

Veri Bilimci Adayları için Veritabanı Eğitimi

Temel SQL Komutları

SELECT & WHERE

SELECT

- Veri tabanından bir kolonu seçmek için kullanılır
- Sonuçlar tablo şeklinde gösterilir
- Seçilen sütunların hangi tablodan olduğu **FROM**'dan sonra belirtilir

** = tüm sütunları seç anlamına gelmektedir*

WHERE

- Tüm tabloyu almak yerine yalnızca belirli bir koşulu karşılayan veriler alınır, kayıtlar filtrelenir
- Sadece SELECT ifadesi için değil, UPDATE ve INSERT gibi ifadelerde de kullanılır

```
SELECT column1, column2 FROM table1 WHERE column1 < 100;
```

Operatörler

Aritmetik Operatörler:

- Matematiksel koşulları sembolize ederler
- +, -, *, /, %, gibi

Karşılaştırma Operatörleri:

- İki değer arasındaki eşit/eşitsizlikleri ifade etmekte kullanılırlar
- =, !=, >, <, ≥, ≤, gibi

Boolean Operatörler:

- Birden fazla koşulun sağlanıp sağlanmaması gerektiğini belirtmek için kullanılırlar
- and, or, not, ...

Arithmetic Operators	Relational Operators	Logical Operators	Special Operators
+	=	And	In ,Not In
-	!= or <>	Or	Between,Not Between
*	>	Not	Like ,Not Like
/	<		Is null , Is Not Null
	>=		
	<=		

IN & LIKE

IN - NOT IN:

- IN operatörü, WHERE'deki bir dizi olası değeri eşleştirmek için kullanılır
- İlgili sütunda IN (NOT IN) ile verilen kümenin elemanlarıyla aynı değere sahip olan (olmayan) kayıtlar gösterilir

LIKE:

- LIKE operatörü, bir sütun değerindeki belirli bir deseni eşleştirmek için WHERE'den sonra kullanılır

Not: '%' sembolü 0 veya daha fazla karakteri sembolize etmek için kullanılır

```
SELECT col1, col2, FROM table1
WHERE col1 IN ('Anakin', 'Ahsoka', 'Kenobi') and col2
LIKE '%sw%';
```

ORDER BY & LIMIT

ORDER BY (DESC):

- SQL, belirtilmediği sürece, verileri belirli bir sırayla görüntüleyemez
- ORDER BY varsayılan olarak verileri küçükten büyüğe listeler
- Kayıtları büyükten küçüğe listelemek için sonuna DESC eklenmelidir

LIMIT:

- Sonuç olarak gösterilecek olan kayıt sayısını sınırlar

Örn: LIMIT 10, sonuçlarda sadece ilk 10 kaydı gösterir

```
SELECT col1, col2, col3 FROM table1
ORDER BY col1
DESC LIMIT 10;
```

SELECT DISTINCT & GROUP BY:

SELECT DISTINCT:

- Bir sütunda yalnızca farklı değerlere sahip olan kayıtları döndürür
- Sonucunda belirtilen sütun için tüm eşsiz değerlere sahip kayıtlar bulunur

```
SELECT COUNT(*) FROM pokemon; --count  
fonksiyonu
```

```
SELECT DISTINCT lastName FROM Employee;
```

GROUP BY:

- Sonuçtaki satırları bir veya daha fazla sütuna göre gruplandırmak için kullanılır
- **COUNT(*)** fonksiyonuyla beraber hangi değerden kaç adet olduğu listelenebilir

```
SELECT classification, COUNT(*)  
FROM pokemon  
GROUP BY classification;
```

Tablo Kısıtlamaları & Alt Sorgular (Subselects)

Tablo Kısıtlamaları:

- Sütunların Primary, Unique, Foreign Key atamaları; NULL/NOT NULL belirtilmesi ve DEFAULT değerleri SQL kullanılarak da belirtilebilir

Alt Sorgular:

- İç içe geçmiş SELECT sorgularıdır
- Bir SELECT sorgusunun sonuçları başka bir SELECT sorgusu için kullanılır

INSERT INTO & UPDATE

INSERT:

- Tabloya yeni bir veya daha fazla kayıt eklemek için kullanılır
- Eklenecek kayıt için herhangi bir sütun değeri verilmemişse, o hücre varsayılan değer veya NULL değeriyle doldurulur

UPDATE:

- Bir tablodaki mevcut satırlardaki verileri değiştirmek için kullanılır
- Yeni değerler **SET** anahtar sözcüğü kullanılarak atanır
- Bir sütunu **NULL** 'a eşitlemek de mümkündür

```
INSERT INTO himym_table (firstname, lastname)
VALUES ('Ted', 'Mosby'), ('Barney', 'Stinson'), ('Marshall', 'Eriksen');
```

```
UPDATE EmployeeTable
SET salary = 7000, address = '34381, Şişli/İstanbul'
WHERE id = 1234;
```

DELETE FROM & TRUNCATE

DELETE FROM:

- Bir tablodan bir veya daha fazla satırı silmek için kullanılır
- Biçim; DELETE FROM, ardından bir tablo adı ve genellikle WHERE ile belirtilen bir koşuldur
- Tablodaki bütün satırları silmek için WHERE kullanmaya gerek yoktur

```
DELETE FROM employee_table WHERE  
firstName = 'Thomas'
```

TRUNCATE:

- Bir tablodaki bütün kayıtları silmek için kullanılır
- Kullanıldığında primary key olarak kullanılan sütun değerleri de sıfırlanır

```
TRUNCATE employee_table
```

CREATE & DROP

CREATE DATABASE & CREATE TABLE:

- Bir veri tabanı veya tablo oluşturmak için kullanılır
- Tablo oluşturulacağı zaman sütunların adı, veri türleri ve diğer kısıtlamaları parantez içinde belirtilir

Halihazırda var olan bir veritabanı (veya tablo) oluşturmaya çalışmak, IF NOT EXISTS kullanılmazsa hatayla sonuçlanır

DROP DATABASE & DROP TABLE:

- Bir veri tabanını veya tabloyu silmek için kullanılır

Aynı şekilde, bir veritabanı oluştururken, çıkmayan bir veritabanını (veya tabloyu) silmeye çalışmak, IF EXISTS kullanılmadığı zaman hatayla sonuçlanır

CREATE ile INDEX İmplementasyonu:

CREATE TABLE işlemi sırasında:

```
CREATE TABLE books (  
    title VARCHAR(255),  
    INDEX index_name (title)  
);  
  
CREATE TABLE books (  
    title VARCHAR(255),  
    UNIQUE INDEX index_name (title)  
);
```

Tablo oluşturulduktan sonra:

```
CREATE INDEX index_name ON  
table_name (column_name);  
  
CREATE UNIQUE INDEX index_name ON  
table_name (column_name);
```

Birçok sütun kullanarak index oluşturma:

```
CREATE INDEX index_name ON  
table_name (column_A, column_B,  
...);
```

ALTER & DROP ile INDEX İmplementasyonu:

ALTER:

```
ALTER TABLE tbl_name ADD INDEX  
index_name (col_name);
```

```
ALTER TABLE tbl_name ADD UNIQUE  
INDEX index_name (col_name);
```

DROP:

```
DROP INDEX index_name ON tbl_name;
```

```
ALTER TABLE tbl_name DROP INDEX  
index_name;
```

Transaction İmplementasyonu

START TRANSACTION;

```
INSERT INTO books (title, author, price)
```

```
VALUES ('Database Systems', 'Christopher', 300);
```

```
UPDATE books SET price = 100 WHERE title LIKE '%Database%';
```

COMMIT; --ROLLBACK;

- COMMIT tablo üzerinde kalıcı olarak değişiklik yaparken; ROLLBACK geçici olarak değişiklikler yapmaktadır

Birden Fazla Tabloda Sorgulama

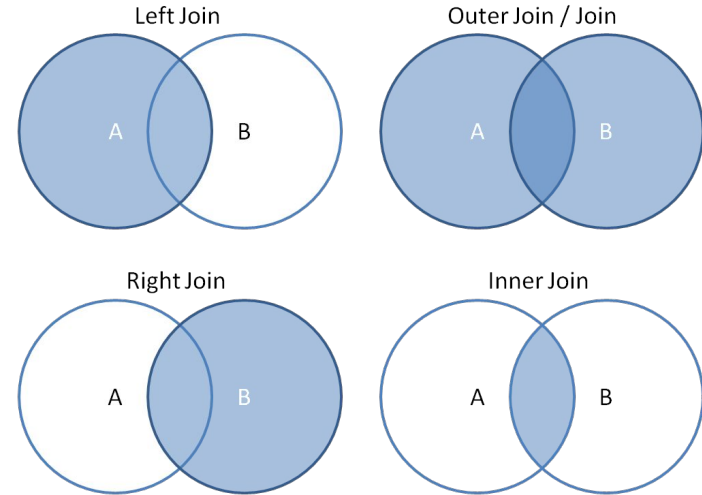
JOIN

INNER JOIN:

- Her iki tabloda da belirli eşleşen değerlere sahip kayıtları seçer

OUTER JOIN:

- Tabloların eşleşen değerleri dışında, bir tablonun tam olarak eşleşmeyen değerlerin de satırlarını içerir
- Eşleşmeyen satırların alınması sonucunda var olmayan değerler için NULL alınır.
- LEFT, RIGHT & FULL OUTER JOIN** olmak üzere 3 tip vardır



Python ile Veritabanı Uygulamaları