

# Fundamentos de Bases de Datos

domingo, 26 de noviembre de 2023 14:10

## Normalización:

Reglas para separar la información y optimizar la base de datos.

alumno	nivel_curso	nombre_curso	materia_1	materia_2
Juanito	Maestría	Data engineering	MySQL	Python
Pepito	Licenciatura	Programación	MySQL	Python

1FN:

Atributos atómicos: No tener campos repetidos

alumnos				
alumno_id	alumno	nivel_curso	nombre_curso	materia
1	Juanito	Maestría	Data engineering	MySQL
1	Juanito	Maestría	Data engineering	Python
2	Pepito	Licenciatura	Programación	MySQL
2	Pepito	Licenciatura	Programación	Python

2FN:

1FN y cada fila de la tabla debe depender de una clave única

alumnos			
alumno_id	alumno	nivel_curso	nombre_curso
1	Juanito	Maestría	Data engineering
2	Pepito	Licenciatura	Programación

materias		
materia_id	alumno_id	materia
1	1	MySQL
2	1	Python
3	2	MySQL
4	2	Python

3FN

Cumple 1FN y 2FN y los campos que no son claves no deben de tener dependencias; atributos no clave dependientes de nada excepto la clave.

alumnos			cursos		
alumno_id	alumno	curso_id	curso_id	nivel_curso	nombre_curso
1	Juanito	1	1	Maestría	Data engineering
2	Pepito	2	2	Licenciatura	Programación

  

materias		
materia_id	alumno_id	materia
1	1	MySQL
2	1	Python
3	2	MySQL
4	2	Python

4FN

Los campos multivaluados se identifican con una clave única.

alumnos			cursos		
alumno_id	alumno	curso_id	curso_id	nivel_curso	nombre_curso
1	Juanito	1	1	Maestría	Data engineering
2	Pepito	2	2	Licenciatura	Programación

  

materias		materias_por_alumno		
materia_id	materia	mpa_id	materia_id	alumno_id
1	MySQL	1	1	1

2	Python
---	--------

2	2	1
3	1	2
4	2	2

#### -----

#### Pasos para montar tablas en la base de datos:

1. Montar las tablas independientes (aquellas que no poseen llaves foráneas)
2. Montar las tablas dependientes (aquellas que tienen relación FK con otras).
3. Montar las tablas transitivas (tablas que se crean con la finalidad de romper las relaciones muchos a muchos)

#### -----

#### Acciones con las FK al borrar la PK asociada:

- Cascade: Si se borra algo en la tabla principal se afectara la tabla con la FK, ej. Si se borra el usuario se borran todos los posts asociados a este.
- Restrict: La base de datos restringirá el borrado hasta que las dependencias de la FK sean resuletas.
- Set Null: En la tabla, en el lugar donde iba la FK se reemplazara por NULL.
- No action: No haga nada.

#### -----

#### Nested queries:

Se usan cuando se necesita una operación de agregación, poco optimo ya que se hace un producto cartesiano, no escala. Big O de  $O(n^2)$ .

#### -----

#### Estructura de un query:

SELECT <Lo que quiero mostrar>

CASE

    WHEN <Condición y lo que quiero hacer cuando se cumple esta>

    ELSE <Lo que quiero hacer cuando no se cumpla la condición>

FROM <De dónde voy a tomar los datos>

WHERE <Los filtros de los datos que quiero mostrar>

GROUP BY <Porque grupos me interesa seleccionar la información>

ORDER BY <El orden en el que quiero mostrar mis datos>

HAVING <Los filtros que quiero poner en los datos agrupados>

#### -----

#### Tipos de bases de datos NoSQL:

- Clave valor  
Almacenar datos de manera rápida, se almacenan grandes cantidades de datos ligadas a un ID.
  - Dynamo DB
  - Cassandra
- Documentales  
Objetos de tipo JSON, con una estructura más definida.
  - MongoDB
  - Firestore
- Grafos  
Nodos o entidades que tiene relaciones muy complejas, relaciones de todos entre todos.
  - Neo4J
  - TitanDB
- Memoria  
Bases de datos volátiles con capacidad limitada de almacenamiento, altamente rápidas para el acceso.
  - MemCache
  - Redis
- Optimizadas para búsqueda  
Hacer queries y búsquedas complejas de manera sencilla
  - BigQuery
  - Elasticsearch

#### -----

#### Jerarquía de datos en Firestore:

Base de datos → Colecciones (tablas en SQL, entidades) → Documentos (info que queremos guardar).

**Top level collections:** Colecciones que se tienen a nivel de la ruta principal

Una **top level collection** se usa para relaciones de tipo **agregación**, por su parte, una **sub collection** se usa para una relación de tipo **composición**.

