1 manejar Python (vsc)

2 bases de datos noSQL

3 manejo de no sql (tabla como referencia) tpos de comandos que se usan en ambos

4 pasar y hechar código (montar una bd no relacioanl para practica)

5 montar el archivo en github

**Definición base de datos relacional**

Una base de datos relacional es un tipo de base de datos que organiza la información en tablas relacionadas. Cada tabla está compuesta por filas y columnas, donde cada fila representa una instancia de datos y cada columna representa un atributo específico. La relación entre las tablas se establece mediante claves primarias y claves foráneas.

Esta se maneja mediante el lenguaje SQL (Structured Query Language)

**Definición base de datos no relacional**

Una base de datos no relacional es aquella que no utiliza el esquema de tablas y filas, en su lugar utilizan modelos optimizado para los requisitos específicos del proyecto

Un ejemplo son los documentos json que se pueden almacenar como clave valor

Además estos modelos tienden a ser más específicos con los tipos de datos que admiten

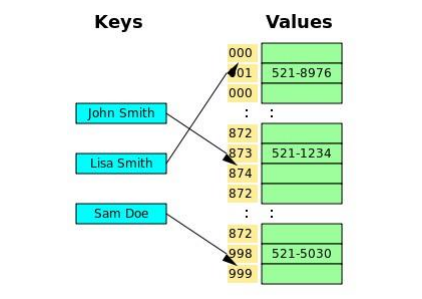
Este por lo general es un tipo de base de datos distribuida, es decir que la base de datos se almacena en diferentes servidores

**Tipos de modelos**

Existen diferentes tipos de modelos de datos en una base de datos no relacional:

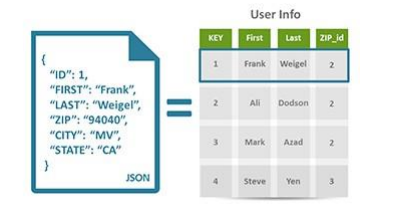
-Almacén de pares clave valor

Este se organiza en un diccionario de pares clave valor donde cada elemento tiene una clave y un valor asignado, la clave debe ser única similar a por ejemplo un id de cualquier tabla en una base de datos relacional, mientras que el valor es una matriz de datos pude ser cualquier atributo de la tabla en cuestión



-Almacén de documentos

Este almacena los datos como documentos, estos se almacenan en formatos como JSON o XML y se utiliza una clave única para cada registro

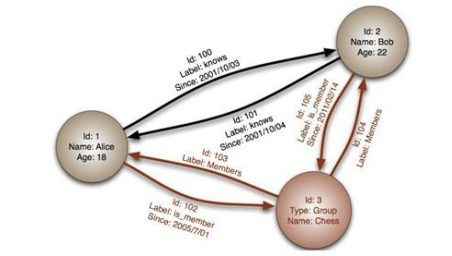


-Almacén de datos distribuido en columnas

Este almacena los datos en columnas lo cual permite acceder a ciertos datos concretos sin necesidad de cargar los demás

-Almacén de grafos

Este se representa la información como nodos de un grafo y las relaciones como aristas de los mismos, cualquier objeto ubicación o persona puede ser un nodo, un ejemplo de ello puede ser un cliente y una empresa las cuales serian nodos y una arista seria categoriza la relación como cliente entre los 2 nodos



|  |  |  |
| --- | --- | --- |
| Comando | SQL | NoSQL |
| Crear tabla | Create table | db.createCollection () |
| Insertar datos | Insert into | db.collection\_name.insertOne() |
| Actualizar datos | Update | db.collection\_name.updateOne() |
| Eliminar datos | Delete from | db.collection\_name.deleteOne() |
| Modificar tabla | Alter table |  |
| Eliminar tabla | Drop table |  |
| Ordenar | Order by | db.collection\_name.find().sort() |
| Menor que | SELECT \* FROM table\_name WHERE column **<** value; | db.collection\_name.find({ field: { **$lt**: value } }); |
| Mayor que | SELECT \* FROM table\_name WHERE column **>** value; | db.collection\_name.find({ field: { **$gt**: value } }); |
| Igual | SELECT \* FROM table\_name WHERE column **=** value; | db.collection\_name.find({ field: { **$eq**: value } }); |
| And | SELECT \* FROM table\_name WHERE condition1 AND condition2; | db.collection\_name.find({ $and: [ { condition1 }, { condition2 } ] }); |
| Or | SELECT \* FROM table\_name WHERE condition1 OR condition2; | db.collection\_name.find({ $or: [ { condition1 }, { condition2 } ] }); |
|  |  |  |

Bibliografía

<https://www.acens.com/comunicacion/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>

<https://www.ibm.com/es-es/topics/nosql-databases>