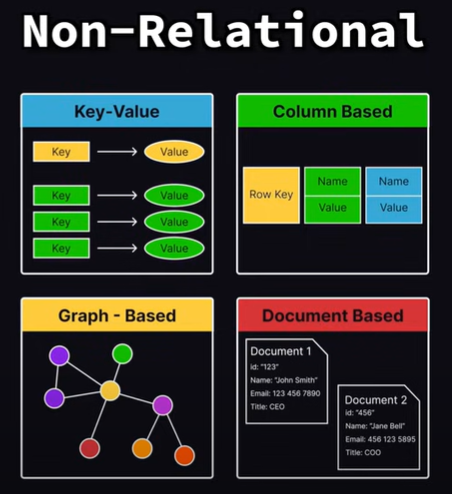
Primero, un database es una colección de datos y se usa para guardar a veces cantidades masivas de datos organizados tabular o no tabularmente, o en otras palabras, existen bases de datos relacionales y no relacionales.

Las bases de datos relacionales son tablas con filas y columnas básicamente. Mientras que las no relacionales pueden estar en forma de grafos, documentos, pares de clave valor (tipo diccionarios) y en columnas.

Relacional:



No Relacional:



Ahora llegamos a los query, un query es una consulta que le haces a la base de datos, esta consulta te permite hacer operaciones CRUD (Create, Read, Update and Delete).

Ahora, dónde demonios se guardan estos databases? Porque para guardar la información necesitas un lugar para almacenarlo no? Bien, puedes correrlo localmente (En tu compu o PC) o puedes correrlo en un servidor, en este curso vamos a correr el database localmente, esta manera de guardar los databases se usa normalmente para testear y desarrollar, de manera que no haces cambios a los datos reales. Las bases de datos pueden estar localizadas en:

* Servidores de la compañía, donde la compañía tiene control por completo del servidor y se le llama On-Prem.
* Servidores en la nube, donde otra empresa hace esto por tí, empresas como Google (GCP), Amazon (AWS) o Microsoft (Azure) tienen sus servicios de la nube, a estos se le llama Serverless.

Recuerda que las bases de datos relacionales almacenan datos en forma de filas y columnas y se usan normalmente para aplicaciónes transaccionales, todas estas tablas tienen una columna de IDs para poder relacionar ambas tablas.

Las bases de datos no relacionales también se les llama NOSQL que se refiere a Not Only SQL.

Dentro de las bases de datos más usadas en el mundo tenemos el ranking:

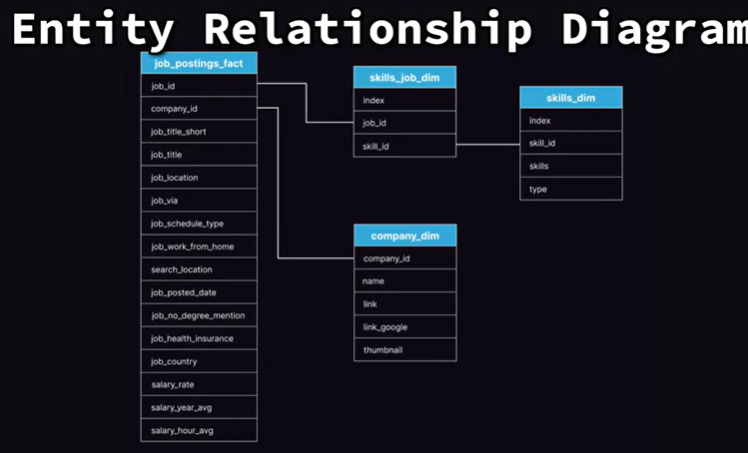
1. PostgreSQL
2. MySQL
3. SQLite
4. MongoDB
5. Microsoft SQL Server

En ese orden, donde 4 de 5 son bases de datos relacionales.

Nuestro codigo lo podemos correr tanto en las plataformas de estos databases, como en una plataforma de la nube como Google Cloud o hasta en un IDE como VSCode.

En este caso usaremos VSCode para correr los queries y una plataforma de database que en este caso escogeremos a SQLite.

Este curso no solo será para informar y tener conceptos, vamos a tomar el rol de ser alguien que está cazando o investigando cuales son los mejores trabajos en el área de datos, que trabajos ganan más que otros y que habilidades técnicas pesan más.



Esto es un ERD, el ERD contiene tablas, en este caso la tabla de job\_postings\_fact es una fact table, las tablas de skills\_job\_dim y skills\_dim son tablas de dos dimensiones y finalmente una tabla de una dimensión que es la de company\_dim.

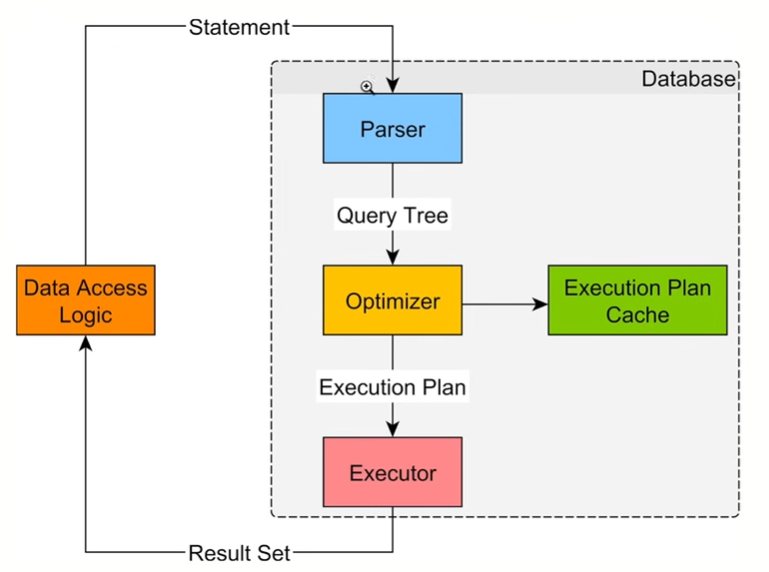
Una fact table contiene los datos centrales o el núcleo de los datos, estas tablas normalmente son las que tienen más columnas y tienen una llave foránea que las linkea a una tabla dimensional, como puedes ver con las lineas que company\_id y job\_id estan en la fact table y en las tablas de dimensiones.

Las tablas dimensionales describen atributos o dimensiones de los datos, estas son muy importantes para filtrar, agrupar y etiquetar hechos en los reportes, normalmente tienen menos datos.

Como puedes ver, en la tabla de job\_postings\_fact tienes una columna única que es job\_id, esta columna también está en la tabla de skills\_job\_dim como una clave foránea, este job\_id lo tenemos en esta tabla porque pueden haber distintos skill\_id relacionados a ese job\_id. Y de la misma manera, puedes ver que la tabla de skills\_dim tiene una llave foránea que es la columna de skill\_id de la tabla skills\_job\_dim. También puedes ver que la fact table tiene una columna que es company\_id que tiene una llave foránea en la tabla de company\_dim, para cada id de compañía hay una columna para el nombre, el link, el link de google y más.

Ahora viene la parte de hacer consultas al database usando un programa, que en este caso será SQLite, que es tan pequeño que hasta corre en el navegador, es el menos complicado de usar. Más adelante usaremos PostgreSQL pero por ahora empezaremos por otro más liviano. Ten en cuenta que SQLite y Postgres no tienen la misma sintaxis para algunas cosas, pero para la mayoría son iguales o muy parecidos.

Para que entiendas, cuando haces un query, el proceso funciona de esta forma:



Todo este proceso funcionará si tu código de SQL esta escrito asi:

