



Bases de Datos

Tema 4: El modelo de datos relacional

Dpto. de Ingeniería Informática

Contexto

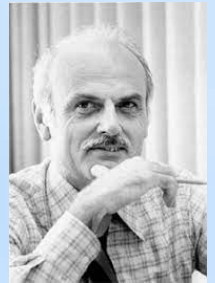
- Asignatura “Bases de datos”, Grado en Ingeniería Informática - UCA
- Competencias
 - CB2 Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio
- Resultados del aprendizaje
 - R1 Conocer la importancia de la información y la necesidad de gestionarla de forma eficiente
 - R2 Saber utilizar los diferentes modelos de datos, sus componentes y la importancia en el diseño de una base de datos
 - R5 Saber realizar un diseño conceptual y lógico para una base de datos
- Tiempo estimado: 2 horas

Contenidos

- Objetivos
- Introducción
- El MD relacional
- Doce reglas de Codd
- Lenguajes
- Referencias

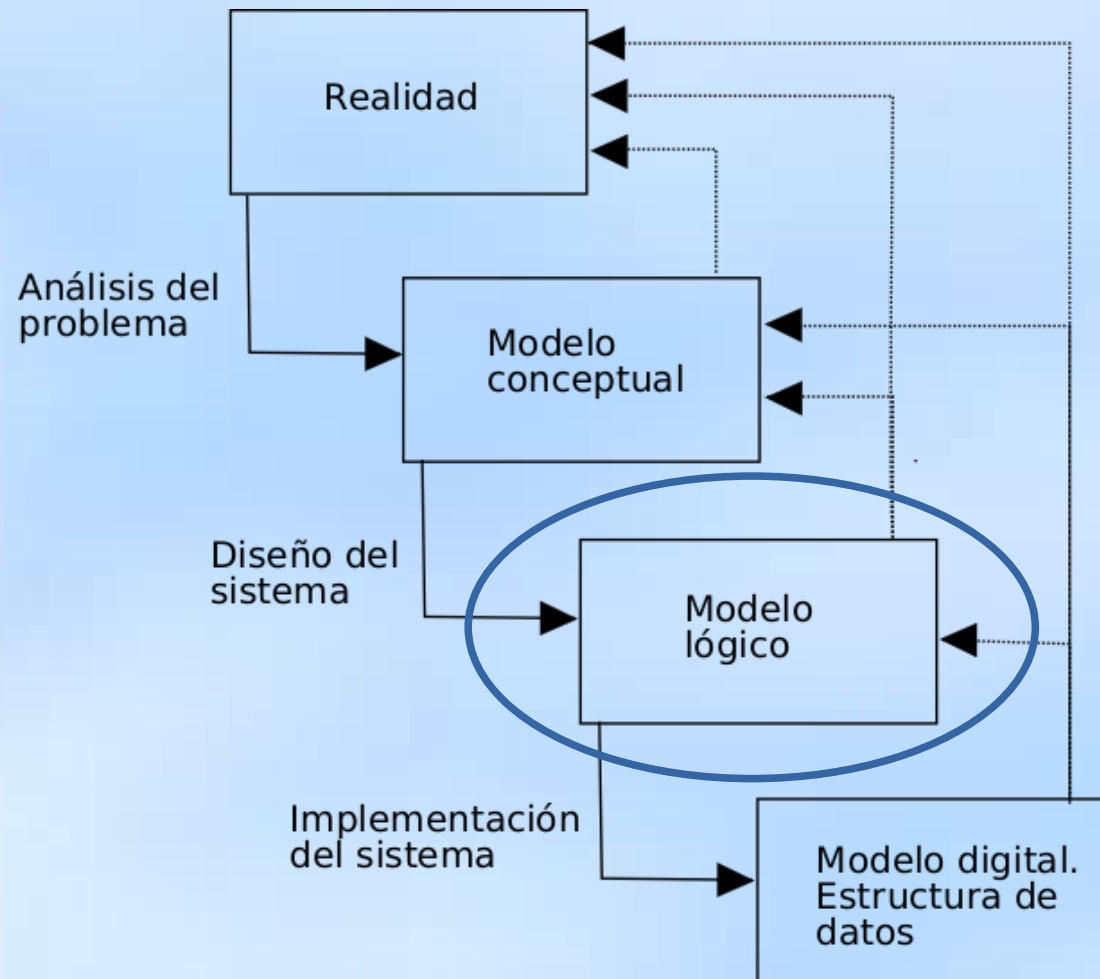
Introducción

- El modelo relacional es un modelo de datos global, llamados convencionales o lógicos
- Propuesto por Edgar F. Codd en 1970
- Modelo sencillo y fácil de entender, formado solamente por el concepto matemático de relación
 - Una relación es una tabla bidimensional donde los datos se muestran utilizando una estructura de registro
- Este modelo se implementa en sistemas de gestión de Bases de Datos Relacionales DB-R
 - Soporta el lenguaje de programación SQL



Introducción

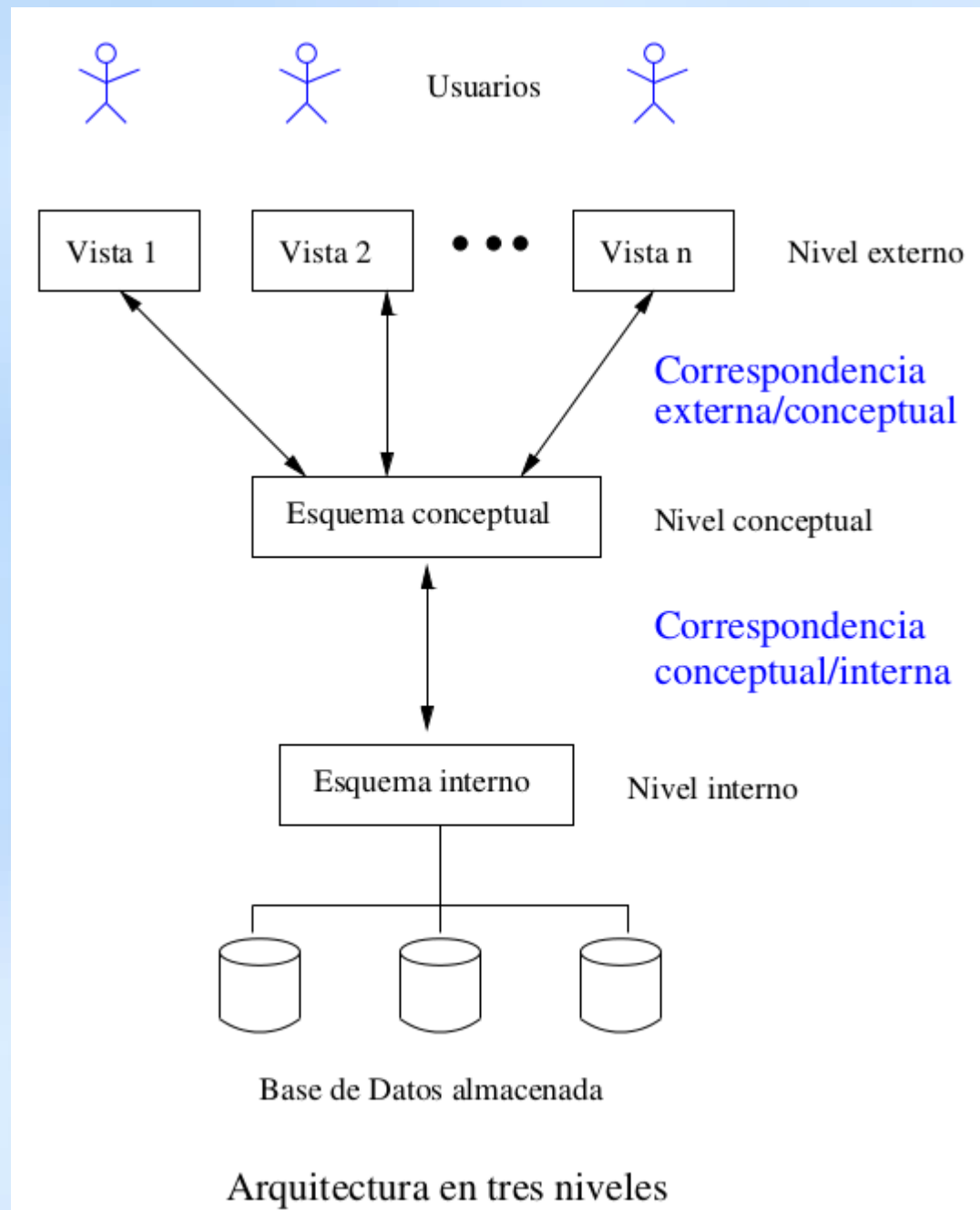
- <https://www.um.es/geograf/sigmur/temariohtml/cascada.png>



Introducción

- Objetivos:
 - **Independencia física:** modificación del almacenamiento físico sin modificación de aplicaciones
 - **Independencia lógica:** manipulación lógica de la BD sin modificación de las aplicaciones
 - **Flexibilidad:** ofrecer los datos a las aplicaciones como se requieran
 - **Uniformidad:** presentación de la información en tablas
 - **Sencillez:** manipulación con lenguajes sencillos
- Se divide en dos partes:
 - *Estática:* está formada por el concepto de tabla o relación y los términos asociados a este concepto
 - *Dinámica:* formada por un conjunto de operadores, que se aplican a las relaciones, conformando el álgebra y el cálculo relacional

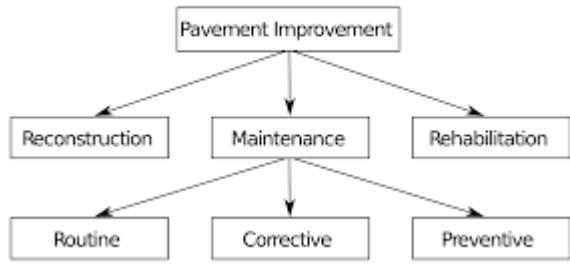




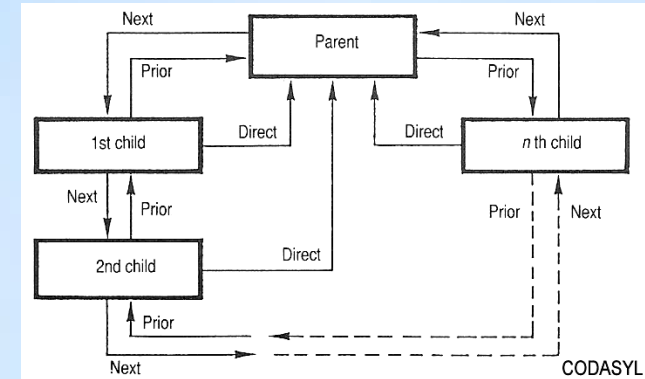
Introducción

- Una BD garantiza que los datos respetarán las restricciones del mundo a modelar mediante reglas:
 - *Generales* para toda BD: claves primarias y foráneas, ...
 - *Específicas* para cada BD: una persona tiene una sola madre, todo vehículo tiene mínimo un propietario, ...
- Diseño basado en la teoría de la normalización, eliminando dependencias entre atributos (tema 6)
 - Las tres FN de Codd (FN1, FN2 y FN3)
 - FN de Boyce-Codd
 - Formas normales de Fagin (FN4, FN5)
 - ...

- Los productos comerciales:
 - Primeros:
 - QBE: 1978, lenguaje de acceso relacional a ficheros VSAM de IBM. No era un verdadero SGBDR
 - Oracle: 1979, primer SGBDR comercial, con lenguaje SQL para definición y manipulación de datos
 - Ingres: 1980, origen en el prototipo de igual nombre de la Univ. de Berkeley, con lenguaje QUEL basado en cálculo relacional. Más utilizado por las universidades, en especial las de EE.UU.
 - “El resto” relacionales
 - Soluciones “de entorno”: MS-Server, IBM-DB2, ...
 - MySQL: para el Internet “masivo”



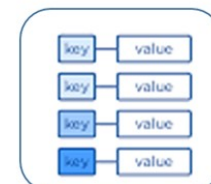
Introducción



- Etapas:
 - **Prerrelacional**: 1ª generación de BD. Los SGBD se soportan con los modelos de datos Codasyl y Jerárquico
 - **Relacional**: 2ª generación de BD. Se van consolidando los sistemas relacionales y van desplazando a los sistemas de 1ª generación
 - **Postrelacional**: 3ª generación de BD. Aparición de otros modelos de datos: basados en objetos, en grafos, NoSQL, ...



Document Store



Key-Value Store



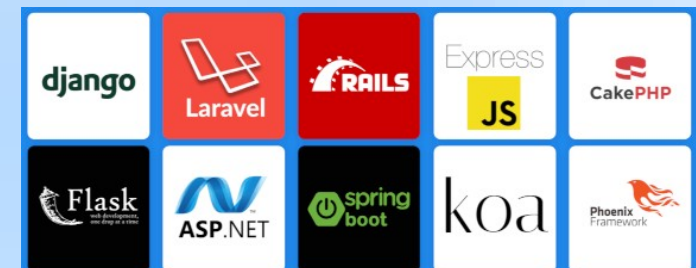
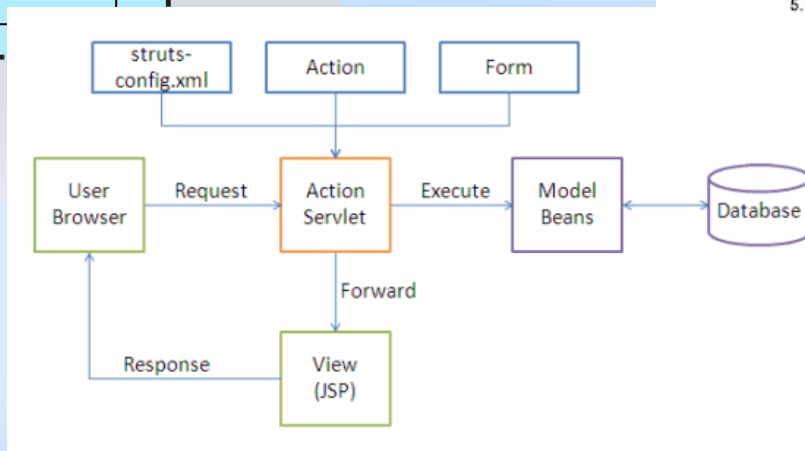
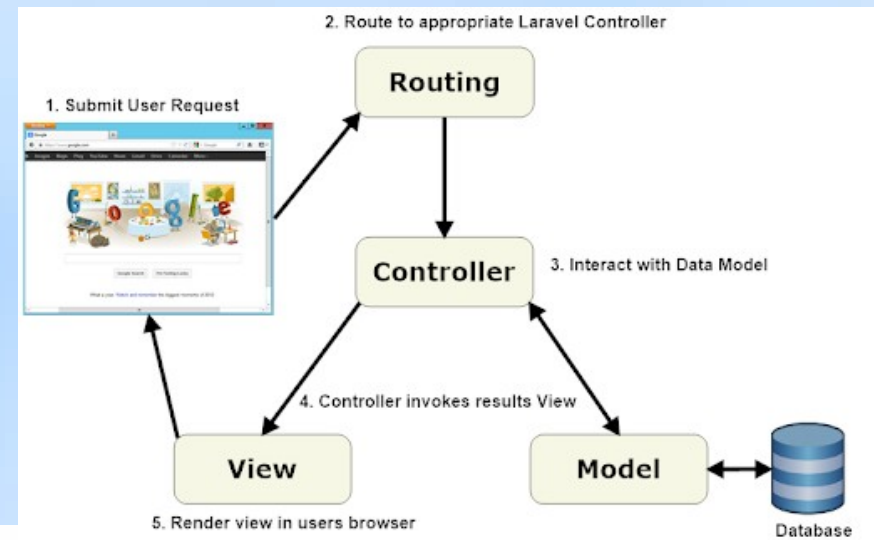
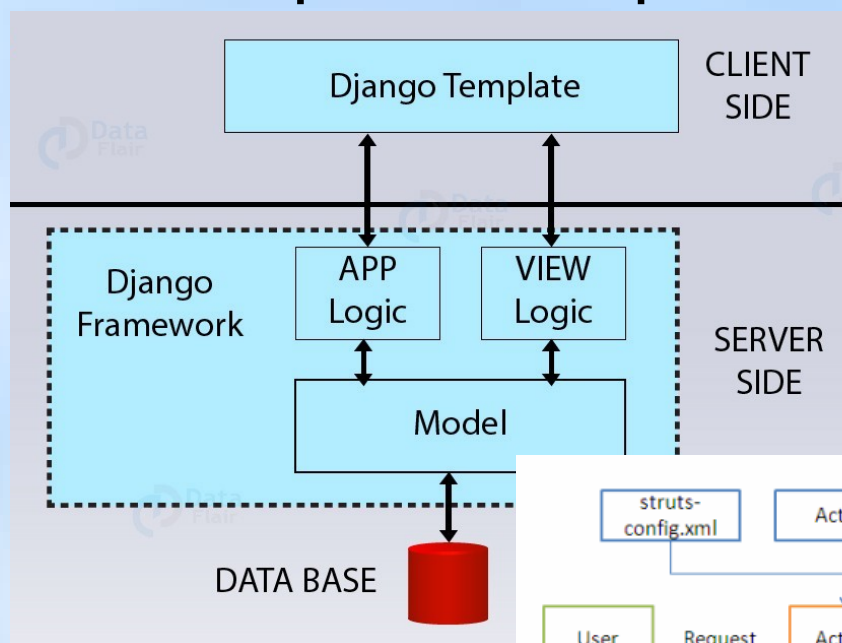
Wide-Column Store



Graph Store

Introducción

- Hoy en día se usan SGBD relacionales
 - Muchas veces con frameworks de programación que los mapean



Estructura

- Base de Datos: conjunto de tablas o relaciones
- Tres partes:
 - **Estructura:** conjunto de relaciones que forman la BD
 - **Integridad:** conjunto de restricciones de la BD
 - **Manipulación:** conjunto de operaciones sobre la información contenida en la BD (CRUD)
 - introducción
 - extracción (consultas/informes)
 - actualización
 - borrado de información



Estructura

- Terminología en BD (~equivalencia)

Académico	Equivalencia “comercial”
Relación	Tabla
Tupla	Fila o registro
Cardinalidad	Número de filas
Atributo	Columna o campo
Grado	Número de columnas
Clave primaria	Identificador único
Dominio	Conjunto de valores

Tablas

- La estructura de las tablas (columnas) son estáticas
- Esquema con atributos y restricciones
 - El diseño de una BD relacional: conjunto de esquemas de relaciones (*esquema de la BD relacional*)
 - Los nombres de tablas son únicos en el esquema
 - Debe cambiar poco en el tiempo
 - Principalmente creciendo
- Instancia: datos (tuplas) en un momento concreto conforme a ese esquema
 - Son los que interesan a los usuarios / aplicaciones
 - Evolucionan en el tiempo



Estructura

- Ejemplo de la base de datos *Editorial*:
 - Esquema de la tabla *Autores*

Nombre	Tipo de datos	Descripción
<i>Código</i>	Entero (único, no nulo)	Código del autor
<i>Nombre</i>	Cadena de caracteres	Nombre del autor
<i>Apellido</i>	Cadena de caracteres	Apellidos del autor

Estructura

- Ejemplo de la base de datos *Editorial*:
 - Identificadores de campos en la cabecera
 - Instancia de la tabla *Autores*

Código	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky

Atributos

- Los atributos de una relación se corresponden con los nombres de sus columnas
- Toda relación está formada por varios atributos que describen el significado de cada entrada de datos en sus columnas
 - Ej: en la relación *Autores*, la columna con el atributo *Código* contiene los códigos numérico de los autores cuyos datos están en dicha relación
- Los nombres de los atributos deben ser únicos en una relación, siendo inequívoca la nomenclatura:

nombre_tabla.nombre_atributo

Esquemas

- El esquema de una relación está formado por el nombre y el conjunto de atributos
- Se utiliza la siguiente nomenclatura: nombre de la relación seguida, entre paréntesis, de los atributos que la forman
- Ej: el esquema de la relación *Autores*:
Autores (Código, Nombre, Apellido)
- Pregunta: ¿Hay orden en los atributos de una relación?

Esquemas

- ¿Hay orden en los atributos de una relación?
 - Algunos autores defienden que no: no es necesario añadir un orden que complique el modelo. Por lo tanto, los atributos serían un conjunto
 - Otros defienden que definir un orden concreto no complica en exceso el modelo y facilita significativamente la nomenclatura
 - Ej: INSERT (8, Richard, Stallman) INTO Autores → Código es 8, Nombre es Richard, Apellido es Stallman
 - Usaremos esta última aproximación

Dominios

- Dominio: conjunto de valores todos del mismo tipo
 - Cada atributo está definido sobre un único dominio
- Se indican en la definición de la BD
- Los dominios restringen las inserciones, comparaciones, las operaciones y las expresiones que pueden realizarse con los atributos
- Cada componente de cada tupla debe ser atómico y de algún tipo “elemental” de datos
 - Representan la menor unidad semántica de información

Dominios

- En esta asignatura consideraremos los siguientes tipos elementales:
 - Booleano
 - Entero
 - Decimal
 - Cadena de caracteres
 - Hora
 - Fecha (incluye hora)
 - BLOB (Binary Large OBjects, objetos binarios grandes)

Dominios

- Los SGBD suelen ofrecer un conjunto de dominios “base”: booleano, entero, decimal, VARCHAR, fecha, BLOB
 - <https://dev.mysql.com/doc/refman/5.7/en/data-types.html>
 - <https://www.postgresql.org/docs/13/datatype.html>
 - https://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm#CNCPT012

Dominios

- Los SGBD suelen ofrecer un conjunto de dominios “base”: booleano, entero, decimal, VARCHAR, fecha, BLOB
 -
- A veces permiten definir dominios más finos
 - Ej: VARCHAR(8) para DNI, provincias de España
 - Importante: que soporten las operaciones que necesitan
 - Puede evitar errores en otros niveles
 - Algunos comprueban que las operaciones no violen las definiciones de los atributos implicados
 - Pero no todos lo hacen → tener en cuenta estas restricciones a la hora de manipular los datos de la BD (cuidado con hacking)

Tuplas e instancias

- Las tuplas son los registros (“filas”) de una relación
 - Suelen tener un valor para cada atributo de la relación
- Todas las tuplas de una misma relación son diferentes
- Cada tupla debe poder identificarse con valor de un campo *identificador único*, que ha de ser distinto para cada una de ellas
 - Este identificador único puede estar formado por un único atributo o un conjunto de ellos

Relaciones

- Desde el punto de vista matemático, una relación es un subconjunto del producto cartesiano de un listado de dominios, no necesariamente todos distintos

$$\times_{i=1}^n D_i$$

- Toda relación tiene un nombre, un esquema y una instancia
- El esquema está formado por una serie fija de atributos que se representan por la pareja atributo-dominio

$$\langle (A_1 : D_1), (A_2 : D_2), \dots, (A_n : D_n) \rangle$$

- donde n es el grado o número de atributos de la relación

Relaciones

- Cada tupla está compuesta por una serie de valores

$$\langle v_{i1}, v_{i2}, \dots, v_{in} \rangle$$

$$(i = 1, 2, \dots, m)$$

donde m es la cardinalidad o número de tuplas de la relación en un instante dado

- Ejemplo de tupla:

$$\langle 2, \text{Lawrence}, \text{Lessig} \rangle$$

Relaciones

- La instancia es el conjunto de tuplas en un momento dado
 - Ejemplo de instancia:
 $\{ \langle 1, \text{Neal}, \text{Stephenson} \rangle, \langle 2, \text{Lawrence}, \text{Lessig} \rangle, \langle 3, \text{Noam}, \text{Chomsky} \rangle \}$
 - O igual:
 $\{ \langle 2, \text{Lawrence}, \text{Lessig} \rangle, \langle 1, \text{Neal}, \text{Stephenson} \rangle, \langle 3, \text{Noam}, \text{Chomsky} \rangle \}$

Propiedades

- Como consecuencia de definición de relación:
 - No existen tuplas repetidas: todas las tuplas de una relación son diferentes
 - Las tuplas no están ordenadas: la instancia de una relación es un conjunto y los elementos de un conjunto no tienen orden
 - Los atributos sí están ordenados: el esquema de una relación es una serie, luego tiene orden
 - ...

Propiedades

- Como consecuencia de definición de relación:
 - Los valores de los atributos son atómicos: los atributos tienen dominios simples
 - ¿Qué hago si tengo un atributo compuesto?

Propiedades

- Como consecuencia de definición de relación:
 - Los valores de los atributos son atómicos: los atributos tienen dominios simples
 - ¿Qué hago si tengo un atributo compuesto?
 - Sólo almaceno los componentes
 - ¿Qué hago si tengo un atributo multivaluado?

Propiedades

- Como consecuencia de definición de relación:
 - Los valores de los atributos son atómicos: los atributos tienen dominios simples
 - ¿Qué hago si tengo un atributo compuesto?
 - Sólo almaceno los componentes
 - ¿Qué hago si tengo un atributo multivaluado?
 - Con carácter general dará lugar a una nueva relación
 - Según el caso puedo repetir el atributo. Ej con teléfonos:
 - Profesor (DNle, nombre, teléfono_desp, teléfono_dpto)
 - O puedo repetir tuplas. Ej: { <1, Pepe, 4565>, <1, Pepe, 4888> }

Propiedades

- Como consecuencia de definición de relación:
 - ¿Puede haber un número máximo de tuplas en una relación?

Propiedades

- Como consecuencia de definición de relación:
 - ¿Puede haber un número máximo de tuplas en una relación?
 - Sí, por ejemplo una relación con las provincias y comunidades autónomas donde tenemos centros:
Centros(Provincia, CA)

Propiedades

- Como consecuencia de definición de relación:
 - ¿Puede haber un número máximo de tuplas en una relación?
 - Sí, por ejemplo una relación con las provincias y comunidades autónomas donde tenemos centros:
Centros(Provincia, CA)
 - Importante tenerlo en cuenta, porque puede ser señal de un diseño erróneo (depende del problema concreto a modelar)
 - La relación anterior no permitiría abrir dos centros en la misma provincia

Tipos de relaciones

- Tipos de relaciones más habituales (I):
 - **Relación base**: relación autónoma, se crea explícitamente y su esquema e instancia se almacenan en la BD
 - **Vista**: son relaciones virtuales con nombre pero sin datos, su nombre está asociado a una consulta
 - Nivel externo de la arquitectura ANSI
 - **Instantánea**: relación derivada con nombre (*snapshot*). Se ejecuta su consulta asociada y el resultado se almacena en la BD, refrescándolo regularmente

Tipos de relaciones

- Tipos de relaciones más habituales (y II):
 - **Resultado de consulta**: es una relación resultante de una consulta, sin existencia en la BD
 - Pueden tener o no nombre
 - **Resultado intermedio**: es una relación, normalmente sin nombre, resultado de una expresión anidada dentro de otra
 - **Relación temporal**: una relación con nombre pero que se destruye de forma automática en cualquier momento

Resumen

- Relación: nombre + esquema formado por columnas
- Las relaciones tienen 0+ tuplas (con un valor por columna)
 - Una tabla vacía es una tabla “válida” que almacena información
- Todos los valores de una columna tienen el mismo tipo de datos, pertenecientes a su dominio
- Las tablas se relacionan entre si por los datos que las contienen. El modelo relacional usa:
 - Clave primaria
 - Clave foránea
 - Otras restricciones

Claves primarias

- Toda relación tiene un identificador único para cada una de sus tuplas. Puede ser:
 - Simple/atómico: un solo atributo del esquema
 - Compuesto: varios atributos del esquema
- **Claves candidatas:** conjunto de identificadores válidos de una relación
 - La **clave primaria** (CP) es una de las claves candidatas
- **Claves alternativas:** claves candidatas que no han sido elegidas como CP

Claves primarias

- Clave candidata:
 - Un atributo K , simple o compuesto, de la relación R es clave candidata de R si satisface dos propiedades independientes en el tiempo:
 - **Unicidad**: en cualquier momento dado, no podrán existir dos tuplas en R con el mismo valor de K
 - **Minimalidad**: si K es compuesto, no será posible eliminar ningún componente de K sin destruir la unicidad
 - Toda CP atómica será mínima
 - Pueden existir CP compuestas que sean mínimas y otras que no

Claves primarias

- Aclaración sobre **minimalidad**
 - Quiere decir que “*no le sobran atributos*” a una clave compuesta
 - No quiere decir que haya que ver si tiene más o menos atributos que otras posibles claves primarias
 - Ej: atendiendo a esta instancia, ¿qué claves candidatas tiene la relación?

Código	Nombre	Ciudad	Cod_local_ciudad
1	Pepe	Cádiz	1
2	Pepe	Jerez	1
3	Pepe	Jerez	2

Claves primarias

- Aclaración sobre **minimalidad**
 - Quiere decir que “*no le sobran atributos*” a una clave compuesta
 - No quiere decir que haya que ver si tiene más o menos atributos que otras posibles claves primarias
 - Ej: atendiendo a esta instancia, claves candidatas son {Código} y {Ciudad, Cod_local_ciudad}
 - Las dos opciones son mínimas
 - No sería mínima {Código, nombre} → (superclave)

Código	Nombre	Ciudad	Cod_local_ciudad
1	Pepe	Cádiz	1
2	Pepe	Jerez	1
3	Pepe	Jerez	2

Claves primarias (resumen)

- Toda relación tiene al menos una clave candidata
 - En el “peor” de los casos: todos sus atributos (ALL-KEY)
- El *DBA* elige la CP entre las claves candidatas
 - Normalmente la más simple/corta/manejable
 - Las descartadas quedan anotadas para garantizar su unicidad (integridad) y por futuros cambios de requisitos
- La *CP* suele ser el principal mecanismo de búsqueda de tuplas → eficiencia (req. no funcional)
- La CP puede estar formada por más de un atributo, luego debemos decir el *conjunto de clave primaria*

Claves primarias (resumen)

- Emplear el término *clave/llave* (a secas) para la clave primaria es ambiguo, porque existen varios tipos de claves en BD:
 - clave primaria
 - claves candidatas
 - claves alternativas
 - super-clave: conjunto de atributos que identifican inequívocamente cada tupla, pero no tiene que ser mínimo (eliminándole atributos mantienen la unicidad)
 - claves surrogadas: la vemos en la siguiente traspas
 - claves foráneas: la vemos en 4 traspas ;)

Claves primarias

- Clave Surrogada: campo sin repeticiones fuera del dominio del problema
 - Algunos entornos obligan a usarla como CP
 - Tiene ventajas e inconvenientes: lo trataremos en detalle en el tema 7
 - En el diseño en principio tenemos que atenernos al mundo que modelamos, aunque después a la hora de implementarlo tomemos otras decisiones
 - Es importante saber cómo distingue el cliente las tuplas
 - No debe ser dependiente de tecnologías concretas
 - Por ejemplo, no pensar que son enteros crecientes


Ejercicio

- Elija un dataset (~instancia) de un repositorio de datos abiertos, mejor en formato CSV, XLS u ODS
 - Ejemplos:
 - <https://opendata.ugr.es>
 - <https://opendata.ugr.es/dataset/desglose-de-gastos-2017/resource/74279ee7-0aa6-4fa9-8f44-c5886d77274e>
 - <https://www.juntadeandalucia.es/datosabiertos/portal/>
 - <https://www.juntadeandalucia.es/datosabiertos/portal/dataset/museos-de-andalucia/resource/510ec36a-b150-4f9a-9b66-c3999cc6b34e>
 - <https://figshare.com>
 - <https://zenodo.org>
 - Responda a las siguientes preguntas:
 - ¿Cuáles son los atributos?
 - ¿Se indica el dominio de cada uno?
 - ¿Hay una descripción de cada atributo?
 - ¿Hay una clave primaria?



Claves foráneas

- Las CF permiten interrelacionar relaciones
 - Fundamento de la integridad referencial
- Clave foránea (CF): es un atributo, o una serie de ellos, de una relación R_2 cuyos valores concuerdan con los de la CP de otra relación R_1
 - Ej: relación Clientes y Pedido



Cód_cli	Nombre	País
C1	Elena	España
C2	Cristina	EEUU
C3	Juanca	UEA

Cód_ped	Cód_cli	Coste	Intermediario
1	C3	500€	Jaime
2	C1	3,000€	Iñaki
8	C1	8,000€	Corina

Claves foráneas

- Las CF permiten interrelacionar/conectar relaciones minimizando la redundancia

Cód_ped	Cód_cli	Nombre	País	Coste	Intermediario
1	C3	Juanca	UEA	500€	Jaime
2	C1	Elena	EEUU	3,000€	Iñaki
8	C1	Elena	EEUU	8,000€	Corina

versus

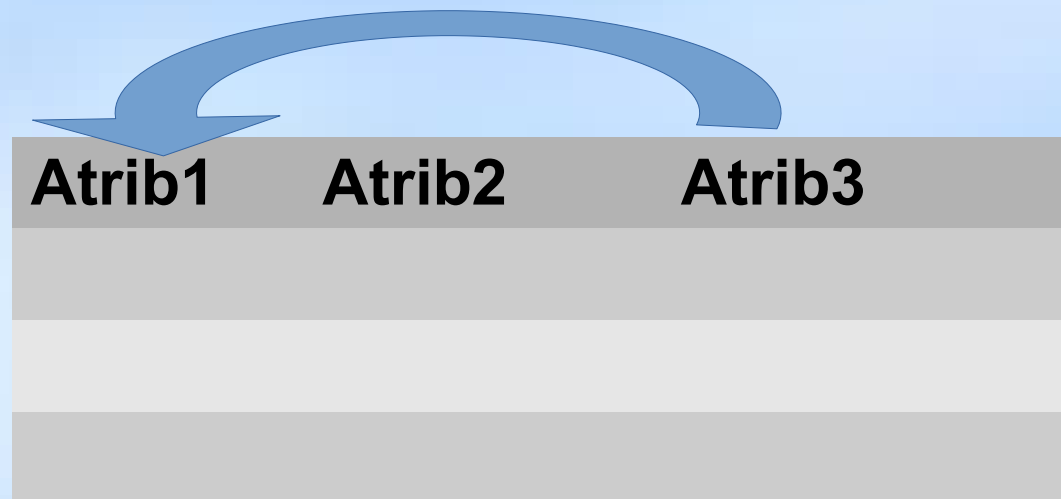
Cód_cli	Nombre	País
C1	Elena	España
C2	Cristina	EEUU
C3	Juanca	UEA

+

Cód_ped	Cód_cli	Coste	Intermediario
1	C3	500€	Jaime
2	C1	3,000€	Iñaki
8	C1	8,000€	Corina

Claves foráneas

- Las dos relaciones R_1 y R_2 no tienen que ser distintas
 - ¿En qué caso podrían no ser distintas?



Claves foráneas

- Las dos relaciones R_1 y R_2 no tienen que ser distintas
 - ¿En qué caso podrían no ser distintas?
 - Relaciones reflexivas. Por ejemplo, alumnos que mentorizan a otros alumnos



Código	Nombre	Mentor
A1	Ana	NULL
A2	Luisa	A1
A3	Antonio	A2

Claves foráneas

- Los atributos CP y la CF deben estar definidos sobre los mismos dominios
- Toda CF debe satisfacer estas dos propiedades independientes en el tiempo:
 - Cada valor de CF es nulo del todo (todos sus componentes son nulos) o bien no nulo del todo
 - Según el problema puede no tener sentido una CF a nulo
 - Ej1: zoo con animales apadrinados y otros no (NULL)
 - Ej2: toda vivienda tiene que tener propietario asociado (NOT NULL)
 - Para cada valor no nulo de CF existe en la relación base R_1 una tupla que tiene dicho valor de CP

Regla de integridad de las relaciones

- El modelo relacional tiene dos reglas de integridad que se cumplen siempre:
 - *Integridad de las relaciones*: ningún componente de la CP de una relación base puede aceptar nulos
 - En una BD relacional, no se registra información de algo que no se puede identificar
 - *Integridad referencial*: la BD no puede contener valores de clave foránea sin concordancia
 - Todo valor de un campo clave foránea de una tupla debe corresponderse con un valor de la clave primaria de la tabla referenciada

Restricciones

- El modelo relacional tiene estructuras y ocurrencias no permitidas
 - Los datos almacenados han de adaptarse a las estructuras impuestas y cumplir las restricciones de usuario para tener ocurrencias válidas del esquema
- Se dividen en:
 - *Restricciones inherentes*: son propias del modelo y quita flexibilidad para representar el mundo real
 - *Restricciones semánticas*: o de usuario, son facilidades que el modelo ofrece para poder representar la semántica del mundo real

Restricciones

- Restricciones inherentes:
 - No son elegidas por el diseñador: son propias del modelo de datos (relacional)
 - Puede ser que alguna me “incomode”
 - Como modelo matemático, las relaciones tienen que cumplir:
 - No hay dos las tuplas iguales
 - Debe existir una CP
 - No hay orden de tuplas
 - Los atributos sólo pueden tomar un único valor sobre el dominio (son atómicos)
 - Regla de integridad de las relaciones

Restricciones

- Restricciones semánticas:
 - Reglas añadidas por el diseñador para reflejar el mundo a modelar
 - Restricciones principales:
 - **Clave primaria** (*primary key*): que exista una CP es restricción inherente, pero la decisión de cuál es semántica
 - **Unicidad** (*unique*): los valores de los atributos no se pueden repetir en una relación
 - **Obligatoriedad** (*not null*): los atributos no pueden ser nulos
 - **Integridad referencial** (*foreign key*): las CF deben concordar con la CP a la que hace referencia o ser nulas
 - **CHECK ...**

Restricciones

- Comprobaciones con CHECK (dependen del SGBD)
 - Tienen un coste computacional asociado
 - MySQL: *The CHECK clause is parsed but ignored by all storage engines*
 - <https://dev.mysql.com/doc/refman/5.7/en/create-table.html>
 - PostgreSQL: *expression producing a Boolean result ... cannot contain subqueries*
 - Ejemplo:

```
CREATE TABLE empleados (  
    id SERIAL PRIMARY KEY,  
    nombre VARCHAR (30), ... ,  
    fecha_nac DATE CHECK (fecha_nac > '1900-01-01'),  
    fecha_contrato DATE CHECK (fecha_nac < fecha_contrato),  
    sueldo numeric CHECK (sueldo BETWEEN 1000 and 10000)  
);
```

Restricciones

- El SGBD vigila que al realizar operaciones que modifiquen la instancia su resultado siga respetando las restricciones del sistema
 - En órdenes de inserción de datos: ¿qué hay que comprobar?

Restricciones

- El SGBD vigila que al realizar operaciones que modifiquen la instancia su resultado siga respetando las restricciones del sistema
 - En órdenes de inserción de datos: comprobar dominios, CP, UNIQUE, NOT NULL y CF
 - ¿Qué pasa si se viola?

Restricciones

- El SGBD vigila que al realizar operaciones que modifiquen la instancia su resultado siga respetando las restricciones del sistema
 - En órdenes de inserción de datos: comprobar dominios, CP, UNIQUE, NOT NULL y CF
 - Solución: rechazar la solicitud (error)
 - En órdenes de borrado de datos: ¿qué hay que comprobar?

Restricciones

- El SGBD vigila que al realizar operaciones que modifiquen la instancia su resultado siga respetando las restricciones del sistema
 - En órdenes de inserción de datos: comprobar dominios, CP, UNIQUE, NOT NULL y CF
 - Solución: rechazar la solicitud (error)
 - En órdenes de borrado de datos: que la CP que se pierda no deje una CF de otra tabla sin emparejar
 - ¿Soluciones?

Restricciones

- El SGBD vigila que al realizar operaciones que modifiquen la instancia su resultado siga respetando las restricciones del sistema
 - En órdenes de inserción de datos: comprobar dominios, CP, UNIQUE, NOT NULL y CF
 - Solución: rechazar la solicitud (error)
 - En órdenes de borrado de datos: que la CP que se pierda no deje una CF de otra tabla sin emparejar
 - Soluciones: rechazar la solicitud o propagar el efecto

Restricciones

- El SGBD vigila que al realizar operaciones que modifiquen la instancia su resultado siga respetando las restricciones del sistema
 - En órdenes de inserción de datos: comprobar dominios, CP, UNIQUE, NOT NULL y CF
 - Solución: rechazar la solicitud (error)
 - En órdenes de borrado de datos: que la CP que se pierda no deje una CF de otra tabla sin emparejar
 - Soluciones: rechazar la solicitud o propagar el efecto
 - En órdenes de modificación de datos (UPDATE): se monta a partir de las anteriores

Valores nulos en el MR

- Definición:
 - Señal utilizada para representar:
 - Información desconocida
 - Inaplicable
 - Inexistente
 - No válida
 - No proporcionada
 - Indefinidad, ...
 - Necesidad de valores nulos:
 - Inserción de tuplas con ciertos atributos desconocidos
 - Añadir atributos nuevos al esquema de una relación
 - Atributos inaplicables a ciertas tuplas



<null>

Valores nulos en el MR

- El tratamiento de valores nulos exige:
 - Comparaciones: problemas para saber si dos valores nulos son o no son iguales
 - Operaciones aritméticas: son nulas si alguno de los operandos es nulo
 - Operaciones algebraicas: problemas en algunas operaciones como la unión externa
 - Funciones de agregación: normalmente los nulos no entran a formar parte de las tuplas evaluadas
 - Ej: media aritmética



Doce reglas de Codd

- *Las 12 reglas de Codd son un sistema de reglas (numeradas del 0 al 12) propuestas por Edgar F. Codd, del modelo relacional para las bases de datos, diseñado para definir qué requiere un sistema de administración de base de datos*
 - *Codd se percató de que existían bases de datos en el mercado las cuales decían ser relacionales, pero lo único que hacían era guardar la información en las tablas, sin estar estas tablas literalmente normalizadas; entonces éste publicó 13 reglas que un verdadero sistema relacional debería tener aunque en la práctica algunas de ellas son difíciles de realizar. Un sistema podrá considerarse "más relacional" cuanto más siga estas reglas*

Doce reglas de Codd

- 0: Cualquier sistema que se proclame como relacional, debe ser capaz de gestionar sus bases de datos enteramente mediante sus capacidades relacionales *[también de Wikipedia]*
- 1 Representación de la información: toda información debe representarse en forma de tabla, principio básico del modelo relacional
- 2 Acceso garantizado: todo dato debe ser accesible mediante una combinación de un nombre de tabla, un valor de su clave y el nombre de una columna
- 3 Tratamiento sistemático de valores nulos: estos valores han de ser tratados sistemáticamente por el sistema, el cual ha de ofrecer las posibilidades necesarias para su tratamiento
- 4 Catálogo en línea basado en el modelo relacional: la representación de la descripción de la BD debe ser igual a la de los otros datos, y su acceso debe poder realizarse por medio del mismo lenguaje relacional. El MD para la metabase debe ser también relacional

Doce reglas de Codd

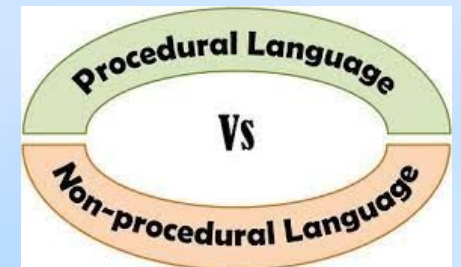
- 5 Sublenguaje de datos completo: debe existir un lenguaje que permita un completo manejo de la BD (definición y manipulación de datos, definición de vistas, restricciones de integridad, autorizaciones y gestión de transacciones)
- 6 Actualización de vista: toda vista teóricamente autorizable debe poder ser actualizada por el sistema
- 7 Inserción, actualización y eliminación de alto nivel: todas las operaciones de manipulación de datos deben operar sobre conjuntos de filas
- 8 Independencia física de los datos: el acceso lógico a los datos debe mantenerse incluso cuando cambien los métodos de acceso o la forma de almacenamiento

Doce reglas de Codd

- 9 Independencia lógica de los datos: los programas de aplicación no deben verse afectados por cambios realizados en el esquema de la BD
- 10 Independencia de integridad: las reglas de integridad de una BD deben ser definibles por medio del sublenguaje de datos relacional y habrán de almacenarse en el DD de la BD, no en los programas de aplicación
- 11 Independencia de distribución: debe existir un sublenguaje de datos que pueda soportar BD distribuidas sin alterar los programas de aplicación cuando se distribuyan los datos
- 12 Regla de la no subversión: si un SGBD soporta un lenguaje de bajo nivel que permita el acceso fila a fila, éste no puede utilizarse para saltarse las reglas de integridad expresadas por medio del lenguaje de más alto nivel

Lenguajes de consulta

- Permiten al usuario manipular la info. de la BD
- Se clasifican en:
 - *Procedimentales*: el usuario indica las operaciones y la secuencia en que se deben realizar. Álgebra relacional
 - *No procedimentales*: el usuario indica lo que quiere obtener sin especificar cómo obtenerlo. Cálculo relacional
- Los lenguajes de consultas pasaron a denominarse de manipulación (*DML*)
 - Porque también permiten insertar, eliminar y modificar



Lenguajes de consulta

- Ejemplo de Álgebra relacional
 - Fechas en las que sacaron todos los libros con código mayor que 10

$$A \leftarrow (\Pi_{\text{cód_Lib, Fecha}}(\text{Préstamos}))$$

$$B \leftarrow (\sigma_{\text{Cód_Lib} > 10}((\Pi_{\text{Cód_Lib}}(\text{Libros})))$$

$$A/B$$

- Hay que indicar cada paso que se da

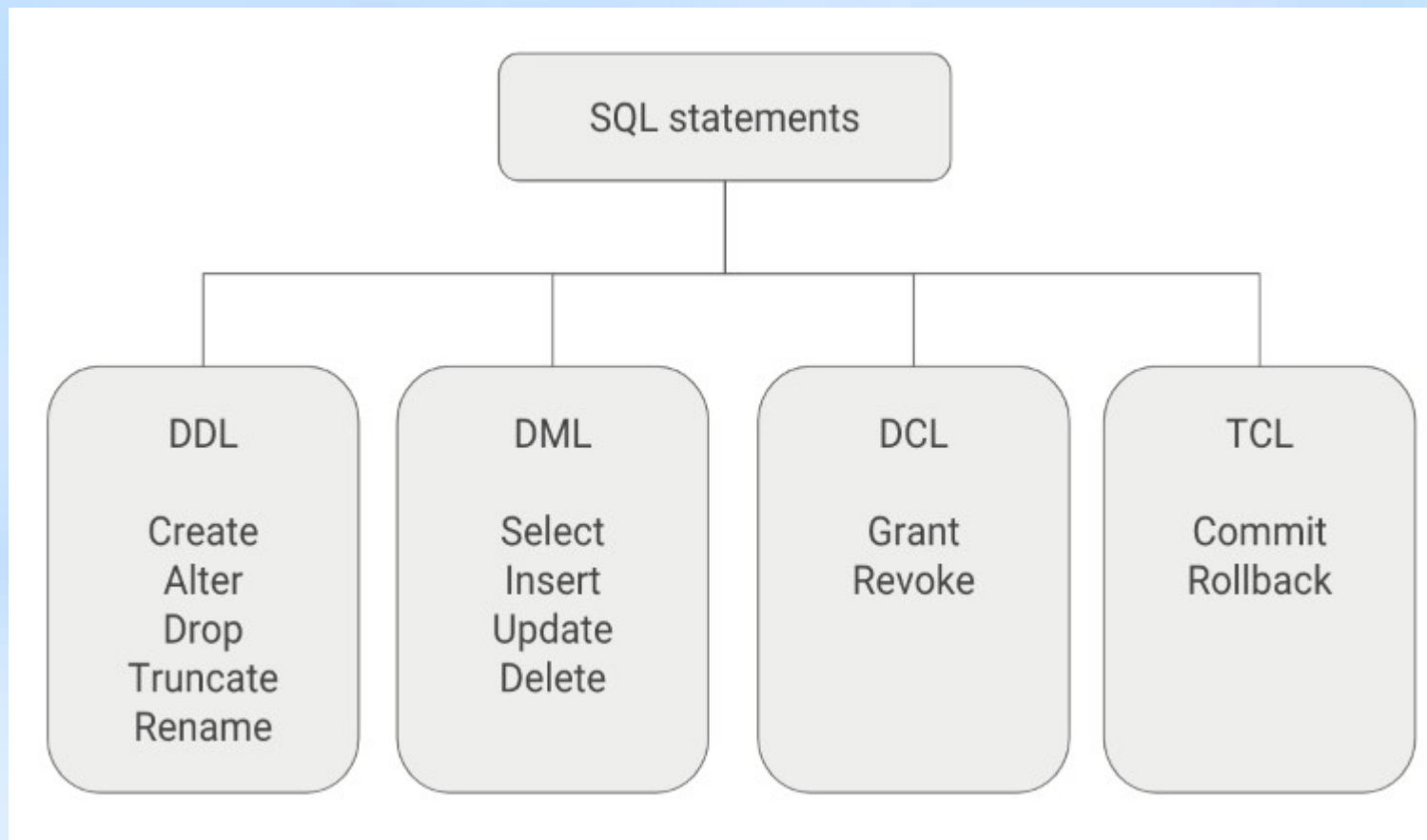
Lenguajes de consulta

- Ejemplo de Cálculo relacional
 - Listar los datos de los socios que tienen al menos un libro en préstamo así como el código del libro y la fecha del préstamo
 - $\text{Resultado}(\text{Cód_Soc}, \text{Nombre}, \text{Apellido}, \text{Cód_Lib}, \text{Fecha})$
 $\{t \mid t \in \text{Resultado}\}$
 $\{t \mid (\exists r) ((r \in \text{Prestamos}) \wedge$
 $(r.\text{Cód_Soc} = t.\text{Cód_Soc})) \wedge$
 $(\exists s) ((s \in \text{Socios}) \wedge (s.\text{Cód_Soc} = t.\text{Cód_Soc}))\}$
 - Hay que indicar las condiciones del resultado

Lenguajes de consulta

- Los SGBDR disponen de:
 - *DML*: Lenguaje de manipulación de datos: manipulación de la información (instancias) de la BD
 - *DDL*: Lenguaje de definición de datos: creación de los esquemas de las relaciones
 - *DCL*: Lenguaje de control de datos: control de la seguridad de los datos, seguridad de las estructuras de esos datos, control de acceso por parte de los usuarios, información y auditoría del sistema, etc
 - *TCL*: Lenguaje de control de transacciones
 - *Lenguajes huéspedes*: lenguaje de alto nivel que soporta instrucciones de un lenguaje de manipulación de datos, como por ejemplo SQL, y que interacciona con la BD

Lenguajes de consulta



Ejercicio

- Buscar esquemas de bases de datos relacionales de software real:
 - De referencia (normalmente software libre)
 - MediaWiki https://www.mediawiki.org/wiki/Manual:Database_layout/es
 - Otros: Moodle, Drupal, etc
 - De PFCs / TFGs de GII en la UCA
 - Otros: <https://rodin.uca.es/>
 - Carpintería metálica <https://rodin.uca.es/handle/10498/9754>
 - Elija una tabla y responda a:
 - ¿Cuáles son los atributos?
 - ¿Se indica el dominio de cada uno?
 - ¿Hay una descripción de cada atributo?
 - ¿Hay una clave primaria?
 - ¿Hay claves foráneas?
 - ¿Se usan restricciones tipo NULL, UNIQUE, CHECK?

Ejercicio

- Navegue instancias de bases de datos:
 - Try demo: <https://www.phpmyadmin.net/try/>
 - MySQL playground:
<https://extendsclass.com/mysql-online.html>

Referencias

- Apuntes Esther Gadeschi
- Libro Elmasri, 3^a ed.

Gracias por la atención
¿Preguntas?