

SQL (Structured Query Language):

- **DML (Data Manipulation Language)** (SELECT, INSERT, UPDATE, DELETE, MERGE)
- **DDL (Data Definition Language)** (CREATE, ALTER, DROP, RENAME, TRUNCATE)
- **DCL (Data Control Language)** (GRANT, REVOKE)
- **TCL (Transaction Control Language)** (COMMIT, ROLLBACK, SAVEPOINT)

- PL/SQL (Oracle)
- T-SQL (Microsoft)
- PL/pgSQL (PostgreSQL)
- SQL-PL (IBM DB2)

Oracle için IDE'ler:

- SQL*Plus
- SQL Developer
- TOAD
- Embarcadero
- SQL Worksheet

SQL Developer Üzerinde Script Çalıştırmak için:

- F5 ile tüm sayfa
- CTRL + Enter ile seçili bölüm
- ✓ Scriptler noktalı virgül (;) ile tamamlanır.

Yorum Satırı Ekleme için:

- --Tek satırlık yorumlar
- /*çok satırlı yorumlar*/

Nesneleri İfade Etmek için:

- Tam Yol: DatabaseName.SchemaName.ObjectName
- Kısa Yol: ObjectName

Nesne Hakkında Bilgi Almak için:

- DESCRIBE ObjectName;
- DESC ObjectName;

Akılda Tutulacaklar:

- ✓ Schema ve User Oracle'da aynı kabul edilir. User kendi nesnelere kısa yol ile erişebilir.
- ✓ Metin tipindeki veriler tek tırnak ile, özel karakter içeren nesnelere çift tırnak ile yazılır.
- ✓ Nesnelere NesneAdı as TakmaAd şeklinde isim verebilir. as kelimesi zorunlu değildir.
- ✓ ROWNUM isimli yapay kolon ORDER BY'dan önce devreye girer ve satırlar numaralandırır.
- ✓ ROWID isimli yapay kolon satırın kimlik değerini verir. Güncelleme ile değer değişebilir.
- ✓ Oracle'da tek satırlı ve kolonlu olup, dummy 'X' kaydı tutan DUAL isimli bir tablo mevcuttur.
- ✓ NULL yokluk anlamına gelen özel bir değerdir. NULL, NULL'a veya başka şeye eşit değildir.

Operatörler:

- Parantez ile operatörlerin işlem önceliği belirtilebilir.

Karşılaştırma	Mantıksal
=	Eşittir
<	Küçüktür
>	Büyükür
<=	Küçük Eşittir
>=	Büyük Eşittir
<>, !=, =	Eşit Değildir
ANY, SOME	Herhangi birinden
ALL	Hepsinden
LIKE	Metinsel arama
EXISTS	Satır mevcut
	Üyelik
	IN(Degerler)
	NOT IN(Degerler)
	Matematiksel
	Değerlerin üyesi
	Üyesi değil değil
	*, -, *, / Dört işlem
	% Mod (5%2=1)
	NULL Kontrolü
	IS NULL
	IS NOT NULL
	Aralık belirtme ve Birleştirme
	BETWEEN x AND y
	x ile y arası
	Metin birleştirir

Küme (Aynı kolon yapısına sahip iki sorgu için)	
UNION	İki sorgu sonucunun birleşim kümesini verir.
UNION ALL	Tekrar edilebilir şekilde İki sorgu sonucunun birleşim kümesini verir.
INTERSECT	İki sorgu sonucunun kesişim kümesini verir.
MINUS	İki sorgu sonucunun fark kümesini verir.

JOIN Çeşitleri:

- JOIN'ler iki tablonun kolonlarını yan yana yazma imkanı verir.
- INNER JOIN, LEFT JOIN, RIGHT JOIN ve FULL JOIN türleri ON ile belirtilen şarta göre çalışır.

JOIN Çeşitleri	
INNER	Her iki tablodan sadece şarta uyan satırlar gelir.
LEFT	Soldaki tablonun tüm satırları gelir. Sağdakinden şarta uyanlar gelir.
RIGHT	Sağdaki tablonun tüm satırları gelir. Soldakinden şarta uyanlar gelir.
FULL	Tablolardaki tüm satırlar gelir. Şarta uyanlar eşleşir. Uymayana NULL atanır.
CROSS	Tablolann her bir elemanı diğerinin tüm elemanları ile eşleşir.

SELECT ... FROM

- ObjectName kümesinden tüm kolonlar gelsin.
- SELECT * FROM SchemaName.ObjectName;**
- ObjectName kümesinden virgüllerle belirtilen kolonlar gelsin.
- SELECT Column1, Column2, Column3 FROM SchemaName.ObjectName;**
- /* Oracle'da FROM ile bir veri kaynağı işaret etmeden SELECT kullanılamıyor. SELECT ile değer yazdırmak için DUAL tablosu kullanılabilir. PROMPT komutu da ekrana değer yazdırmak için kullanılabilir.*/
- SELECT value;** ✗ **SELECT value FROM DUAL;** ✓ **PROMPT value;** ✓
- DISTINCT ile belirtilen kolonlardan oluşan mükerrer satırları eleyebiliriz.
- SELECT DISTINCT Column1, Column2 FROM ObjectName;**

SELECT ... FROM ... WHERE

- Price alanı 10 ile 100 arasında veya Color alanı Red, Gray olan satırlar gelsin.
- SELECT Name, Color, Price FROM Products**
- WHERE (Price >= 10 AND Price <= 100) OR (Color = 'Red' OR Color = 'Gray'); --alternatif
- WHERE BETWEEN 10 AND 100 OR Color IN('Red','Gray');**
- /* Price alanı 0'dan farklı olup Color alanı NULL olmayan satırlar gelsin. Aynı şeyi NOT ile yazmak istersek kapsadığı tüm operatörlerin tersini alırız.*/
- SELECT Name, Color, Price FROM Products**
- WHERE Price != 0 AND Color IS NOT NULL; --alternatif
- WHERE NOT(Price = 0 OR Color IS NULL);**

ORDER BY ... OFFSET ... FETCH

- Akılda Tutulacaklar:**
- ✓ **ORDER BY** belirtilen alanlara göre sıralamak için kullanılır. En sona yazılır.
 - **ASC:** a-z veya küçükten büyüğe (varsayılan)
 - **DESC:** z-a veya büyükten küçüğe sıralanabilir.
- ✓ Kolon adı, takma ad veya kolonun SELECT'teki yazım sırası kullanılabilir.
- ✓ Sıralanması istenen alanlarda NULL varsa:
 - **NULLS FIRST** ile NULL değerleri sıranın başına
 - **NULLS LAST** ile NULL değerleri sıranın sonuna alabiliriz.
- ✓ Dilerse **OFFSET** ile x sayıda satır atlayıp, **FETCH** ile sonraki y saydaki satırı çekebiliriz.

- Kullanılan renkleri, NULL'lar başta olacak şekilde alfabetik olarak sıralayalım.
- SELECT DISTINCT Color FROM Products**
- ORDER BY Color ASC NULLS FIRST;**
- /* Price alanı 0'dan büyük olanları Price alanına göre büyükten küçüğe, sonra Color ve ardından Name alanına göre küçükten büyüğe sıralayalım. En yüksek fiyata sahip 5 ürün getirelim.*/
- SELECT Name, Color as Renk, Price "Urun Fiyat" FROM Products**
- WHERE Price > 0**
- ORDER BY**
- 3 DESC** --3. sırada belirtilen Price kolonuna göre DESC sıralar.
- ,Renk ASC** --Sonra takma ad ile Color kolonuna göre ASC sıralar.
- ,Name;** --Sonra kolon adı ile Name kolonuna göre ASC sıralar.
- OFFSET 0 ROWS** --Sıralanmış sonuçtan 0 satır atlar. OFFSET kullanılmayabilir.
- FETCH FIRST 5 ROWS ONLY;** --Elde edilen sonuçtan sadece ilk 5 satırı getirir.

LIKE, REGEXP_LIKE

- Akılda Tutulacaklar:**
- ✓ **LIKE** bazı özel karakterler (wildcard) ile oluşturulan desene uyan metinler için **TRUE** üretir.
 - % sıfır veya daha fazla sayıda karakteri temsil eder.
 - _ tek bir karakteri temsil eder.
 - Metin içinde özel karakter aramak için **ESCAPE** ' ' ile tanımlanan kaçış karakteri kullanılır.
- ✓ **REGEXP_LIKE(kolon, desen, eşleştirme_şartı)** fonksiyonu regex desentleri kullanır.
 - Büyük küçük harfe duyarlı olması için eşleşme şartı 'i', duyarlı olması için 'c' verilir.

- ✓ **WHERE Name LIKE '_m%'** --2. karakter 'm', diğerleri opsiyonel. örn: Emma, Summer, İmge
- ✓ **WHERE Name REGEXP_LIKE(Name, '^a.{1}[a-k].{1}m'i')** --regex deseni ile: örn: Autumn

GROUP BY ... HAVING

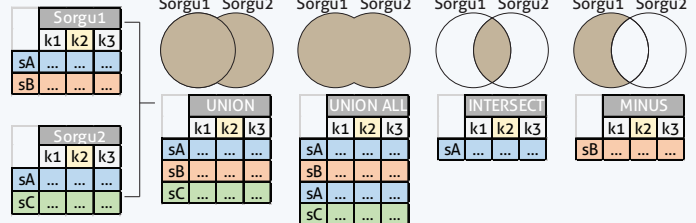
- Akılda Tutulacaklar:**
- ✓ **Gruplama fonksiyonları** yanında **SELECT** edilecek kolonlar(exp) **GROUP BY**'da belirtilmelidir.
- ✓ **GROUP BY** da belirtilen kolonların tümünü **SELECT** etmek zorunlu değil.
- ✓ **WHERE, FROM** da belirtilen kaynağı; **HAVING**, gruplanmış sorgu sonucunu filtreler.

Gruplama Fonksiyonları					
COUNT(*)	Satır adedi	SUM(exp)	Toplam	AVG(exp)	Ortalama
COUNT(exp)	NULL olmayanlar	MAX(exp)	En büyük	MEDIAN(exp)	Ortakdaki
COUNT(DISTINCT exp)	Farklıların adedi	MIN(exp)	En küçük	STDDEV(exp)	STD sapma

- Türkiye'deki çoklu siparişlerin OrderNumber alanına göre temel istatistiklerini getir.
- SELECT**
- OrderNumber,** --Gruplama fonksiyonlarıyla çağırılan alanlar **GROUP BY**'da belirtilmeli.
- SUM(SalesAmount) as "Toplam",**
- AVG(SalesAmount) as "Ortalama",**
- MAX(SalesAmount) as "Max",**
- MIN(SalesAmount) as "Min",**
- STDDEV(SalesAmount) as "Standart Sapma",**
- COUNT(*) as Adet**
- FROM FactInternetSales**
- WHERE Country = 'Türkiye'** --FactInternetSales satırları önce bu alana göre filtrelenir.
- GROUP BY OrderNumber** --Sonra satırları OrderNumber alanına göre gruplar.
- HAVING COUNT(*) > 1** --Sonra grubu filtreler. OrderNumber 1'den fazla ise çoklu sipariştir.
- ORDER BY 7 DESC;** --En son 7. kolona yani Adet kolonuna göre büyükten küçüğe sıralar.

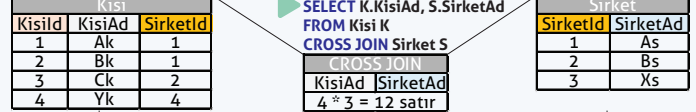
UNION, UNION ALL, INTERSECT, MINUS

- Akılda Tutulacaklar:**
- ✓ Her iki sorgunun da kolon yapıları (kolon sayısı, sırası ve veri tipleri) aynı olmalıdır.



- Sorgu1 sonuç kümesi satırları ile Sorgu2 sonuç kümesi satırlarının birleşim kümesini getir.
- SELECT k1, k2, k3 FROM Sorgu1**
- UNION** --UNION ALL, INTERSECT, MINUS operatörleri ihtiyaca göre tercih edilir.
- SELECT k1, k2, k3 FROM Sorgu2**

INNER JOIN, OUTER JOINS, CROSS JOIN



- SELECT K.KisiAd, S.SirketAd**
- FROM Kisi K**
- CROSS JOIN Sirket S**
- CROSS JOIN**
- KisiAd | SirketAd**
- 4 * 3 = 12 satır**
- SELECT K.KisiAd, S.SirketAd**
- FROM Kisi K**
- INNER JOIN Sirket S**
- ON K.SirketId = S.SirketId**
- SELECT K.KisiAd, S.SirketAd**
- FROM Kisi K**
- RIGHT JOIN Sirket S**
- /*FROM Sirket S --alternatif**
- LEFT JOIN Kisi K*/**
- ON K.SirketId = S.SirketId**
- RIGHT JOIN**
- KisiAd | SirketAd**
- Ak | As**
- Bk | Bs**
- Ck | Bs**
- Yk | NULL**
- NULL | Xs**