**Introducción a Consultas a bases de Datos**

**Sobre la importancia de las consultas (o Queries) en una base de datos relacional.**

Las consultas son la parte fundamental de una base de datos. De qué te sirve hacer una consulta o un Query efectivo?

Pues bien, una consulta bien hecha podría, incluso, *salvar un negocio o una empresa.* Las empresas tienen una gran necesidad por información correcta, exacta, oportuna; dicha información, se puede extraer generalmente de las bases de datos que ya tenemos (de aquí la importancia de las *consultas*).

En las bases de datos tenemos muchas veces *clientes, productos, proveedores* y diferentes tipos de entidades que, por separado, no tendrían mucho sentido; pero, cuando se comienzan a unir, a través de Queries, puedes presentar: un informe, o una tendencia, etc...; de tal forma que, las personas que están encargadas de *maniobrar* la empresa para un lado o para el otro,... tengan la información que necesitan para tomar las decisiones correctas.

Entonces, alrededor de toda esa cultura de “*extraer y transformar información”,* se han creado varias especialidades en el esquema de datos que tiene que ver con, por ejemplo, ***ETL*** (transformación de datos), ***Business Intelligent*** (extracción & utilización de datos para la toma correcta de decisiones) e, incluso, que tiene que ver con: ***Machine leearning*** (toma los datos existentes de una base de datos y los presenta de tal manera que haga sentido, que sea coherente y, sobre todo, que sea útil).

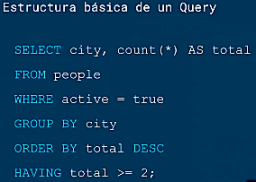
*En resumen, los Queries son una parte fundamental de una base de datos;* pues, nos permiten conocer las estructuras de los **Query** y resolver cualquier duda que tenga cualquier miembro de la empresa que pueda ser trabajada a través de una base de datos.

En resumen, con los ***Queries*** es que se brinda valor a una organización al final del día; pues, es lo que nos trae la información, *el diario vivir dentro de una organización*; con lo cual, es que se podrán tomar decisiones de gestión e inversión diariamente. Entonces, para traer toda esa información de nuestro interés, de una base de datos, necesitas hacer ***Queries.*** Los ***Queries*** son la forma en la que estructuramos las preguntas que le vamos a hacer a la base de datos; es decir, al final del día, nuestro camino a seguir para encontrar todas las respuestas.

Ahora, lo complicado es que hay muchas formas de hacer un ***Queri,*** desde lo más básico a lo más complejo.

***Vamos de las Queries básicas a las complejas.***

Empecemos a practicar la habilidad de convertir nuestras preguntas en ***Queries.*** El ***Queri*** tiene básicamente dos partes: la sentencia ***Select*** y la sentencia ***From*** y, en varias ocasiones, nos encontramos con la sentencia ***Where.***



***Select*** nos permite traer los datos que queremos mostrar, de qué ***columna***; en este caso puntual, queremos traer los datos de la columna o atributo ***city*** y una cuenta ***count(\*)*** la cual se proyecta como ***total*** (más adelante explicaremos más a fondo los dos últimos elementos señalados en negrilla).

En ***From*** indicamos de dónde vamos a seleccionar esos datos que queremos mostrar con el ***select;*** es decir, más precisamente, de que ***tabla***; en contexto, entonces, estamos indicando que queremos los datos de la columna o atributo ***city*** de la tabla ***people.***

Ahora, en ***Where,*** el filtro de la consulta y recolección de datos se vuelve más exigente; pues, con ***where,*** lo que se establece es una condición; en consecuencia, dicha condición debe cumplirse para poder seleccionar los datos deseados de una columna para una tabla en especifica. En contexto, cuando definimos un where con ***active = true*** le estás diciendo a nuestra base de datos de trabajo actual que desea solamente los datos de la columna ***city*** de la tabla ***people*** que cumplan con la condición ***active = true.*** De hecho, ***active*** vendría siendo también otra columna o atributo de la misma tabla ***people*** y ***true*** es uno de los posibles valores que puede tener, dentro de dicha tabla, la columna ***active.***

***Group by*** lo que nos permite es agrupar los datos, justamente, según un criterio pasado; en este caso, el criterio pasado fue: ***city;*** es decir, vamos a agrupar los datos de la tabla ***people***, que cumplen con la condición ***active = true*** y son de la la columna o atributo ***city,*** dentro de la misma columna ***city*** (nos mostrará la info de ***city***).

***Order by*** define de qué manera se organizarán los datos recolectados; en este caso, nos ayuda a definir de qué manera ordenar los datos seleccionados de la columna ***city***, de la tabla ***people***, que cumplen con la condición ***active = true*** (la totalidad de ellos); por ejemplo, definir el orden de los datos seleccionados de tal manera que: se *enlisten de mayor a menor* las ciudades ***(city)*** según la cantidad de personas que tenga cada una.

Por último, en ***Having,*** sedefine otro parámetro de limitación, va a aplicar otro filtro o condición mucho más especifico; en este caso, va a filtrar (elegir) el ***total*** de los datos seleccionados de la columna ***city***, de la tabla ***people*** y que tengan un ***active = true;*** pero, además, *cuando haya 2 o más de 2 personas en cada ciudad* ***(total >= 2).*** Tengo la sensación que ***total*** es otra columna o atributo de la tabla ***people.***

***un dato:*** La sentencia ***Select \**** selecciona todos los datos. Por ejemplo, evaluemos nuestro primer caso de ***Query:***



Con esto estamos diciendole a la base de datos que nos traiga absolutamente todo ***(SELECT \*)*** de la tabla ***posts (FROM posts),*** que nos muestre absolutamente todo de ahí (nos muestra todos sus campos o atributos).

Y ahora sí, generalmente, te encuentras las ***Queries*** con una tercera sentencia declarada que es, como te contabamos, ***Where;*** que nos permite filtrar cuáles de esos datos, de la tabla ***posts***, queremos ver.



En este caso, sólo hemos filtrado los datos de la tabla ***posts*** que tienen como fecha de publicación una fecha igual o superior al año ***2024.***

Es decir, la base de datos nos arrojará todas las noticias, ***posts,*** que sean del ***2024*** para ***adelante*** en su ***fecha\_publicacion.***

***Nota:*** *para SQL “>” es igual a “>=”, por lo que en el caso anterior se incluye también al año 2024. Sin embargo, “<” no es lo mismo que “<=”.*

En adelante, probaremos ***Queries*** más complejas según el nivel de complejidad que tenga, propiamente, la consulta de los datos que deseas ver; que, obviamente, deberían de ser en la medida unos datos con unas caracteristicas más puntuales o mucho más especificas.