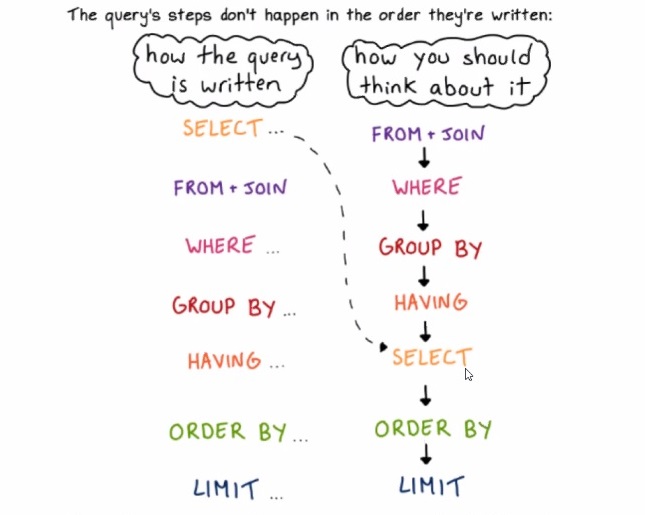
**SQL INCLASS NOTLARI**



* Önce **FROM**'u okur. ***Hangi tabloya gideceğini belirler***.
* Sonra  **WHERE**'i okur; ***gittiği bu tablodan hangi filtrelemeyi yapacağını belirler.***
* Sonra varsa **GROUP BY** ile ***gruplama***yapar.
* **Aggragate**fonsiyonlar için **GROUP BY**kullanılıyorsa **filtrelemeyi HAVING**ile yapıyoruz**. HAVING, Aggregate**fonksiyonların filtrelenmesinde kullanılır. (**Tablo içindeki sütunlarda filtreleme yaparken WHERE kullanıyoruz, Aggregate fonksiyonlar için ise HAVING..)**
* **SELECT**'i ise bütün bunlardan sonra okur. Artık ***şekillenen tablodan hangi alanı getirmek istiyorsanız onu SELECT'te belirtirsiniz.***
* **ORDER BY** ve **LIMIT** ise select ile belirlenen çıktının ***hangi sırayla getirileceğini*** ve hangi satırlarının ***limitlenip***getirileceğini belirliyor.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Birden fazla satıra işlem yapan AVG, SUM, COUNT vs. kullanırken önüne bir sütun ismi yazmıyoruz. Çünkü yazarsak o sütunun bir satırına karşılık değer geleceği için tek satırı alır.**

**Örnek; COUNT tek satır sonuç döndürdüğü için önüne yazacağın sütun için tek satırı alır.**

**Aggragate fonsiyon kullanırken buna dikkat et. Ama MIN, MAX’ın önüne sütun ismi girebilirsin.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Aggregate**fonksiyonları **yalnızca bir veri**üretir.

**SUM**ve **AVG**, yalnızca **integer**değerlere uygulanır.

**COUNT \***, **NULL**dahil **tüm satırları sayar**.

**COUNT (colum\_name)** şeklinde sütun ismi girilirse **NULL’lar hariç satırları sayar.**

**COUNT \***hariç diğer hepsi **(min, max vs.) Null ları göz ardı eder.**Zaten**MIN, MAX**vs. de**\* ile kullanma şekli yok.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**SQL, WHERE’i Aggregate fonksiyonundan önce çalıştırır ki önce ANA TABLO ÜZERİNDE filtreleme yapıp, filtrelenmiş veriye fonksiyon işlemi uygulansın.**

**Böylece sistemi yormamış oluyor.GROUP BY da WHERE’den sonra çalıştırılıyor. Aynı nedenden dolayı.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Hatırlayın biz EDA yaparken hep verimizi gruplayıp bir aggregate uyguluyorduk. Çünkü genellikle **verimizi kategorilere göre analiz etmek isteriz.**

SQL'de, tıpkı Pandas'ın **groupby()**metodunda olduğu gibi **GROUP BY** ifadesi; aynı kategori value'suna sahip satırları özet satırlarında gruplandırır.

Böylece daha iyi analiz için veriyi ayrı gruplara bölüp hesaplama yapabiliyoruz.

**df.groupby("company")[[quantity]].mean()**  
&  
**SELECT company AVG(quantity) FROM bla-bla GROUPBY company;**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**GROUP BY, aggregate function’ı çağırmadan önce sonuçları gruplandırır. Bu, tüm sorgu yerine gruplara aggregate func.  uygulamanıza olanak tanır.**

* **WHERE cümlesi, aggregate’ten önceki ana tablodaki veriler üzerinde çalışır.**
* **WHERE cümlesi, GROUP BY cümlesinden önce çalışır. Dolayısıyla ana tablomuzun sadece WHERE cümlesindeki şartları sağlayan satırları gruplanır.**
* **ORDER BY, GROUP BY’dan sonra gelir. (Sonucu sıralar.)**

1. **Yani ilk önce WHERE şartı uygulanıp data filtrelenir,**
2. **sonra üzerine GROUP BY uygulanır,**
3. **sonra aggregate function uygulanır.**
4. **Son olarak da sonuç ORDER BY a göre sıralanır.**

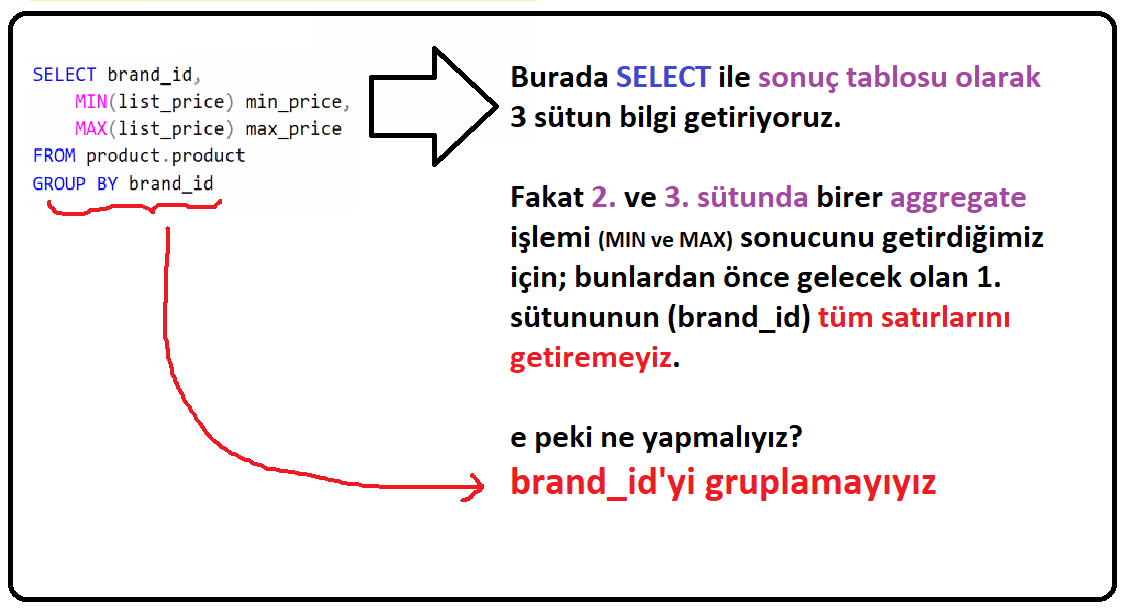
***çalışma sütunlarını +***

***koşula göre istenen satırları seç +***

***bu satırları kullanılacak fonksiyona göre gruplandır +***

***sıralamasını belirle***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**DİKKAT!!**

**WHERE**ile ana tabloda bir filtreleme yapıyoruz. Ana tablo içinde herhangi bir filtreleme yapmayacaksan WHERE satırı kullanmayacaksın demektir.

**ORDER BY,** SELECT'ten sonra çalışıyor. Dolayısıyla SELECT'te yazdığım Allias'ı kabul eder!

**SELECT** satırında yazdığın sütunların hepsi **GROUP BY**'da olması (önceden gruplanmış olması) gerekiyor!

**ORDER BY** satırındaki ilk parametre (örneğin) 2 ise bu SELECT satırındaki 2. sütuna göre sırala demektir.

**HAVING** ile; query ile dönen sonuç tablosu üzerinde bir filtreleme yapıyoruz.

**HAVING** ile sadece aggregate sonucuna bir filtre uyguluyoruz. Dolayısıyla HAVING, GROUP BY ile birlikte kullanılıyor.

**HAVING**’de kullandığın sütun, aggregate te kullandığın sütunla aynı olmalı.

**HAVING,** SELECT’ten önce çalıştığı için daha sonra SELECT statement’ta oluşacak olan sütunların **allias**’ı HAVING’te **KULLANILAMAZ**. Bizzat kendisi kullanılmalı.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*