
Tema 3: Modelo Relacional

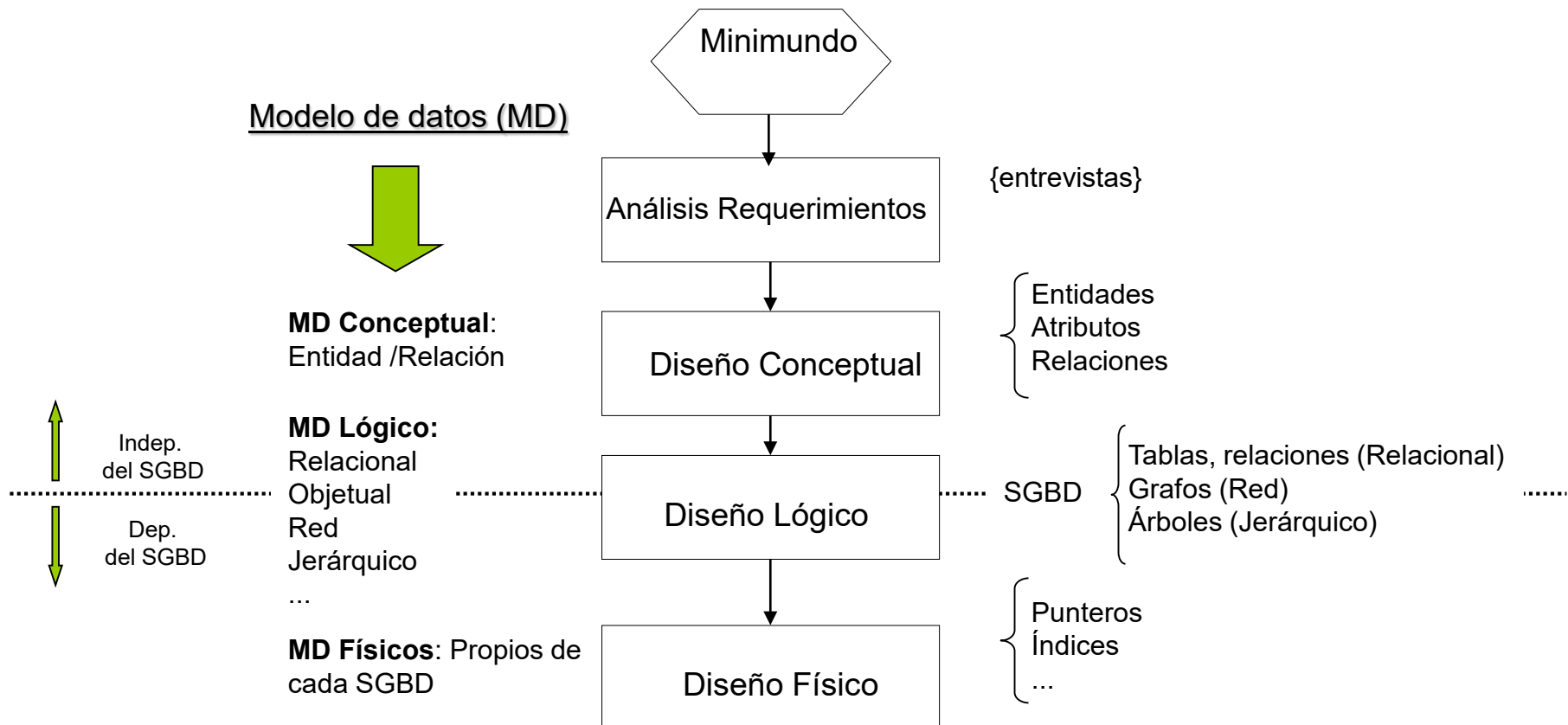


Índice

- I. Introducción
- II. Orígenes del Modelo Relacional
- III. Estructura de datos Relacional
- IV. Restricciones del Modelo Relacional
- V. Bibliografía

3.1. Introducción

→ *Fases del diseño de una BD*



3.1. Introducción

→ *Fases del diseño de una BD*

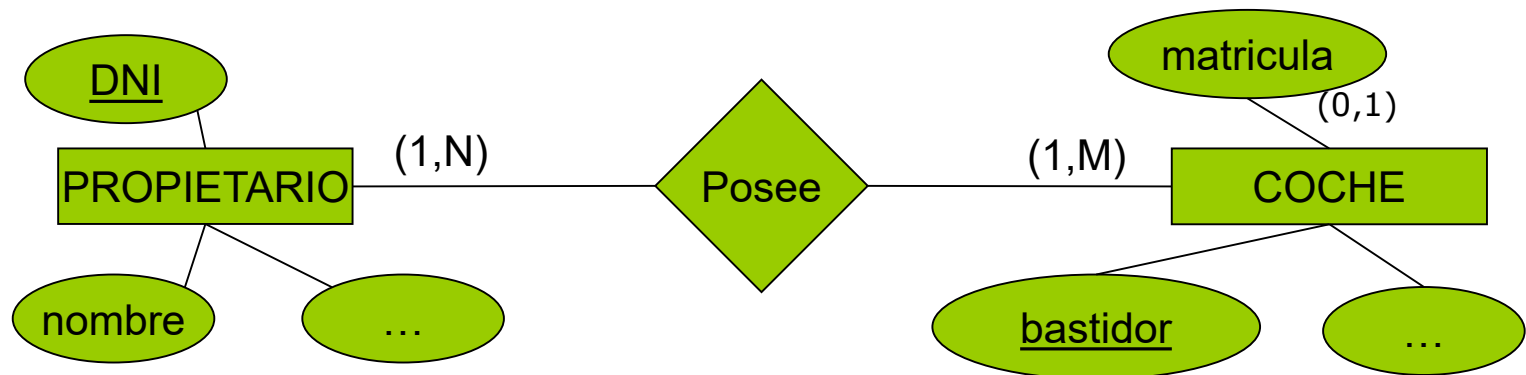
- ❑ **Base de datos de coches:** servirá como seguimiento de los coches y los propietarios que estos tienen a lo largo de toda su vida útil.
- ❑ **ANÁLISIS DE REQUISITOS**
 - Tras la fase **de recopilación de requisitos y análisis**, los diseñadores de la BD proporcionan la siguiente **descripción del *minimundo*** (*los requisitos de datos*):
 - ❑ Se almacenará el nombre, DNI, dirección y fecha de nacimiento de cada propietario.
 - ❑ Se almacenará la matrícula, número de bastidor, marca, modelo, potencia y color de cada coche.
 - ❑ En el momento de agregar un coche a la BD puede conocerse o no su matrícula, aunque sí el nº de bastidor.
 - ❑ Cada propietario puede poseer 1 o varios coches.
 - ❑ Cada coche pertenece a 1 o a varios propietarios.

3.1. Introducción

→ *Fases del diseño de una BD*

□ **DISEÑO CONCEPTUAL**

- Se crea un **esquema conceptual** para la BD mediante un MD conceptual de alto nivel.



- Se utilizan conceptos como entidades, relaciones, restricciones, pero no se incluyen detalles de implementación.

3.1. Introducción

→ *Fases del diseño de una BD*

□ **DISEÑO LÓGICO**

- Se implementa la BD en un SGBD comercial. Se utilizará el **MD Lógico** proporcionado por el SGBD. Será necesario transformar el MD Conceptual a un MD Lógico.

COCHE					
<u>bastidor</u>	matricula	marca	modelo	potencia	color
PK		NN	NN	NN	NN

PROPIETARIO			
<u>dni</u>	nombre	direccion	fecha_nacimiento
PK	NN	NN	NN

PROPIETARIO_COCHE	
<u>dni</u>	bastidor
PK	
FK (PROPIETARIO)	FK (COCHE)

- Los conceptos utilizados son propios del MD implementado por el SGBD.
- Tras la transformación, se obtiene el **esquema lógico** de la BD.

3.1. Introducción

→ *Fases del diseño de una BD*

□ **DISEÑO FÍSICO**

- Se especifican las estructuras de almacenamiento interno, los índices, las rutas de acceso, organización de los ficheros de la BD, etc.
- Los conceptos que se utilizan son propios del SGDB.
- Como resultado se obtendrá **esquema interno** de la BD.

3.2. Orígenes del MR

- ❑ Propuesto por Codd en un artículo publicado en la ACM en 1970.
- ❑ Los objetivos que perseguía con su nuevo modelo eran:
 - Independencia lógica
 - Independencia física
 - Flexibilidad
 - Uniformidad
 - Sencillez
- ❑ Se basa en dos ramas de las matemáticas:
 - Teoría de conjuntos
 - Lógica de predicados de primer orden
- ❑ El tener una base matemática lo convierte en un modelo ***seguro, robusto, fiable y predecible***.

3.2. Orígenes del MR

Modelo de Datos: conjunto de conceptos que sirven para describir, en distintos niveles de abstracción, la **estructura** (esquema) de una BD. Es decir, los tipos de datos, relaciones y **restricciones** que han de cumplirse, así como las **operaciones** para consultar y actualizar los datos.

- En MD Relacional se distinguen 3 partes:
 - **Estructural**: Estructuras de datos que soporta.
 - **Semántica**: Definida mediante un conjunto de restricciones o reglas de integridad.
 - **Manipulativa**: Operadores asociados.

3.3. Estructura de datos relacional

- Se basa en el concepto matemático de **relación**.
- De manera informal, una relación puede entenderse como **una tabla con columnas y filas** en la que cada fila representaría una colección de valores relacionados entre sí.
- Ej.: Tabla para almacenar inf. sobre coches

columnas

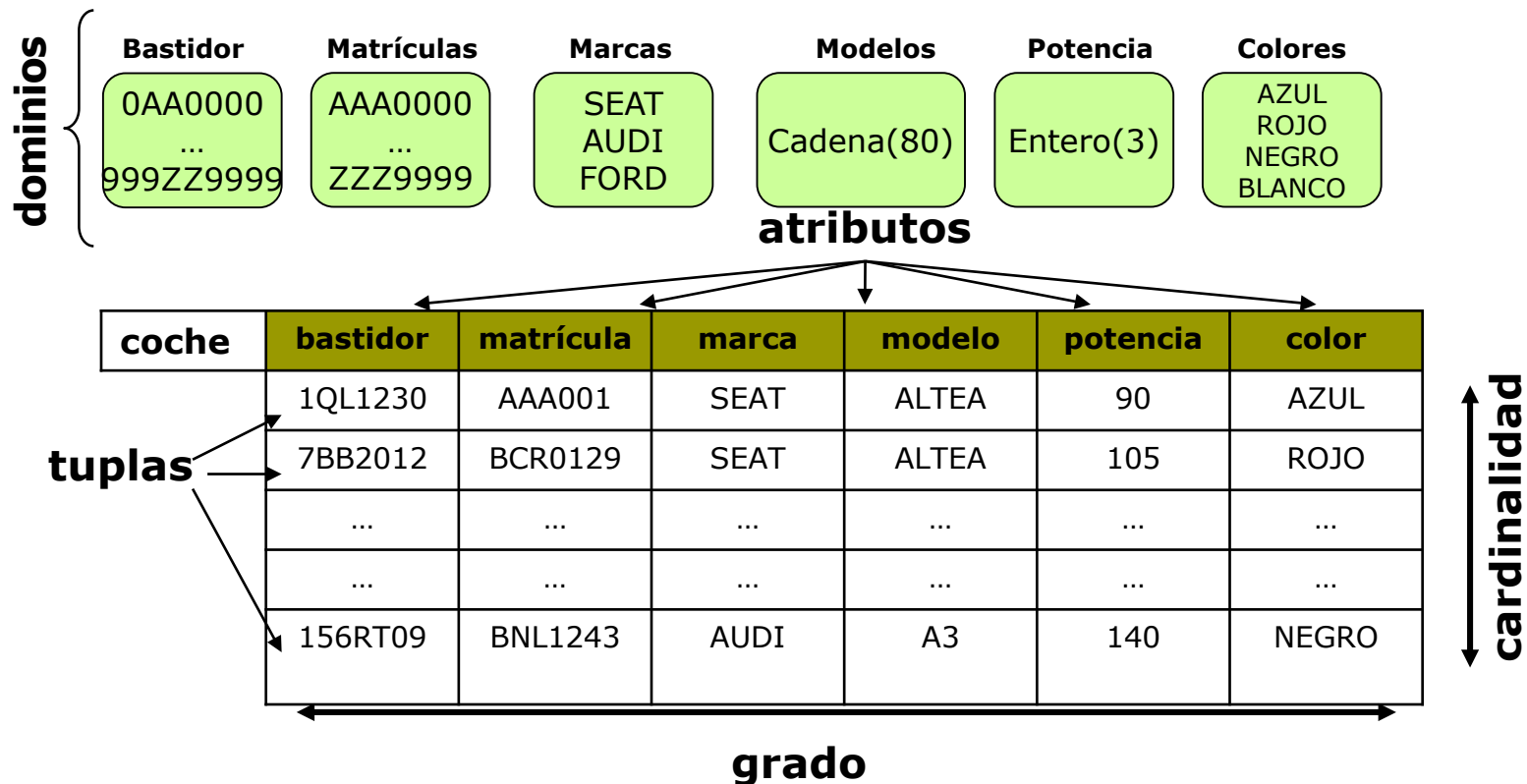
coche	bastidor	matrícula	Marca	modelo	potencia	color
	1QL1230	AAA001	SEAT	ALTEA	90	AZUL
	7BB2012	BCR0129	SEAT	ALTEA	105	ROJO
	156RT09	BNL1243	AUDI	A3	140	NEGRO

filas

- Todos los valores de una columna tienen el mismo tipo de datos.
- Base de Datos Relacional = Conjunto de relaciones

3.3. Estructura de datos relacional

□ Correspondencia relación – tabla:



3.3. Estructura de datos relacional


- **Fila (Tupla):** unidad de datos más pequeña que se puede insertar o borrar en una tabla.
 - Cada fila representa una relación entre un conjunto de valores.

coche	bastidor	matrícula	marca	modelo	potencia	color
	1QL1230	AAA001	SEAT	ALTEA	90	AZUL
	7BB2012	BCR0129	SEAT	ALTEA	105	ROJO
	156RT09	BNL1243	AUDI	A3	140	NEGRO

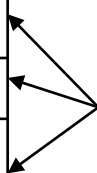
3.3. Estructura de datos relacional

- **Dominio:** es el **conjunto de valores atómicos** que puede tomar un atributo.
 - Cada atributo se define sobre un dominio (pudiendo haber varios atributos definidos sobre el mismo dominio).
 - Cada valor es **atómico**.
 - Los valores son **homogéneos** (todos del mismo tipo).
 - Incluyen el valor **nulo**.
 - Se puede definir por **intensión** o por **extensión**.

atributos



coche	bastidor	matrícula	marca	modelo	potencia	color
	1QL1230	AAA001	SEAT	ALTEA	90	AZUL
	7BB2012	BCR0129	SEAT	ALTEA	105	ROJO
	156RT09	BNL1243	AUDI	A3	140	NEGRO



tuplas

Atributo	Nombre del Dominio	Descripción	Definición
matricula	NUM_MAT	Posibles valores para las matrículas	Cadena de 7 caracteres; rango: AAA0000-ZZZ9999
color	COLORES	Posibles valores para colores	{AZUL, ROJO, NEGRO, BLANCO}

3.3. Estructura de datos relacional

- Formalmente, una **relación** es un subconjunto del producto cartesiano de dominios.
- $r \subseteq \text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_n)$

r	A_1	A_2	...	A_n
	a_{11}	a_{21}	...	a_{n1}
	a_{12}	a_{22}	...	a_{n2}
	a_{1m}	a_{2m}	...	a_{nm}

n = grado de la relación (*nº de atributos*)
 m = cardinalidad (*nº de tuplas*)

- Los dominios de 2 ó más atributos pueden ser coincidentes.

3.3. Estructura de datos relacional

□ **Esquema de Relación** (*Intensión*):

- Está constituido por el **nombre de la relación** R y la **cabecera** (conjunto de n pares atributo-dominio $\{(A_i:D_i)\}_{i=1..n}$ donde n es el grado).

□ Ej:

COCHE $\{(matricula:NUM_MAT),marca:MARCAS),..., (color:COLORES)\}$

□ **Estado de la relación** (*Extensión*):

- Está constituido por el **esquema** de la relación y el **cuerpo** de la relación (conjunto de tuplas que contiene en un instante determinado).
- El estado de relación de un esquema $R\{(A_i:D_i)\}_{i=1..n}$ se denota por **$r(R)$** .

3.3. Estructura de datos relacional

- Propiedades de las relaciones:
 - Cada relación tiene un nombre que ha de ser distinto de todas las demás.
 - Los valores de los atributos son atómicos. Se dice entonces que una relación está normalizada, o en primera forma normal.
 - En una relación no pueden existir dos atributos con el mismo nombre.
 - El orden de los atributos es irrelevante, es decir, no están ordenados.
 - En una relación no pueden existir dos tuplas iguales.
 - El orden de las tupla es irrelevante, es decir, no están ordenadas.

3.3. Estructura de datos relacional

- Tipos de relaciones (I):
 - **Relaciones base:** existen por sí mismas, no en función de otras relaciones, y se crean especificando explícitamente su esquema (nombre y conjunto de pares: atributo/dominio).
 - **Relaciones temporales:** a diferencia de las tablas base, desaparecen de la BD en un cierto momento sin necesidad de una acción de borrado específica del usuario.

3.3. Estructura de datos relacional

- Tipos de relaciones (II):
 - **Vistas** : es una representación de una tabla o de una combinación de tablas. Toma sus datos de las tablas en que se basa quienes, a su vez, pueden ser vistas. No contienen datos: las operaciones afectan a las tablas de las que derivan. Permiten hacer diferentes representaciones de los mismos datos para diferentes tipos de usuarios. Se emplean para:
 - Proporcionar un nivel extra de **seguridad**, restringiendo el acceso a un conjunto predeterminado de filas y/o columnas de una tabla.
 - **Ocultar la complejidad** de los datos (p.ej, una vista puede tener filas de distintas tablas, pero el usuario sólo tiene que recordar el nombre de la vista).
 - **Reducir la complejidad sintáctica** (el usuario puede seleccionar información de múltiples tablas sin saber ejecutar una operación de join).
 - Presentar los datos desde **otra perspectiva**

3.3. Estructura de datos relacional

- Tipos de relaciones (III):
 - **Instantáneas** : son relaciones derivadas igual que las vistas, es decir, se definen en términos de otras relaciones, pero tienen datos propios almacenados, los cuales son el resultado de ejecutar una consulta específica.
 - No se actualizan cuando cambian los datos de las relaciones sobre las que están definidas, pero se “refrescan” cada cierto tiempo, de acuerdo con lo indicado por el usuario en el momento de su creación. Son, por tanto, sólo de lectura, no pudiendo ser actualizadas por el usuario, sino únicamente “refrescadas” por el sistema.

3.3. Estructura de datos relacional

- Sea R una relación **$R(A1:D1, A2:D2, \dots, An:Dn)$**
- **Superclave:** subconjunto de atributos SK de R , tal que cumple la restricción de **unicidad**, es decir:
 - NO existen dos tuplas con la misma combinación de valores para SK.
- **Clave:** es una superclave K de R que cumple la restricción de **irreductibilidad**, es decir:
 - NINGÚN subconjunto de K cumple la restr. de unicidad

3.3. Estructura de datos relacional

□ *Ejemplo de clave de una relación:*

coche	bastidor	matrícula	marca	modelo	potencia	color
	1QL1230	AAA001	SEAT	ALTEA	90	AZUL
	7BB2012	BCR0129	SEAT	ALTEA	105	ROJO
	156RT09	BNL1243	AUDI	A3	140	NEGRO

- ¿el atributo marca es clave?
- ¿el atributo bastidor es clave?
- ¿el conj. formado por los atributos bastidor y marca es clave?

3.3. Estructura de datos relacional

- **Clave candidata**
 - Cada una de las claves de una relación.
- **Clave compuesta**
 - Clave candidata compuesta por más de un atributo
- **Clave primaria** (*Primary Key*)
 - Clave candidata elegida como clave de la relación.
- **Clave alternativa**
 - Clave candidata no elegida como clave primaria.

3.3. Estructura de datos relacional

□ Clave Foránea (Foreign Key)

- Se quiere representar la información de empleados y departamentos. Podría realizarse de este modo:

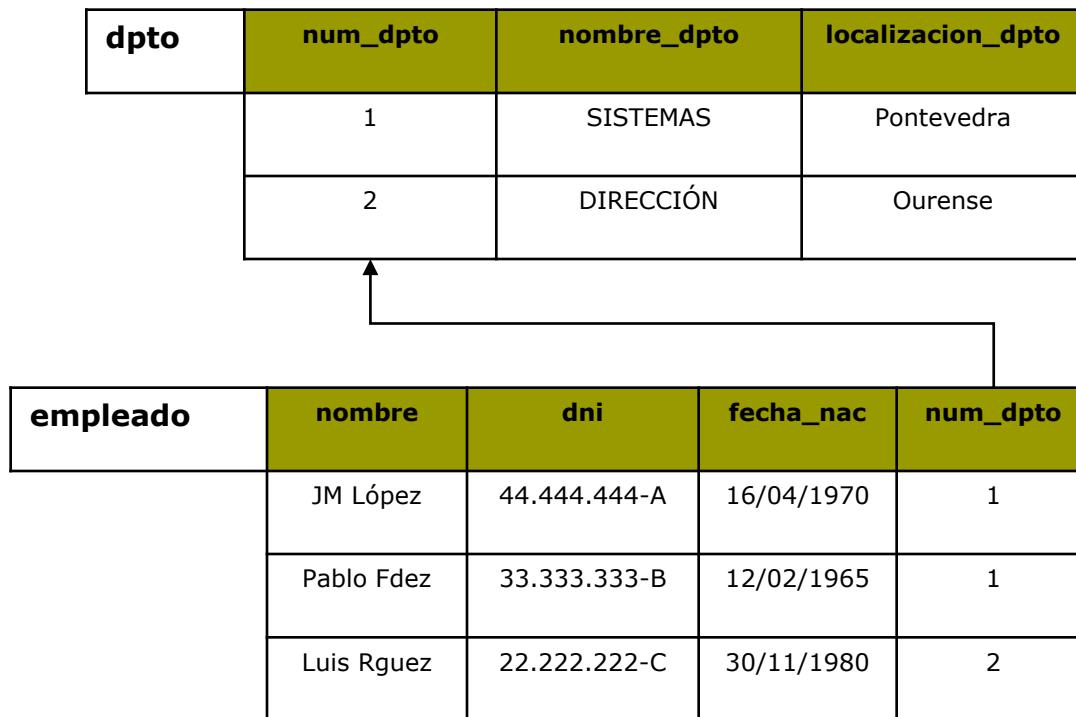
empleado	nombre	dni	fecha_nac	num_dpto	nombre_dpto	localizacion_dpto
	JM López	44.444.444-A	16/04/1970	1	SISTEMAS	Pontevedra
	Pablo Fdez	33.333.333-B	12/02/1965	1	SISTEMAS	Pontevedra
	Luis Rguez	22.222.222-C	30/11/1980	2	DIRECCIÓN	Ourense

- Esta solución presenta varios problemas

3.3. Estructura de datos relacional

□ Clave Foránea (Foreign Key)

- Situación ideal: dos relaciones



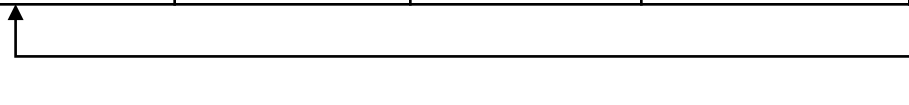
3.3. Estructura de datos relacional

❑ Clave foránea (*Foreign Key*)

- Conjunto de atributos que, si no son nulos, identifican unívocamente una tupla de otra relación (es decir, son clave primaria de otra relación).
 - ❑ Debe estar definida sobre el mismo dominio que la correspondiente PK y tener un valor correspondiente en la tabla donde es clave primaria.
 - ❑ Puede referenciar a la misma relación.
 - ❑ Puede tener valores *nulos*.

editorial	cod_ed	direccion	pais	ciudad
	Ra-Ma	Canillas, 144	España	Madrid
	Mc Graw-Hill	Basauri 17	España	Madrid

libro	cod	titulo	...	cod_ed
	01q2w3	Fundamentos y mod...	...	Ra-Ma
	0033344	Fudamentos de dis...	...	Mc Graw-Hill



3.3. Estructura de datos relacional

□ **Valor Nulo**

- En el mundo real existe información...
 - **Desconocida** en un determinado momento
 - Ej: N° de matrícula en el momento de la compra de un coche
 - **Ausente**
 - Ej: Teléf. móvil aplicado a una persona que no dispone de él
 - **No aplicable** a ciertos atributos
 - Ej: Sueldo aplicado a un estudiante
- Para representar esa información en los sistemas de bases de datos se emplea el valor NULO.
 - Nulo NO representa ni la cadena vacía ni cero, sino que especifica información *desconocida*.
 - No hay dos nulos iguales
- Representan problemas de tratamiento y es conveniente no utilizarlos.

3.4. Restricciones del modelo relacional

Modelo de Datos: conjunto de conceptos que sirven para describir, en distintos niveles de abstracción, la **estructura** (esquema) de una BD. Es decir, los tipos de datos, relaciones y **restricciones** que han de cumplirse, así como las **operaciones** para consultar y actualizar los datos.

- En MD Relacional se distinguen 3 partes:
 - **Estructural:** Estructuras de datos que soporta.
 - **Semántica:** Definida mediante un conjunto de restricciones o reglas de integridad.
 - **Manipulativa:** Operadores asociados.

3.4. Restricciones del modelo relacional

- ❑ Define los diferentes tipos de **restricciones** sobre los datos que se pueden especificar en un esquema de una base de datos relacional.
- ❑ Estas reglas son las que los datos almacenados en dicha estructura deben cumplir para garantizar que son correctos.
- ❑ Algunas configuraciones de valores NO tienen SENTIDO
 - pues no representan ningún estado posible del minimundo
 - ❑ 2 personas distintas con el mismo DNI
 - ❑ Un empleado sin NSS
 - ❑ Una película sin director

3.4. Restricciones del modelo relacional

- **Restricciones inherentes:** No las define el usuario, sino que son impuestas por el propio modelo.
 - No hay dos tuplas iguales
 - ➔ Obligatoriedad de PK
 - El orden de las tuplas no es significativo
 - El orden de los atributos no es significativo
 - Cada atributo sólo puede tomar un único valor del dominio sobre el que está definido (relación *normalizada*). Es decir, no están permitidos los atributos multivaluados.

3.4. Restricciones del modelo relacional

- ❑ **Restricciones inherentes:** No las define el usuario, sino que son impuestas por el propio modelo.
 - **Regla de Integridad de Entidad:** Ningún atributo que forme parte de la PK de una relación puede tomar un valor nulo.
 - Ejemplo: En el momento de agregar un coche a la BD puede conocerse o no su matrícula, aunque sí nº de bastidor

coche	bastidor	matrícula	marca	modelo	potencia	color
	1QL1230	AAA001	SEAT	ALTEA	90	AZUL
	7BB2012	----	SEAT	ALTEA	105	ROJO
	156RT09	BNL1243	AUDI	A3	140	NEGRO

3.4. Restricciones del modelo relacional

- ❑ **Restricciones semánticas:** Son ofrecidas por el modelo para que el usuario pueda reflejar en el esquema, lo más fielmente posible, la semántica del mundo real.
- ❑ Tipos de restricciones semánticas del MR:
 - **Clave primaria** (PRIMARY KEY)
 - **Unicidad** (UNIQUE)
 - **Obligatoriedad** (NOT NULL)
 - **Integridad Referencial** (FOREIGN KEY)
 - **De usuario:** Condición sobre un predicado definido sobre un conjunto de atributos, tuplas o dominios que debe ser verificado en toda operación de actualización.
 - ❑ Verificación (CHECK)
 - ❑ Aserción (ASSERTION)
 - **Disparadores** (Triggers)
 - **Dependencias Funcionales**

3.4. Restricciones del modelo relacional

- ❑ **Clave primaria** (PRIMARY KEY)
 - Permite declarar un atributo o conjunto de atributos como clave primaria de una relación.

- ❑ **Unicidad** (UNIQUE): Indica que los valores de un atributo (o conjunto) no pueden repetirse en una relación.
 - Permite definir claves candidatas.

- ❑ **Obligatoriedad** (NOT NULL):
 - Indica que el atributo (o conjunto) no admite nulos.

3.4. Restricciones del modelo relacional

- **Integridad Referencial (FOREIGN KEY):** Permite declarar un atributo o conjunto de atributos como clave foránea.
 - Además se debe determinar las consecuencias de las operaciones de borrado y modificación sobre la relación referenciada:
 - **NO ACTION:** El borrado de tuplas de la relación (*editorial*) que contiene la clave referenciada (*cod_ed*) (o la modificación de dicha clave) sólo se permite si no existen tuplas con este valor en la relación (*libro*) que contiene la clave foránea. Opción por defecto.
 - **CASCADE:** El borrado de tuplas de la relación (*editorial*) que contiene la clave referenciada (o la modificación de dicha clave) lleva consigo el borrado (o modificación) en cascada de las tuplas de la relación (*libro*) que contiene la clave foránea.

editorial	cod_ed	direccion	pais	ciudad
	Ra-Ma	Canillas, 144	España	Madrid
	Mc Graw-Hill	Basauri 17	España	Madrid

libro	cod	titulo	...	cod_ed
	01q2w3	Fundamentos y mod...	...	Ra-Ma
	0033344	Fudamentos de dis...	...	Mc Graw-Hill

3.4. Restricciones del modelo relacional

- **SET NULL:** El borrado de tuplas de la relación (*editorial*) que contiene la clave referenciada (*cod_ed*) (o la modificación de dicha clave) lleva consigo poner a nulos los valores de las claves foráneas de la relación (*libro*) que referencia.
- **SET DEFAULT:** El borrado de tuplas de la relación (*editorial*) que contiene la clave referenciada (*cod_ed*) (o la modificación de dicha clave) lleva consigo poner un valor por defecto a la clave foránea de la relación (*libro*) que referencia.

- Las opciones de borrado y modificación pueden ser distintas

editorial	cod_ed	direccion	pais	ciudad
	Ra-Ma	Canillas, 144	España	Madrid
	Mc Graw-Hill	Basauri 17	España	Madrid

libro	cod	titulo	...	cod_ed
	01q2w3	Fundamentos y mod...	...	Ra-Ma
	0033344	Fudamentos de dis...	...	Mc Graw-Hill

3.4. Restricciones del modelo relacional

□ Ejemplos de algunas restricciones en SQL

```
CREATE TABLE editorial (  
    nombre_e CHAR(20) PRIMARY KEY,  
    dirección CHAR(50) NOT NULL,  
    ciudad CHAR (15),  
    país CHAR(15));
```

```
CREATE TABLE libro (  
    codigo char(3),  
    titulo char(50) UNIQUE,  
    idioma char(25),  
    nombre_e char(20),  
    PRIMARY KEY (codigo),  
    FOREIGN KEY (nombre_e) REFERENCES editorial (nombre_e)  
    ON DELETE SET NULL  
    ON UPDATE CASCADE  
);
```

3.4. Restricciones del modelo relacional

- ❑ **De usuario:** Condición sobre un predicado definido sobre un conjunto de atributos, tuplas o dominios que debe ser verificado en toda operación de actualización.
 - **Verificación** (CHECK): Comprueba, en toda operación de actualización, si el predicado es cierto o falso y, en el segundo caso, rechaza la operación.
 - ❑ Sólo puede afectar a un elemento
 - ❑ Puede tener nombre
 - ❑ Ej: CHECK FECHA_NAC < 1990
 - **Aserción** (ASSERTION): Similar a la anterior, pero puede afectar a varios elementos (p. ej: dos tablas).
 - ❑ Siempre debe tener un nombre
 - ❑ Ej: CREATE ASSERTION solo_un_presi CHECK (SELECT COUNT(*) FROM emp WHERE puesto='PRESIDENTE') < 2);

3.4. Restricciones del modelo relacional

- ❑ **Disparadores** (Triggers): Restricción en la que el usuario puede especificar la respuesta (acción) ante una determinada condición.
 - Ej: si una beca es solicitada por más de 50 alumnos, se introduce un texto en una tabla de mensajes para que la persona que gestiona las becas considere si es necesario ofrecer más becas.

```
CREATE TRIGGER Comprobar_Matriculados
AFTER INSERT ON SOLICITA
DECLARE
NUM_SOLICITUDES Number;
BEGIN
SELECT COUNT(*) INTO NUM_SOLICITUDES FROM SOLICITA;
IF NUM_SOLICITUDES > 50 THEN
INSERT INTO MENSAJES VALUES ('Hay más de 50 solicitudes');
END IF;
END Comprobar_Matriculados;
```

3.4. Restricciones del modelo relacional

- **Dependencia Funcional (DF):** Establece una relación funcional entre dos conjuntos de atributos X e Y.
 - Especifica que el valor de X determina el valor de Y en todos los estados de una relación.
 - Las DF's se utilizan como herramientas para analizar la calidad de los diseños de relación y para normalizar las relaciones que mejoran su calidad.

3.4. Restricciones del modelo relacional

Modelo de Datos: conjunto de conceptos que sirven para describir, en distintos niveles de abstracción, la **estructura** (esquema) de una BD. Es decir, los tipos de datos, relaciones y **restricciones** que han de cumplirse, así como las **operaciones** para consultar y actualizar los datos.

- ❑ En MD Relacional se distinguen 3 partes:
 - **Estructural:** Estructuras de datos que soporta.
 - **Semántica:** Definida mediante un conjunto de restricciones o reglas de integridad.
 - **Manipulativa:** Operadores asociados.

3.4. Restricciones del modelo relacional

- ▣ Los lenguajes del MR permiten manipular la estructura de datos relacional y especificar consultas de datos.
- ▣ Los operadores asociados a la parte manipulativa del modelo relacional constituyen el **álgebra y el cálculo relacional**.

3.5. Sistemas de Gestión de Bases de datos móviles

- Componentes:
 - Servidor de base de datos y SGBD que gestiona y almacena los datos corporativos y proporciona aplicaciones corporativas.
 - Base de datos remota y SGBD que gestiona y almacena los datos móviles.
 - Dispositivo móvil
 - Enlaces de comunicación bidireccionales entre el SGBD y el SGBD móvil
- Además de las características de un SGDB. Incluye:
 - Comunicación con el servidor centralizado de BD usando redes inalámbricas y acceso a Internet.
 - Replicar los datos en servidor central y móvil.
 - Sincronización de datos entre servidor y móvil.
 - Captura de datos de varias fuentes (ej: internet).
 - Gestionar datos en el dispositivo móvil.
 - Analizar los datos almacenados en móvil.
 - Crear aplicaciones móviles personalizadas.

3.5. Sistemas de Gestión de Bases de datos móviles

- Ejemplos de sistemas:
 - IBM: DB2 Everyplace.
 - SYBASE Adaptive server anywhere.
 - MICROSOFT: SQL Server CE
 - SQLite

3.6. Sistemas de Gestión de Bases de datos NoSQL

- Surgen por la necesidad de dar respuesta a las aplicaciones donde un usuario puede subir contenido.
- Permiten almacenar información en aquellas situaciones en las que las bases de datos relacionales generan problemas de escalabilidad y rendimiento, debido a la presencia de miles de usuarios concurrentes con millones de consultas diarias.
- No cumplen con el esquema entidad-relación. Tampoco utilizan una estructura de datos en forma de tabla donde se van almacenando los datos sino que para el almacenamiento hacen uso de otros formatos como clave-valor, mapeo de columnas o grafos.
- VENTAJAS:
 - Se ejecutan en máquinas con pocos recursos
 - Escalabilidad horizontal.
 - Pueden manejar gran cantidad de datos (tablas Hash).
 - No generan cuellos de botella.

3.6. Sistemas de Gestión de Bases de datos NoSQL

3.6.1.- Bases de datos SQL vs NoSQL

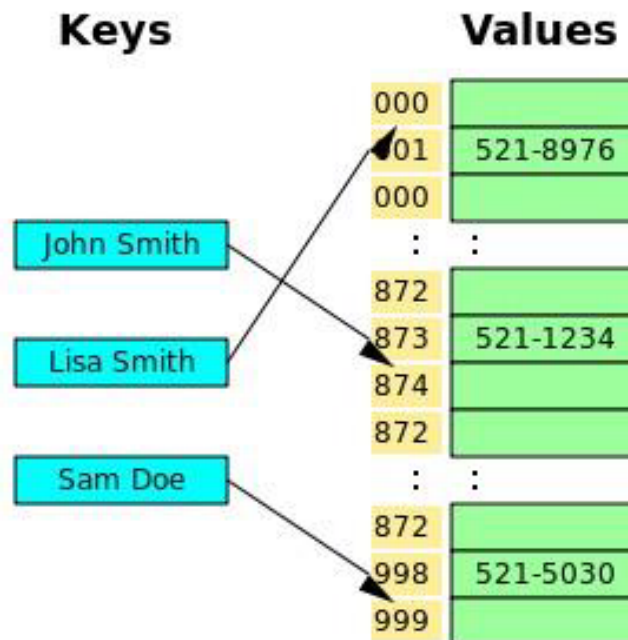
- No utilizan SQL como lenguaje de consultas.
- No utilizan estructuras fijas como tablas para el almacenamiento de los datos.
- No suelen permitir operaciones JOIN.
- Arquitectura distribuida

3.6. Sistemas de Gestión de Bases de datos

NoSQL

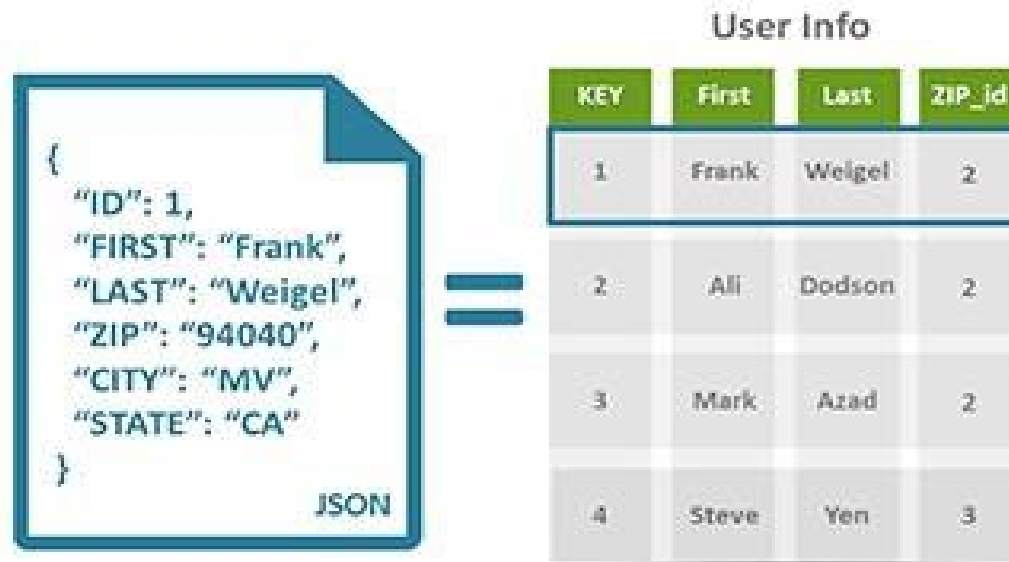
3.6.2.- Tipos de Bases de datos

- BD clave-valor: Cassandra (Facebook, Twitter), Redis (Flickr, Instagram, Github), Hbase (Yahoo, Adobe).
- Almacenan la información como BLOB



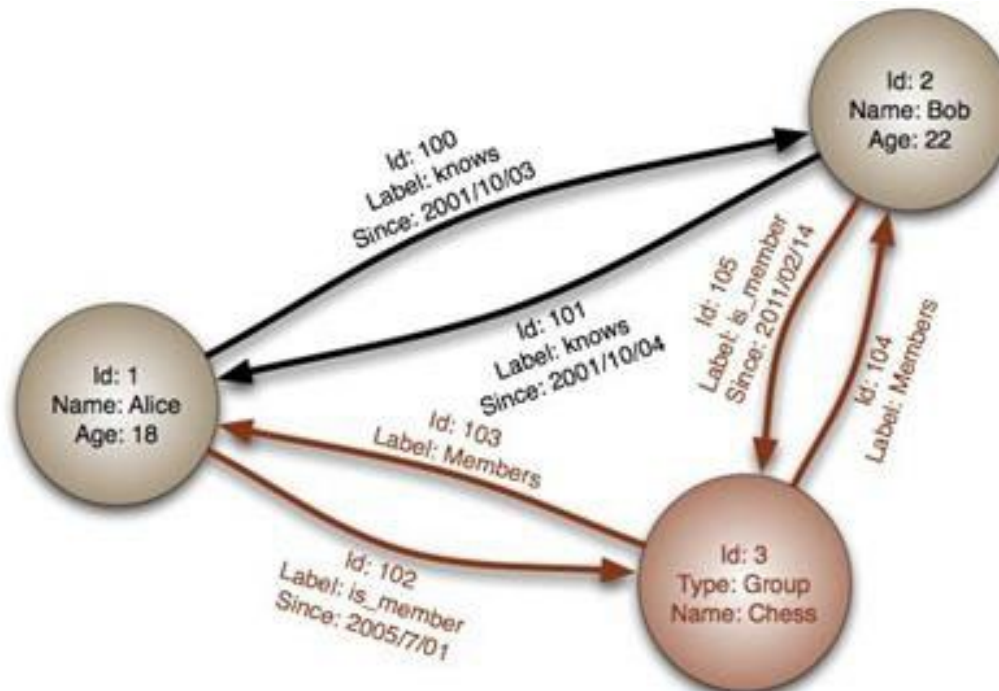
3.6. Sistemas de Gestión de Bases de datos NoSQL

- BD documentales: MongoDB (FourSquare, SourceForge, CERN), CouchDB
- Permiten la búsqueda por clave-valor y por contenido del documento.



3.6. Sistemas de Gestión de Bases de datos NoSQL

- BD en grafo: Neo4j (Infojobs), InfoGrid, Virtuoso
- La información se representa como nodos de un grafo y sus relaciones con las aristas del mismo, de manera que se puede hacer uso de la teoría de grafos para recorrerla



V. Bibliografía

- *[deMP99] de Miguel, A.; Piattini, M. **Fundamentos y modelos de bases de datos** (2ª edición). Madrid. Ra-ma, 1999. [cap. 5]*
- *[EN07] Ramez A. Elmasri, Shamkant B. Navathe. **Fundamentos de Sistemas de Bases de Datos** (5º edic.). Prentice-Hall. 2007. [cap. 6]*
- <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>