponibilidad
 - o Eficaz recuperación, codificación y entrada.
- Resultados:
 - o Mas coherentes
 - o Mayor valor
- Usuarios:
 - o Acceso rápido y sencillo
 - o Fácil compartir datos
 - o Mas flexibilidad.

Desventajas (3)

- Implementación:
 - o Costosa a nivel lógico y físico.
 - o No hay muchos estándares.
 - o Larga y difícil puesta en marcha.
 - o Recién es rentable a medio plazo.
- Requiere personal especializado.
- La teoría y práctica están desfazadas.

Componentes de los sistemas de base de datos (6)

1. BD
2. SGBD/Motores de bases de datos
3. Programas de Aplicación
4. Conjunto de usuarios
5. Hardware
6. Programas utilitarios (Ayudan a manejar la BD, backup, dev, etc)

Arquitectura General (3)

Tiene 3 tipos de vistas/niveles:

- ⇒ Externa
Visión del usuario/grupo de usuarios.
- ⇒ Conceptual
Representación total y abstracta de los datos que componen la base, podría ser el DER.

Nivel Interno

Nivel mas bajo de la arquitectura, como esta compuesto el almacenamiento físico de los datos de la base. Es dependiente del SGBD (Es decir, dependiendo que sistema para gestionar la base de datos utilizamos, cambiara el nivel interno), pero algunas cosas se comparten como:

- Estrategias de almacenamiento
- Técnicas de tuning y optimización
- Caminos de acceso (claves primarias, foráneas y índices)
- Control de acceso
- Organizaciones físicas
- // otras cosas locas

Funciones del DBA (6)

Son quienes administran la BD.

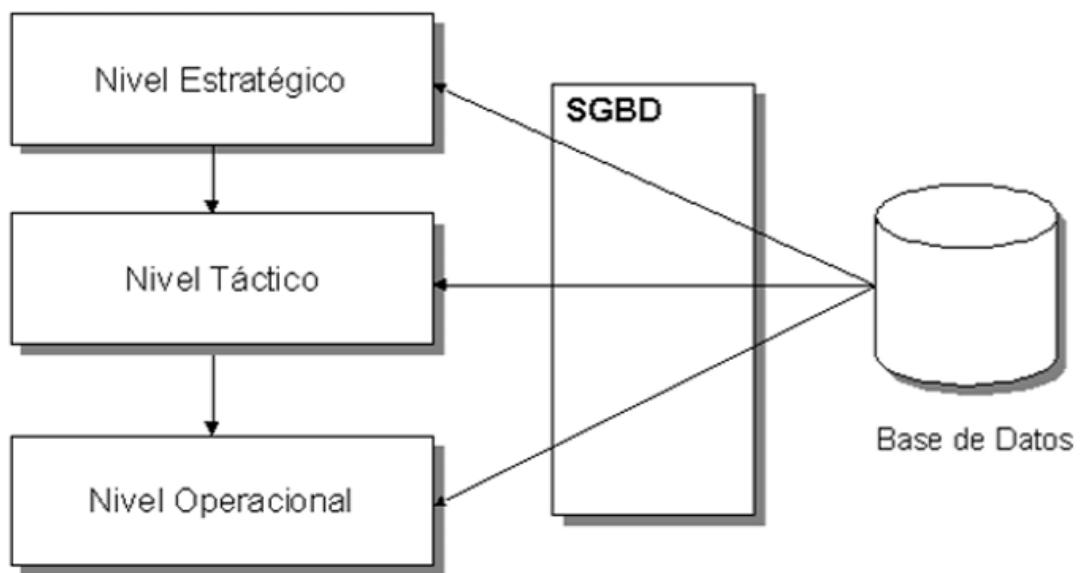
1. Decidir el contenido, como se identifican las entidades y que datos tendrán.
2. Decidir la estructura del almacenamiento y estrategia del acceso, como se representarán los datos de forma interna.
3. Vincularse con los usuarios.
4. Seguridad.
5. Definir una estrategia de respaldo y recuperación.
6. Controlar la BD para que tenga un desempeño aceptable.

SGBD

Es la colección de datos (SBD) + Programas para acceder a estos datos.

SBD + Programas = SGBD

Provee un lenguaje, herramientas y funciones para poder administrar la información de la bd, ya se extraer o almacenarla.



Funciones del SGBD

Debe proveer al DBA las herramientas adecuadas para implementar el diseño conceptual.

Debe proveer un lenguaje de definición de datos, y debe ser capaz de permitir definir las estructuras de datos de los 3 niveles, el externo, conceptual y interno.

⇒ A nivel interno:

- Espacio reservado para la base.
- Longitud de campos.
- Modo de representación de datos.
- Caminos de acceso como punteros e índices.

⇒ Nivel externo y conceptual:

- Instrumentos para definir entidades, sus atributos, relaciones, etc.

⇒ Manipulación de datos:

- Permite a los usuarios buscar, eliminar o modificar datos según las normas que da el DBA. (CRUD)

Esto se realiza gracias al lenguaje huésped (SQL o las instrucciones DML que se usen), y el uso del lenguaje anfitrión, que es el cual usamos para llamar a estas instrucciones.

4. Lenguaje huésped vs. lenguaje anfitrión

Esto es más técnico, pero lo explico:

- **Lenguaje huésped:** El SQL o las instrucciones DML que querés usar.
- **Lenguaje anfitrión:** El lenguaje de programación desde donde llamas a esas instrucciones. Ejemplo:

cpp

Copiar

Editar

```
// C++ sería el anfitrión
db.execute("SELECT * FROM productos"); // SQL sería el huésped
```

A veces no necesitás un anfitrión porque el lenguaje de base de datos ya es completo. Ahí entra el...

5. Lenguaje autocontenido:

Es un lenguaje que no necesita estar embebido en otro. Ejemplo: PL/SQL (de Oracle) o T-SQL (de SQL Server). Te permite hacer desde scripts simples hasta procedimientos almacenados, sin salirte del entorno de base de datos.

⇒ De utilización:

- Proporciona las interfaces necesarias para que los usuarios y el Admin la usen, dando funciones como cambiar el tamaño de los archivos, obtener estadísticas de uso, respaldos, seguridad, etc.

Lenguajes de SGBD

Existen distintos tipos de lenguajes depende la función, hay lenguajes de definición y de manipulación.

DDL: Data Definition Language

Lenguaje para definir la base de datos, acá se encuentran instrucciones como create, alter o drop.

DML: Data Manipulation Language

Lenguaje para interactuar con los datos, acá se encuentran instrucciones como select, drop, update, insert.

Normalmente la relación va así, de un lenguaje huésped a un lenguaje anfitrión, donde el lenguaje huésped es quien tiene las instrucciones para interactuar o definir la base de datos, ósea es el DML o DDL, aunque SQL ya acapara estos 2 lenguajes y otros mas (es como un contenedor de lenguajes), y luego el lenguaje anfitrión (que podría ser C++, por ejemplo) lo utiliza lanzando instrucciones SQL.

DER

Generalización:

(T,S)

T es de total, también existe P de parcial. S de superpuesta, también existe E de exclusiva.

Vamos a poner el ejemplo de la entidad Empleado, de la cual heredan Administrador y Técnico. Si T es total, quiere decir que, si tengo un empleado, este es administrador o técnico, alguna de las 2 tiene que ser, no puede ser ninguna; mientras que P nos dice que puede no ser ni administrador ni técnico.

En resumen, poner total es algo así como indicar que Empleado es una clase abstracta, por lo que no puede usarse.

S de superpuesta quiere decir que un empleado puede ser Admin y técnico a la vez, mientras que exclusiva, no hace falta ser experto para darse cuenta por el nombre que o es uno o es otro, pero no puede ser ambos.

Teoría de Normalización

Existen 6FN, pero solo se suelen aplicar las primeras 3.

La idea de la normalización es agarrar datos complejos y transformarlos en conjuntos de datos mas pequeños, de tal forma que se eliminen redundancias, dependencias y sea más fácil de entender y mantener.

Mejora el esquema lógico, elimina duplicidad de datos.

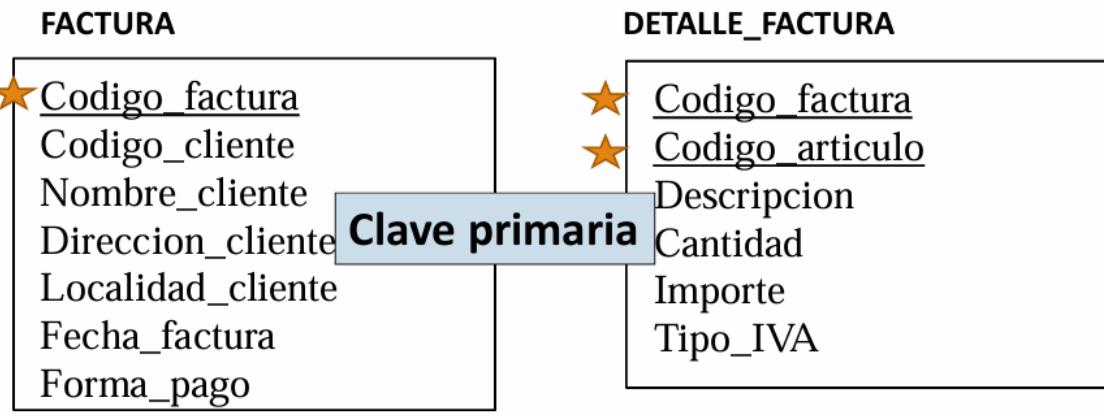
Evita anomalías, y minimiza el impacto por cambios en los datos a futuro. Da mas claridad y seguridad, y da mas rendimiento (entre otras ventajas).

Codigo_factura

Codigo_cliente
Nombre_cliente
Direccion_cliente
Localidad_cliente
Fecha_factura
Forma_pago
Codigo_articulo_1
Descripcion_1
Cantidad_1
Importe_1
Tipo_IVA_1
...
Codigo_articulo_N
Descripcion_N
Cantidad_N
Importe_N
Tipo_IVA_N

1FN

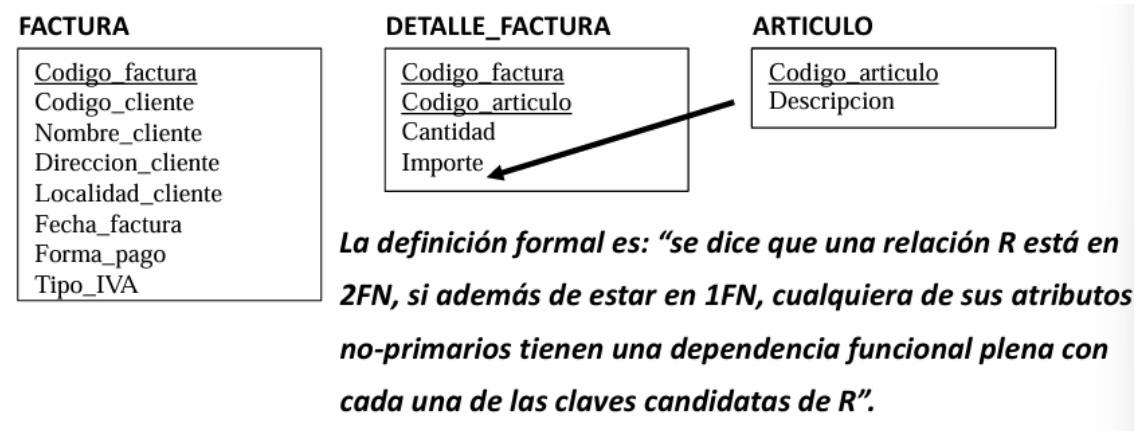
Para que una estructura de datos este en 1FN tenemos que garantizar que cada uno de sus datos tenga una estructura atómica, es decir, **que no haya datos duplicados o multivaluados**.



2FN

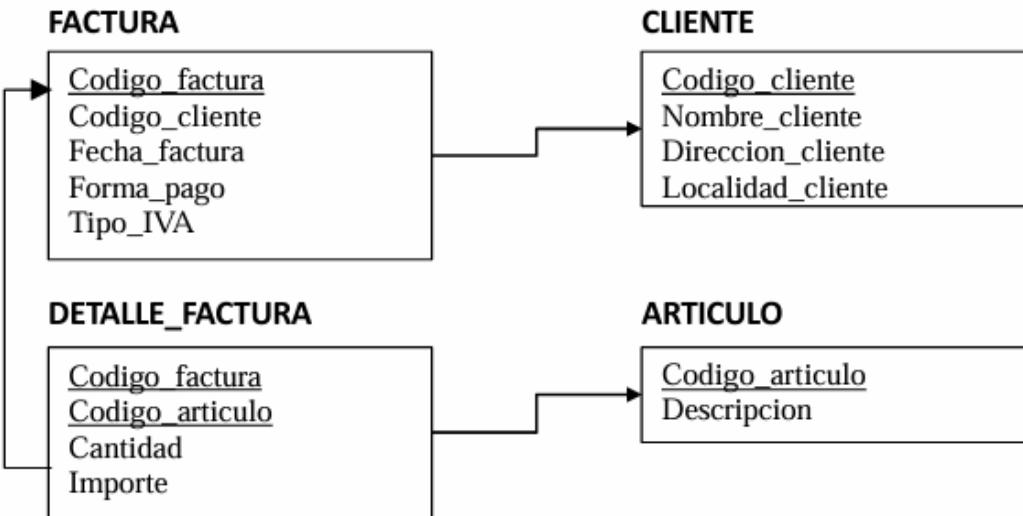
Debe estar en 1FN, y tenemos que garantizar que los atributos no clave dependan de la clave completa, y no solo de una parte. (Es decir, los atributos que no son clave dependan de la clave).

En criollo, si ves algo que no pinta nada con la clave de la tabla, fletealo.



3FN

Debe estar en 2FN y evitar la transitividad de los datos. Todos los datos que no son clave deben ser independientes entre sí.



Claves en una BD

Es un atributo que identifica al registro.

Clave Primaria

Identifica inequívocamente al registro. Puede ser un atributo o un conjunto de atributos.

Clave candidata

Columna con potencial para ser clave primaria, ya que cumple con los requisitos para representar al registro.

Clave secundaria

Se usa como un índice, permite al SGBD tener otra forma de ordenar una tabla, usando un criterio que no sea necesariamente el de la clave primaria. Es decir, que pueden ser otros campos de la propia tabla, la cual ya contiene una tabla primaria.

No tiene por qué ser único. Es decir, que no tiene por qué ser inequívoca.

Clave foránea

Apunta al registro único de otra tabla.

Cabecera_Factura

* Código

Tipo_Comprobante

Letra

Sucursal

Número

Fecha_Emision

Cod_Cliente

Domicilio_Entrega