**Normalización**

La normalización de una base de datos es un paso que aplicamos a un modelo Relacional con el objetivo de minimizarla redundancia de datos. Existen 5 formas normales y son acumulativas.

**1FN:** A efectos prácticos, para pasar a 1FN debemos eliminar los atributos multivaluados.

**2FN:** Todos los atributos que no son de la clave primaria dependen de manera COMPLETA de la clave primaria, y no parcial.

**3FN:** No hay dependencias transitivas. Los atributos que no son clave, son independientes entre sí.

**Claves**

Claves candidatas, clave principal/primaria (PK), claves alternativas, claves ajenas o foráneas (FK).

**Transformación del modelo entidad-relación al modelo relacional (ver pdf 4 a partir de página 29)**

**Definiciones (página 8 del libro)**

**Tipos de lenguajes de bases de datos**

* **DDL (Data Definition Language):** Vamos a definir cómo se estructura una base de datos (número de tablas, tipo de datos, cuál va a ser la clave primaria de cada una de las tablas…), o a reestructurarla (quitando o añadiendo campos, cambiando la FK…). Continente donde vamos a meter el contenido.
* **DML (Data Manipulation Language):** Manipulamos la información de una base datos que previamente hemos definido con el DDL, modificando un dato, insertándolo o eliminándolo.
* **DCL (Data Control Language):** Es el control de acceso a una base de datos. Crea un control de que usuarios pueden acceder a una base de datos y que usuarios no pueden.
* **TCL (Transaction Control Languaje):** Si queremos hacer una transaction para pasar una base datos de A a B, controla que se haga o no se haga. En caso de quedarse a medias, no se hace y se queda en A. Si la transacción es exitosa, pasará a B.

**Comandos**

CREATE DATABASE prueba; SHOW DATABASES; USE prueba;.

CREATE TABLE ejemplo; SHOW TABLES; DROP TABLE ejemplo;.

ALTER TABLE … RENAME…; … DROP COLUMN…; … ADD…; …ADD…AFTER…; …ADD…FIRST; …ADD…PRIMARY KEY…; …MODIFY COLUMN…; auto\_increment = valor\_a\_partir\_del\_cual\_comienza\_a\_contar;

set insert\_ID = ….

DESCRIBE tabla (para mostrar el continente); INSERT INTO (introducir valores); SELECT \* FROM tabla (mostrar contenido) LIMIT 2,3 (para mostrar solo las 3 primerasa partir de la tercera posición; REPLACE INTO tabla (id, nombre) VALUES (1, “ejemplo”); SELECT nombre, poblacion DIV 1000 AS poblacion\_en\_miles FROM tabla;

SELECT cod, titulo FROM libro UNION SELECT \* FROM autor (deben coincidir en columnas las tablas);

SELECT \* FROM libro WHERE titulo LIKE “%de%” (% = nos da igual lo que tenga antes y/o después);

SELECT COUNT(\*) FROM libro WHERE autor=4 (para saber cuantos libros ha escrito);

SELECT autor.nombre (sumnistramos dos tablas), COUNT(\*) AS num\_libros FROM libro INNER JOIN (para que también aparezca el nombre del autor) autor ON autor.cod\_autor = libro.autor = libro.autor WHERE autor=4;

SELECT autor.nombre, COUNT(\*) FROM autor INNER JOIN libro ON libro.autor = autor.cod\_autor GROUP BY (autor) (agrupamos por autor para saber cuantos libros ha escrito cada autor con COUNT);

SELECT autor.nombre, COUNT(\*) FROM **autor INNER JOIN libro ON** libro.autor = autor.cod\_autor GROUP BY (autor); = … **autor, libro WHERE libro.autor**…

**Subconsultas**

SELECT nombre, existencias FROM inventario INNER JOIN producto ON cod\_producto = codigo where existencias = (select MIN(existencias) from inventario);

**Integridad:** se encarga de que los datos que la componen sean lo más correctos posibles.

**Atomicidad (*Atomicity*):** actúa como un proceso atómico, es decir, o todo (modificación, agregación o borrado) se realiza con éxito, o nada. Basta con que falle una mínima parte para que la operación no sea satisfactoria.

**• Consistencia (*Consistency*):** cuando se ejecuta la transacción, el sistema debe pasar de en un estado consistente a otro que también lo sea pese a los cambios que se han realizado.

**• Aislamiento (*Isolation*):** cada transacción debe actuar de forma secuencial.

**• Permanencia (*Durability*):** todos los cambios que se hayan producido cuando se realiza una transacción no se pierden, sino que permanecen