Definimos un **Sistema Gestor de Bases de Datos** o **SGBD**, también llamado DBMS (*Data Base Management System*) como una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina **Base de Datos** o **BD**, (*DB Data Base*).

Antes de aparecer los SGBD (década de los setenta), la información se trataba y se gestionaba utilizando los típicos sistemas de gestión de archivos que iban soportados sobre un sistema operativo.

Éstos consistían en un conjunto de programas que definían y trabajaban sus propios datos. Los datos se almacenan en archivos y los programas manejan esos archivos para obtener la información.

Si la estructura de los datos de los archivos cambia, todos los programas que los manejan se deben modificar; por ejemplo, un programa trabaja con un archivo de datos de alumnos, con una estructura o registro ya definido; si se incorporan elementos o campos a la estructura del archivo, los programas que utilizan ese archivo se tienen que modificar para tratar esos nuevos elementos.

En estos sistemas de gestión de archivos, la definición de los datos se encuentra codificada dentro de los programas de aplicación en lugar de almacenarse de forma independiente, y además el control del acceso y la manipulación de los datos viene impuesto por los programas de aplicación.

Esto supone un gran inconveniente a la hora de tratar grandes volúmenes de información. Surge así la idea de separar los datos contenidos en los archivos de los programas que los manipulan, es decir, que se pueda modificar la estructura de los datos de los archivos sin que por ello se tengan que modificar los programas con los que trabajan.

Se trata de estructurar y organizar los datos de forma que se pueda acceder a ellos con independencia de los programas que los gestionan.

Inconvenientes de un sistema de gestión de archivos:

• **Redundancia e inconsistencia de los datos,** se produce porque los archivos son creados por distintos programas y van cambiando a lo largo del tiempo, es decir, pueden tener distintos formatos y los datos pueden estar duplicados en varios sitios. Por ejemplo, el teléfono de un alumno puede aparecer en más de un archivo. La redundancia aumenta los costes de almacenamiento y acceso, y trae consigo la inconsistencia de los datos: las copias de los mismos datos no coinciden por aparecer en varios archivos.

• **Dependencia de los datos física-lógica,** o lo que es lo mismo, la estructura física de los datos (definición de archivos y registros) se encuentra codificada en los programas de aplicación. Cualquier cambio en esa estructura implica al programador identificar, modificar y probar todos los programas que manipulan esos archivos.

• **Dificultad para tener acceso a los datos,** proliferación de programas, es decir, cada vez que se necesite una consulta que no fue prevista en el inicio implica la necesidad de codificar el programa de aplicación necesario. Lo que se trata de probar es que los entornos convencionales de procesamiento de archivos no permiten recuperar los datos necesarios de una forma conveniente y eficiente.

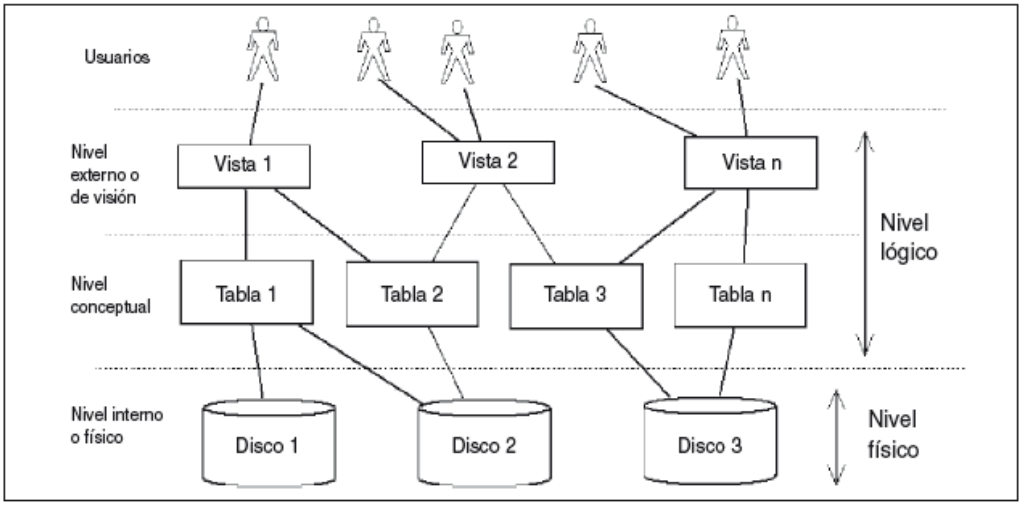
Arquitectura de los sistemas de bases de datos

En 1975, el comité ANSI-SPARC (*American National Standard Institute - Standards Planning and Requirements Committee*) propuso una arquitectura de tres niveles para los SGBD cuyo objetivo principal era el de separar los programas de aplicación de la BD física.
En esta arquitectura el esquema de una BD se define en tres niveles de abstracción distintos:

• **Nivel interno o físico:** el más cercano al almacenamiento físico, es decir, tal y como están almacenados en el ordenador. Describe la estructura física de la BD mediante un esquema interno.
Este esquema se especifica con un modelo físico y describe los detalles de cómo se almacenan físicamente los datos: los archivos que contienen la información, su organización, los métodos de acceso a los registros, los tipos de registros, la longitud, los campos que los componen, etcétera.

• **Nivel externo o de visión:** es el más cercano a los usuarios, es decir, es donde se describen varios esquemas externos o vistas de usuarios.
Cada esquema describe la parte de la BD que interesa a un grupo de usuarios en este nivel se representa la visión individual de un usuario o de un grupo de usuarios.

• **Nivel conceptual:** describe la estructura de toda la BD para un grupo de usuarios mediante un esquema conceptual. Este esquema describe las entidades, atributos, relaciones, operaciones de los usuarios y restricciones, ocultando los detalles de las estructuras físicas de almacenamiento. Representa la información contenida en la BD.



Lenguajes de los SGBD

Todos los SGBD ofrecen lenguajes e interfaces apropiadas para cada tipo de usuario:
administradores, diseñadores, programadores de aplicaciones y usuarios finales.
Los lenguajes van a permitir al administrador de la BD especificar los datos que componen la BD, su estructura, las relaciones que existen entre ellos, las reglas de integridad, los controles de acceso, las características de tipo físico y las vistas externas de los usuarios.

Los lenguajes del SGBD se clasifican en:

• **Lenguaje de definición de datos (LDD o DDL):** se utiliza para especificar el esquema de la BD, las vistas de los usuarios y las estructuras de almacenamiento.
Es el que define el esquema conceptual y el esquema interno. Lo utilizan los diseñadores y los administradores de la BD.

• **Lenguaje de manipulación de datos (LMD o DML):** se utilizan para leer y actualizar los datos de la BD. Es el utilizado por los usuarios para realizar consultas, inserciones, eliminaciones y modificaciones. Los hay *procedurales*, en los que el usuario será normalmente un programador y especifica las operaciones de acceso a los datos llamando a los procedimientos necesarios. Estos lenguajes acceden a un registro y lo procesan.

Las sentencias de un LMD procedural están embebidas en un lenguaje de alto nivel llamado *anfitrión*. Las BD jerárquicas y en red utilizan estos LMD procedurales.

No procedurales son los lenguajes declarativos. En muchos SGBD se pueden introducir interactivamente instrucciones del LMD desde un terminal, también pueden ir embebidas en un lenguaje de programación de alto nivel. Estos lenguajes permiten especificar los datos a obtener en una consulta, o los datos a modificar, mediante sentencias sencillas. Las BD

relacionales utilizan lenguajes no procedurales como SQL (*Structured Quero Language*) o QBE (*Query By Example*).

• La mayoría de los SGBD comerciales incluyen **lenguajes de cuarta generación (4GL)** que permiten al usuario desarrollar aplicaciones de forma fácil y rápida, también se les llama *herramientas de desarrollo*. Ejemplos de esto son las herramientas del SGBD

El diccionario de datos

El **diccionario de datos** es el lugar donde se deposita información acerca de todos los datos que forman la BD. Es una guía en la que se describe la BD y los objetos que la forman.
El diccionario contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo nombre, descripción, alias, contenido y organización. Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información.
En una BD relacional, el diccionario de datos proporciona información acerca de:
• La estructura lógica y física de la BD.
• Las definiciones de todos los objetos de la BD: tablas, vistas, índices, disparadores, procedimientos, funciones, etcétera.
• El espacio asignado y utilizado por los objetos.
• Los valores por defecto de las columnas de las tablas.
• Información acerca de las restricciones de integridad.
• Los privilegios y roles otorgados a los usuarios.
• Auditoría de información, como los accesos a los objetos.
Un diccionario de datos debe cumplir las siguientes características:
• Debe soportar las descripciones de los modelos conceptual, lógico, interno y externo de la BD.
• Debe estar integrado dentro del SGBD.
• Debe apoyar la transferencia eficiente de información al SGDB. La conexión entre los modelos interno y externo debe ser realizada en tiempo de ejecución.

• Debe comenzar con la reorganización de versiones de producción de la BD. Además debe reflejar los cambios en la descripción de la BD. Cualquier cambio a la descripción de programas ha de ser reflejado automáticamente en la librería de descripción de programas con la ayuda del diccionario de datos.
• Debe estar almacenado en un medio de almacenamiento con acceso directo para la fácil recuperación de información.

Seguridad e integridad de datos

Un SGBD proporciona los siguientes mecanismos para garantizar la seguridad e integridad de los datos:

• Debe garantizar la protección de los datos contra accesos no autorizados, tanto intencionados como accidentales. Debe controlar que sólo los usuarios autorizados accedan a la BD.

• Los SGBD ofrecen mecanismos para implantar restricciones de integridad en la BD. Estas restricciones van a proteger la BD contra daños accidentales. Los valores de los datos que se almacenan deben satisfacer ciertos tipos de restricciones de consistencia y reglas de integridad, que especificará el administrador de la BD. El SGBD puede determinar si se produce una violación de la restricción.

• Proporciona herramientas y mecanismos para la planificación y realización de copias de seguridad y restauración.

• Debe ser capaz de recuperar la BD llevándola a un estado consistente en caso de ocurrir algún suceso que la dañe.

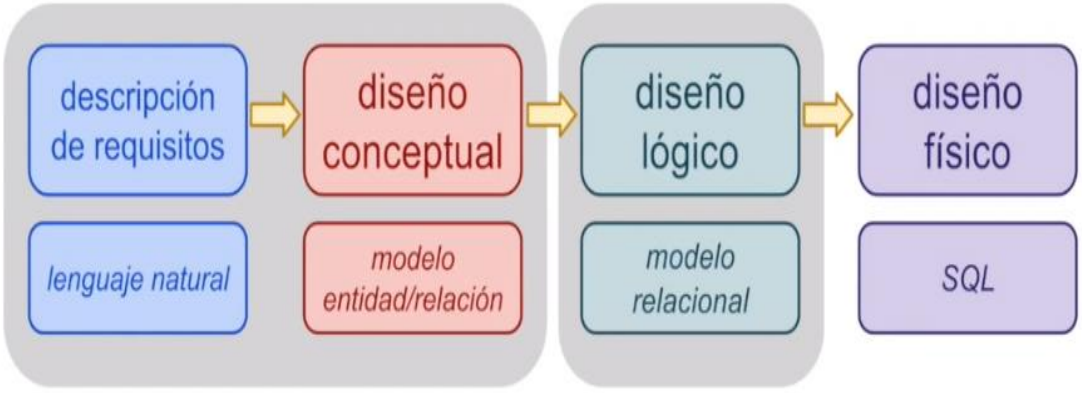
• Debe asegurar el acceso concurrente y ofrecer mecanismos para conservar la consistencia de los datos en el caso de que varios usuarios actualicen la BD de forma concurrente.

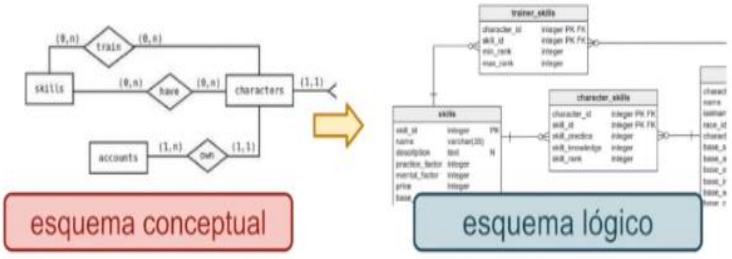
El administrador de la BD

En los sistemas de gestión de BBDD actuales existen diferentes categorías de usuarios.
Estas categorías se caracterizan porque cada una de ellas tiene una serie de privilegios o permisos sobre los objetos que forman la BD.
El DBA tiene una gran responsabilidad ya que posee el máximo nivel de privilegios. Será el encargado de crear los usuarios que se conectarán a la BD. En la administración de una BD siempre hay que procurar que haya el menor número de administradores, a ser posible una sola persona.
El objetivo principal de un DBA es garantizar que la BD cumple los fines previstos por la organización, lo que incluye una serie de tareas como:

Instalar SGBD en el sistema informático.
• Crear las BBDD que se vayan a gestionar.
• Crear y mantener el esquema de la BD.
• Crear y mantener las cuentas de usuario de la BD.
• Arrancar y parar SGBD, y cargar las BBDD con las que se ha de trabajar.
• Colaborar con el administrador del S.O. en las tareas de ubicación, dimensionado y control de los archivos y espacios de disco ocupados por el SGBD.
• Colaborar en las tareas de formación de usuarios.
• Establecer estándares de uso, políticas de acceso y protocolos de trabajo diario para los usuarios de la BD.
• Suministrar la información necesaria sobre la BD a los equipos de análisis y programación de aplicaciones.
• Efectuar tareas de explotación como:
– Vigilar el trabajo diario colaborando en la información y resolución de las dudas
de los usuarios de la BD.
– Controlar en tiempo real los accesos, tasas de uso, cargas en los servidores, anomalías,
etcétera.
– Llegado el caso, reorganizar la BD.
– Efectuar las copias de seguridad periódicas de la BD.

– Restaurar la BD después de un incidente material a partir de las copias de seguridad.
– Estudiar las auditorías del sistema para detectar anomalías, intentos de violación de la seguridad, etcétera.
– Ajustar y optimizar la BD mediante el ajuste de sus parámetros, y con ayuda de las herramientas de monitorización y de las estadísticas del sistema. En su gestión diaria, el DBA suele utilizar una serie de herramientas de administración de la BD.
Con el paso del tiempo, estas herramientas han adquirido sofisticadas prestaciones y facilitan en gran medida la realización de trabajos que, hasta no hace demasiado, requerían de arduos esfuerzos por parte de los administradores.





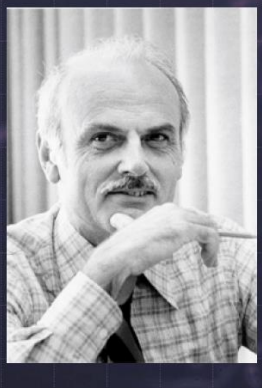
Esquema conceptual

- Válido para cualquier tipo de SGBD
- Primer esquema de la base de datos (determina al resto)
- Realizado por el/la analista (o diseñador)
- Se obtiene de la información obtenida durante la fase de análisis
- Saltarse este esquema es alejarse en exceso del problema real

Modelos lógicos

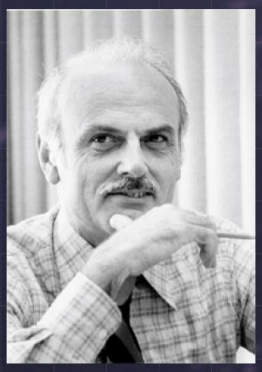
- Definen tipos de bases de datos
- Se acercan más al ordenador
- Son la base del diseño de la base de datos
- Al elegir un modelo lógico estamos tomando una decisión fundamental sobre la manera de trabajar con los datos

Aparición del modelo relacional



- *1970: "A Relational Model of data for Large Shared Data Banks"*
  "Un modelo relacional de datos para grandes bancos de datos compartidos"
- Influencia: Teoría de conjuntos (Cantor y Childs)
- Primeros intentos: IBM
- Primera BDR comercial: Oracle

Objetivos de Codd



- Independencia física.
- Independencia lógica.
- Flexibilidad.
- Uniformidad.
- Sencillez.

[2] Estructura de las Bases de Datos Relacionales

Unidad 3) Modelo Relacional
Gestión de Bases de Datos, ciclo de ASIR

La tabla o relación

- La base del modelo relacional es lo que se conoce como **tabla** (Codd lo llamó **relación**).
- El concepto de relación de **Codd** no tiene nada (o poco) que ver con el concepto de relación de **Chen** (modelo E/R)
- Las tablas constan de:
  - **Atributos**
  - **Tuplas** (**filas**)

Estructura del MR. La tabla o relación

atributo 1	atributo 2	atributo 3	....	atributo n	
valor 1,1	valor 1,2	valor 1,3	....	valor 1,n	← tupla 1
valor 2,1	valor 2,2	valor 2,3	....	valor 2,n	← tupla 2
.....	.....	.....	....	.....	....
valor m,1	valor m,2	valor m,3	....	valor m,n	← tupla m

Conceptos

- Fila=Tupla
- Columna=Atributo
- Reglas:
  - No hay dos tuplas iguales
  - Tupla=Datos de un elemento del mundo real
- Dominio
  - Valores posibles que puede tomar un atributo
- Grado=nº columnas
- Cardinalidad=nº de filas

Equivalencias

- Tabla=Archivo=Relación
- Fila=Tupla=Registro
- Columna=Atributo=Campo
- Grado=nº columnas
- Cardinalidad=nº de filas

Propiedades de las tablas

- Cada tabla debe tener un nombre distinto
- Cada atributo de la tabla toma un solo valor en cada fila
- Cada atributo tiene un nombre distinto en cada tabla (aunque puede coincidir en tablas distintas)
- Cada fila es única (no hay tuplas duplicadas)
- El orden de los atributos no importa
- El orden de las filas no importa

Tipos de tablas

- Persistentes. Sólo pueden ser borradas por los usuarios
  - Bases.
  - Vistas.
  - Instantáneas o vistas materializadas
- Temporales.

Valores nulos

- El modelo relacional usa un valor especial: Nulo (NULL)
- Indica que un atributo está vacío en una determinada fila
- No es igual a cero, ni es un texto sin valor. Es simplemente la ausencia de valor
- Su gestión es primordial en el modelo relacional
- Si alguien no tiene teléfono, la columna teléfono valdrá NULL para ese alguien

Valores nulos. Tabla de la verdad

Operación	Resultado
Verdadero AND Nulo	Nulo
Verdadero OR Nulo	Verdadero
Falso AND Nulo	Falso
Falso OR Nulo	Nulo
NOT Nulo	Nulo

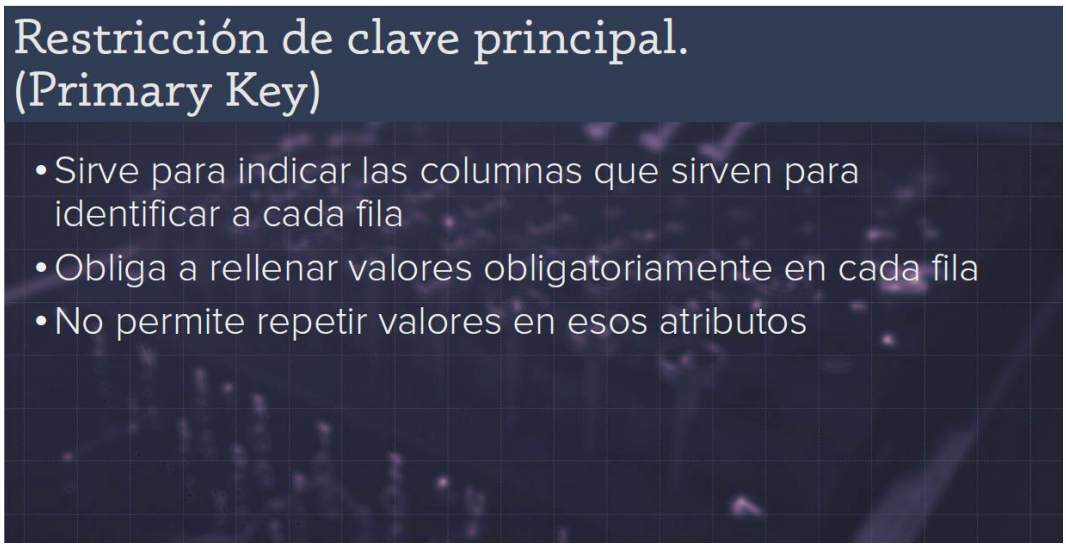
[3] Restricciones

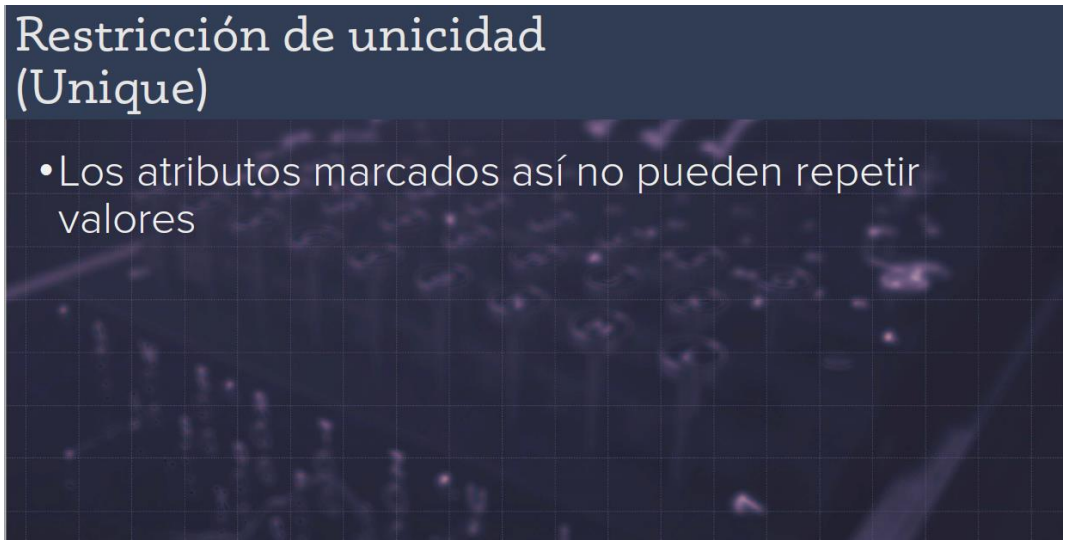
Inherentes

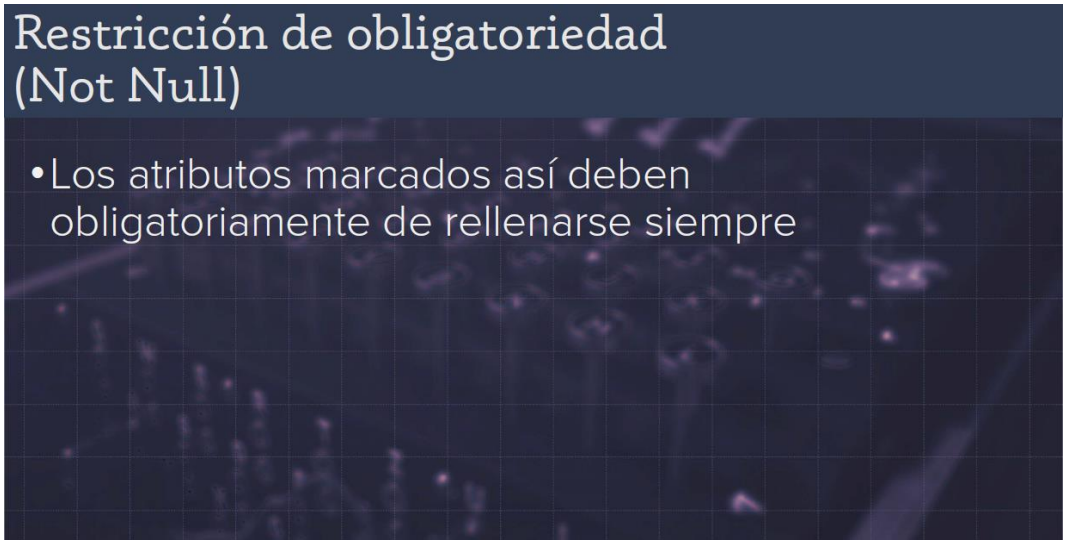
- Cada tabla tiene un nombre distinto
- Cada atributo de la tabla toma un solo valor en cada fila
- Cada atributo tiene un nombre distinto en cada tabla (aunque puede coincidir en tablas distintas)
- Cada fila es única (no hay tuplas duplicadas)
- El orden de los atributos no importa
- El orden de las filas no importa

Semánticas

- Clave principal
- Unicidad
- Obligatoriedad
- Clave candidata
- Integridad referencial
- Validación
- Triggers


Restricción de clave principal.
(Primary Key)
• Sirve para indicar las columnas que sirven para identificar a cada fila
• Obliga a rellenar valores obligatoriamente en cada fila
• No permite repetir valores en esos atributos


Restricción de unicidad
(Unique)
• Los atributos marcados así no pueden repetir valores


Restricción de obligatoriedad
(Not Null)
• Los atributos marcados así deben obligatoriamente de rellenarse siempre

Restricción de clave alternativa (Alternate Key)

- En casi ningún gestor de base de datos existe como tal
- Los atributos que son claves alternativas se marcan con restricción Unique y Not Null

Restricción de integridad referencial (Foreign Key)

- Marca las claves secundarias
- Restringe los valores de modo que las claves secundarias no pueden tomar valores que no existan en la clave principal
- No podremos, p. ej., marcar una nota a un nº de alumno que no exista

Restricción de integridad referencial (Foreign Key). Problemas

- Si borramos o modificamos claves principales, sus secundarias relacionadas quedarían sin cumplir la restricción
- Para gestionar este problema las bases de datos tienen políticas especiales al eliminar y al insertar (las políticas pueden ser distintas)

Restricción de integridad referencial (Foreign Key). Problemas

- Políticas:
  - **No hacer nada (Do Nothing).** No podremos eliminar y/o modificar la clave principal
  - **Cascada (Cascade).** Se modifican o eliminan todas las claves secundarias relacionadas
  - **Poner nulos (Set Null).** Las claves secundarias se ponen con valor nulo
  - **Valor por defecto (Set Default).** Las claves secundarias se marcan con un valor por defecto

Restricción de integridad referencial (Foreign Key). Problemas

Alumno	Cod Alumno
Juanjo	1
Ana	2
Sonia	3

Cod alumno	Cod asignatura	Nota
1	1	9
1	2	9
1	3	7
2	1	8
2	2	6
2	3	5
3	2	7
3	3	1

Restricción de integridad referencial (Foreign Key). Problemas

Actualizar en cascada

Alumno	Cod Alumno
Juanjo	7
Ana	2
Sonia	3

Cod alumno	Cod asignatura	Nota
1	1	9
1	2	9
1	3	7
2	1	8
2	2	6
2	3	5
3	2	7
3	3	1

Restricción de integridad referencial (Foreign Key). Problemas

Actualizar en cascada

Alumno	Cod Alumno
Juanjo	7
Ana	2
Sonia	3

Cod alumno	Cod asignatura	Nota
7	1	9
7	2	9
7	3	7
2	1	8
2	2	6
2	3	5
3	2	7
3	3	1

Restricción de integridad referencial (Foreign Key). Problemas

Eliminar en cascada

Alumno	Cod Alumno
Juanjo	1
Ana	2
Sonia	3

Cod alumno	Cod asignatura	Nota
1	1	9
1	2	9
1	3	7
2	1	8
2	2	6
2	3	5
3	2	7
3	3	1

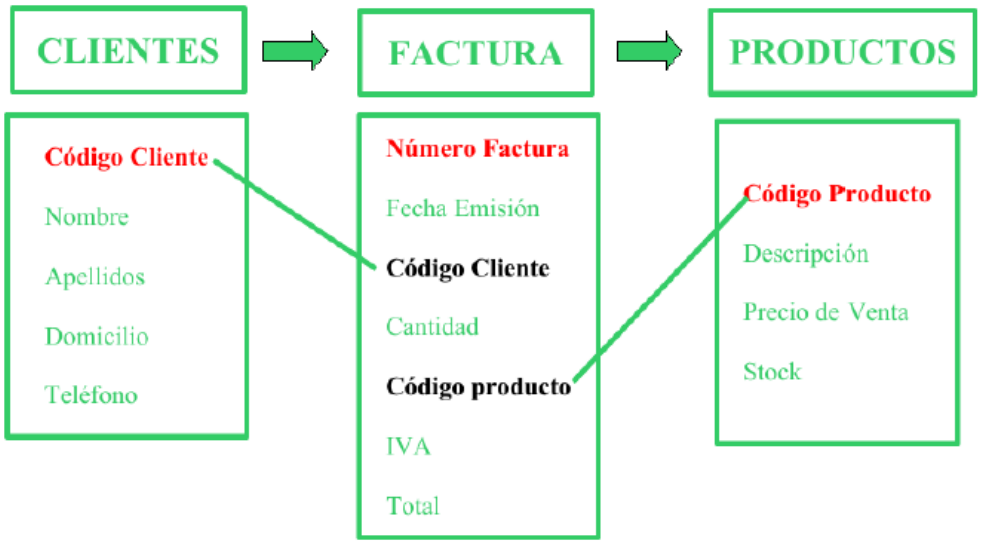
Restricción de integridad referencial (Foreign Key). Problemas

Alumno	Cod Alumno
~~Juanjo~~	~~1~~
Ana	2
Sonia	3

Cod alumno	Cod asignatura	Nota
~~1~~	~~1~~	~~9~~
~~1~~	~~2~~	~~9~~
~~1~~	~~3~~	~~7~~
2	1	8
2	2	6
2	3	5
3	2	7
3	3	1


Restricción de integridad referencial
(Foreign Key). Problemas
Alumno
Cod Alumno
Ana
2
Sonia
3
Cod alumno
Cod asignatura
Nota
2
1
8
2
2
6
2
3
5
3
2
7
3
3
1

EJEMPLO



CLAVES PRIMARIAS:

Código Cliente es la clave primaria de CLIENTES. A cada cliente se le asocia un código y a cada código le corresponde un cliente.
Número Factura es clave primaria de FACTURAS.
Código Producto es clave primaria de PRODUCTOS.

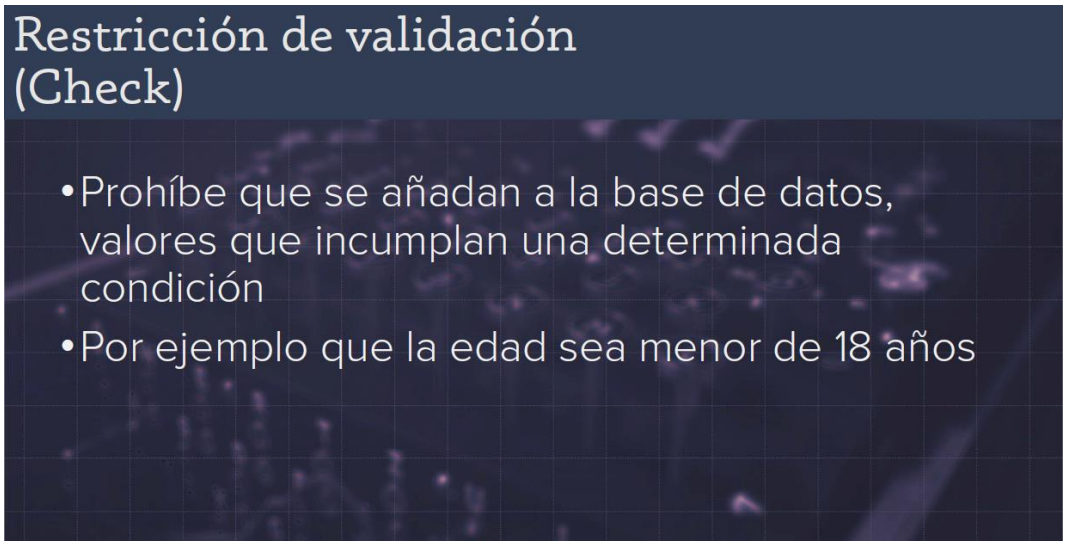
CLAVES FORÁNEAS:

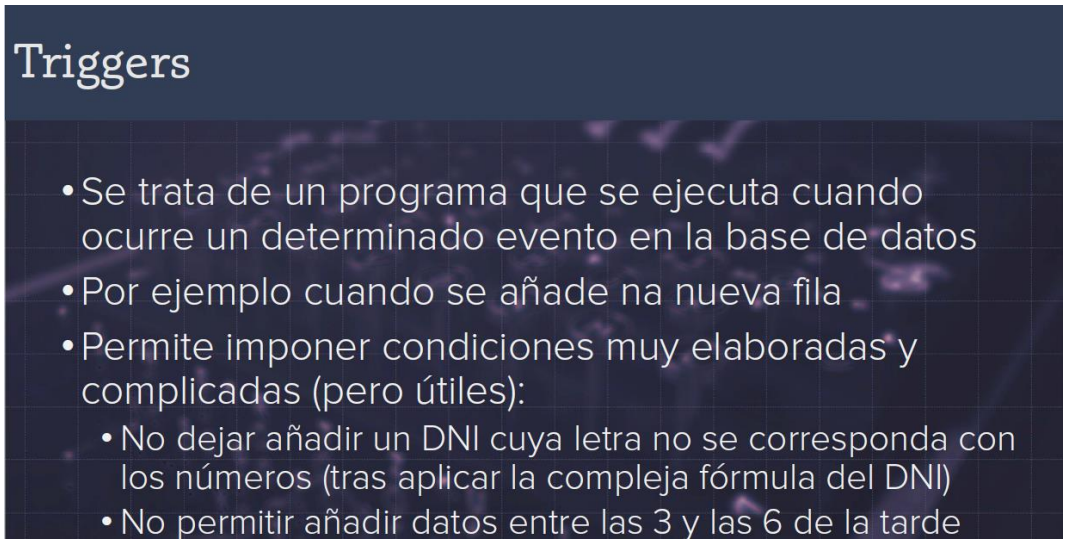
En FACTURAS, son claves foráneas Código Cliente y Código Producto. CLIENTES se relaciona con FACTURAS a través del Código Cliente que figura en ambas tablas y con PRODUCTOS mediante el Código Producto

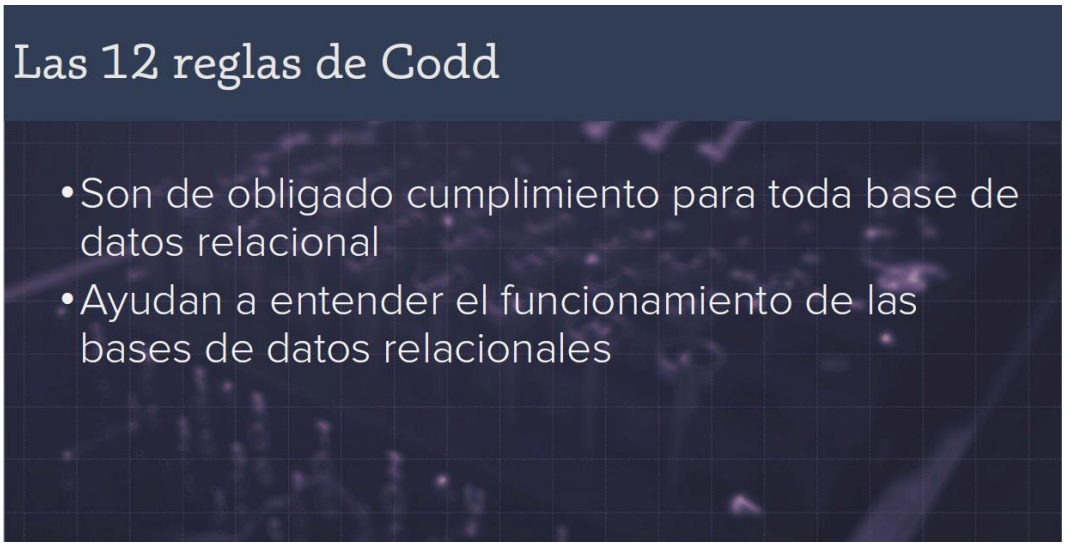
RESTRICCIONES DE INTEGRIDAD REFERENCIAL

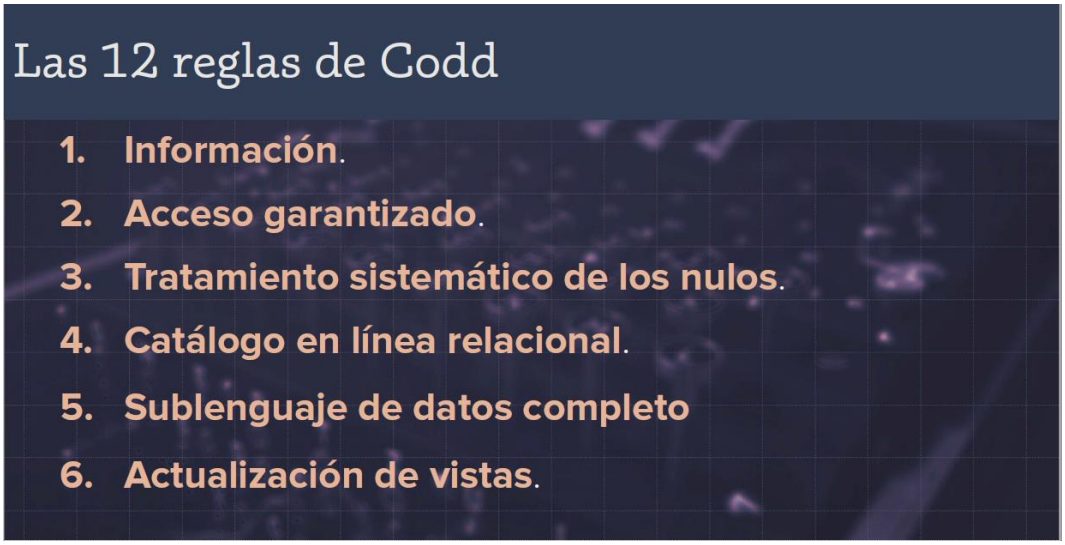
Código Cliente en Facturas debe cumplir que exista en Clientes y que sea clave primaria
Código Producto en Facturas debe cumplir que exista en Productos y que sea clave primaria
Retomando la Definición de Base de Datos, la cual señala que ésta ". . . es un conjunto de datos relacionados entre sí
y que tienen un significado implícito", se observa en la imagen que los datos de las tablas se relacionan a través de las claves

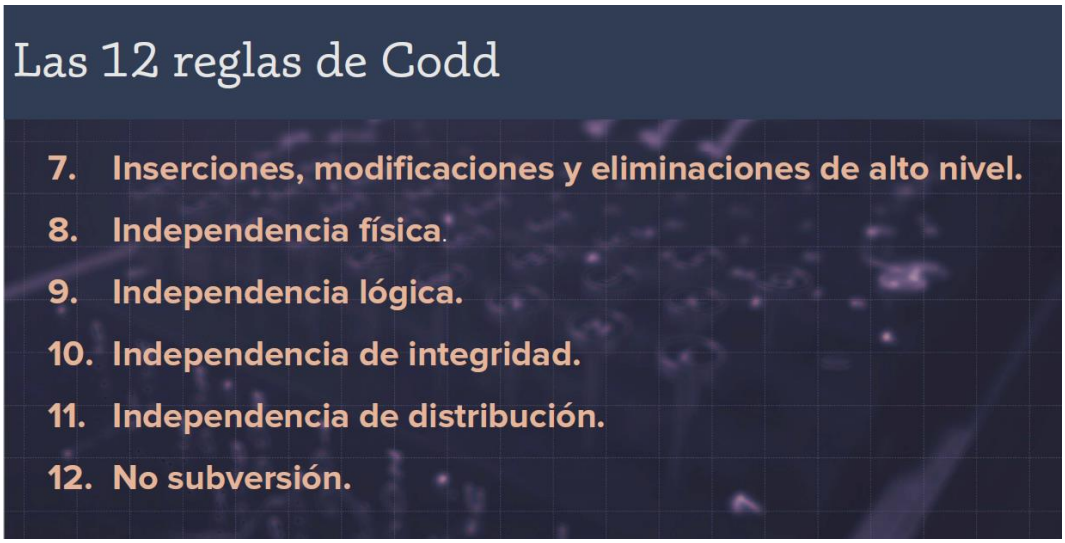
y que éstos tienen el significado implícito que se les atribuye en dicho contexto. Así, por ejemplo, el significado del dato Nombre se refiere al del CLIENTE, el de Fecha emisión a la de la FACTURAS y el de Descripción a la del PRODUCTO.






Las 12 reglas de Codd
•Son de obligado cumplimiento para toda base de datos relacional
•Ayudan a entender el funcionamiento de las bases de datos relacionales


Las 12 reglas de Codd
1. Información.
2. Acceso garantizado.
3. Tratamiento sistemático de los nulos.
4. Catálogo en línea relacional.
5. Sublenguaje de datos completo
6. Actualización de vistas.



En 1985, Codd publica sus famosas doce reglas analizando algunos de los productos comerciales
de la época, que debe cumplir cualquier base de datos para ser considerada relacional:

Regla de información. Toda informacion de una base de datos relacional esta representada mediante valores en tablas.

Regla de acceso garantizado. Se garantiza que todos los datos de una base relacional son lógicamente accesibles a través de una combinación de nombre de tabla, valor de clave primaria y nombre de columna.

Tratamiento sistematico de valores nulos. Los valores nulos se soportan en los SGBD para representar la falta de información de un modo sistemático e independiente de los tipos de datos.

Catálogo en línea dinámico basado en el modelo relacional. La descripción de la base de datos se representa en el ambito logico de la misma forma que los datos ordinarios, de modo que los usuarios autorizados pueden acceder a ellos utilizando el mismo lenguaje relacional.

Regla de sublenguaje completo de datos. Un sistema relacional puede soportar varios lenguajes y varios modos de uso terminal. Sin embargo, debe haber al menos un lenguaje cuyas sentencias se puedan expresar mediante alguna sintaxis bien definida, como cadenas de caracteres, y que ofrezca completamente todos los puntos siguientes:
– Definición de datos.
– Definición de vistas.
– Manipulación de datos (interactiva y por programa).
– Restricciones de integridad.
– Autorización.
– Gestión de transacciones (comienzo, confirmación y vuelta atrás).

Regla de actualización de vista. Todas las vistas, que sean teóricamente actualizables, son tambien actualizables por el sistema.

Inserción, actualización y supresión de alto nivel. La capacidad de manejar una relación de base de datos o una relación derivada como un único operando se aplica no solamente a la recuperación de datos, sino también a la inserción, actualización y supresión de los datos.

Independencia fisica de los datos. Los cambios que se efectúan tanto en la representación del almacenamiento, como en los métodos de acceso no deben afectar ni a los programas de aplicación ni a las actividades con los datos.

Independencia lógica de los datos. Del mismo modo, los cambios que se efectúen sobre las tablas de la base de datos no modifican ni a los programas ni a las actividades con los datos.

Independencia de la integridad. Las restricciones de integridad especificas de una base de datos relacional deben ser definidas mediante el sublenguaje de datos relacional y almacenarse en el catálogo de la base de datos.

Independencia de la distribución. Un SGBD es independiente de la distribución.

Regla de no subversión.
Si un SGBDR tiene un lenguaje de bajo nivel (una fila cada vez) no se puede utilizar para destruir o evitar las reglas de integridad o las restricciones expresadas en el lenguaje relacional de alto nivel (varias filas al mismo tiempo).
El modelo relacional propone una representación de la información que:
Origine esquemas que representen fielmente la información, los objetos y las relaciones existentes entre ellos forman el dominio del problema.
Sea fácilmente entendida por los usuarios.
Sea posible ampliar el esquema de la base de datos sin modificar la estructura lógica existente y los programas de aplicación.
Permita la máxima flexibilidad en la formulación de los interrogantes sobre la información mantenida en la base de datos.