

8 April SQL

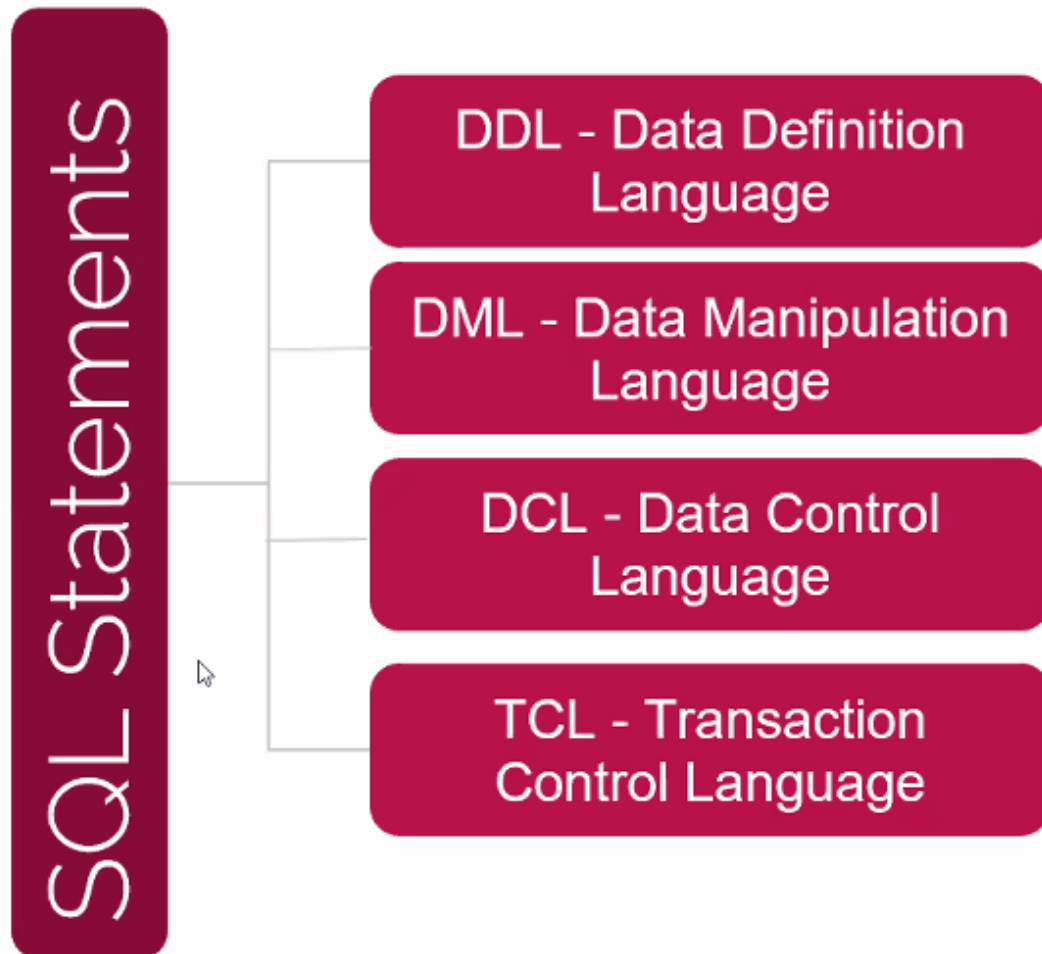
SQL 4 alt kategoriye bölmüşler.

DDL ⇒ Create(Oluştur), Alter(Değiştir), Drop(sil) (Tablo seviyesi)

DML ⇒ Insert, Update, Delete(KayıtSilme), Select(bazıları farklı gruplandırabiliyor.) (Veri seviyesi)

DCL ⇒ Revoke, Grant (tabloya erişim/silme vs. yetki verme-alma)

TCL ⇒ Commit, Savepoint, Rollback (commit, Server'a veriler iletilmiş mi? Rollback, geri alma. Savepoint, recovery noktası oluşturuyor.)



▼ Data Types

Integer, Numeric, character, date, time, binary,

STRING ⇒ CHAR, VARCHAR, BINARY, TEXT, SET, ENUM, BLOB, VARBINARY

DATE ⇒ DATE(2022-12-31), DATETIME(2022-12-31 23:50:00), YEAR(2022),
TIMESTAMP

NUMERIC ⇒ INTEGER(VEYA INT), -SMALLINT, TINYINT, MEDIUMINT, BIGINT-
(Aralarında kapasite tutma farkı var)0, DECIMAL, NUMERIC, FLOAT(32bit),
DOUBLE(64bit)

Create Table

```
CREATE TABLE table_name  
(column_name1 data_type,  
column_name1 data_type)
```

NUMERIC(5,2) ⇒ 3tam sayı 2 virgülden sonrası olabilir

NOT: SQL'de TRUNCATE TABLE komutu bulunmasına karşın SQLite bu komutu desteklememektedir. Truncate komutu bir tabloyu değil içindeki tüm verileri silmek için kullanılır.

▼ Insert Table

```
INSERT INTO table_name(column1, column2,,,)
VALUES(value1, value2...)
/* direk values yazarak değerleri girebiliriz. gireceğimiz değerlerin sırasını ve
değişkene uygun yazılması gerekir. Yoksa hata verir.)
```

▼ Constraints (kısıtlamalar)

Constraints

Constraint Name	Definition
NOT NULL	Ensures that a column cannot have a NULL value
DEFAULT	Sets a default value for a column when no value is specified
UNIQUE	Ensures that all values in a column are different
PRIMARY KEY	Uniquely identifies each row in a table
FOREIGN KEY	Uniquely identifies a row/record in another table

```
CREATE TABLE table_name(
  column_1 INT PRIMARY KEY,
  column_2 TEXT,
  ....);
-- iki çeşit kullanım vardır.
```

```
CREATE TABLE table_name(
  column_1 INT,
  column_2 TEXT,
  ....
  PRIMARY KEY (column_1)
);
-- PRIMARY KEY kullanımı iki çeşittir. Üstteki gibi de kullanılabilir. Ama genellikle
-- bu kullanım tercih edilir.
-- PRIMARY KEY boş bırakılamaz NOT NULL'dur.
```

```
CREATE TABLE orders(
  order_id INT PRIMARY KEY,
  order_number INT,
  customer_id INT,
  FOREIGN KEY (customer_id)
  REFERENCES customers (customer_id)
);
-- FOREIGN KEY kullanımı.
```

FK ve PK veri tipleri tutarlı olması gerekir.

NOT NULL ⇒ Boş bırakılamaz.

AUTOINCREMENT ⇒ Otomatik arttırarak değer verir. Primary key için kullanılabilir.

```
/*=====
DDL COMMANDS (CREATE TABLE, DROP TABLE,ALTER TABLE)
=====*/

/*-----
/* CREATE TABLE
/*-----

/*personel adında bir tablo oluşturunuz. Tabloda first_name, last_name
age(INT) ve hire_date (Date) sütunları bulunmalıdır. */
--yoksa oluşturur varsa hata vermez ama oluşturmaz.
CREATE TABLE personel(
    first_name TEXT,
    last_name TEXT,
    age INT,
    hire_date date
);

/* Aynı isimle yeniden bir veritabanı oluşturulmak istenirse hata verir. Bu hatayı
almamak için IF NOT EXISTS keywordu kullanılabilir */

CREATE TABLE IF NOT EXISTS personel(
    first_name TEXT,
    last_name TEXT,
    age INT,
    hire_date date
);

/*Veritabanında vacation_plan adında yeni bir tablo oluşturunuz. Sütun isimleri
place_id, country, hotel_name, employee_id, vacation_length,budget */

CREATE TABLE vacation_plan(
    place_id INT,
    country NVARCHAR(50),
    hotel_name NVARCHAR(100),
    employee_id INTEGER,
    vacation_length INT,
    budget REAL
);

/*-----
/* DROP TABLE
/*-----
/* personel tablosunu siliniz */

DROP TABLE personel;

/* Bir tabloyu silerken tablo bulunamazsa hata verir.
```

```

Bu hatayı görmemek için IF EXISTS keywordu kullanılabilir.*/
DROP TABLE IF EXISTS personel;

-- NOT: SQL'de TRUNCATE TABLE komutu bulunmasına karşın SQLite bu komutu
-- desteklememektedir. Truncate komutu bir tabloyu değil içindeki tüm verileri
-- silmek için kullanılır. SQLite'da bu komut yoktur.

INSERT INTO vacation_plan VALUES(34, "TURKEY", "HOTEL İSTANBUL", 1,7, 4000);
-- Birden fazla kez çalıştırırsak tekrar tekrar aynı satırları ekler. Bunun önüne geçmek için
-- PRIMARY KEY yapılması gerekir.
INSERT INTO vacation_plan VALUES(42, "TURKEY", "HOTEL MEVLANA", 2, 4, 2000);

/*vacation_plan tablosuna vacation_lenght ve budget sütunlarını eksik olarak veri girişi
yapınız*/

INSERT INTO vacation_plan (place_id, country, hotel_name, employee_id)
VALUES(06, "TURKEY", "HOTEL BAŞKENT", 3);

/* table_name'den sonra sütun isimlerini yazarak sonrasında sütun isimlerine uygun sırayla
VALUES içine değişken tiplerine uygun değerler yazılarak ekleme yapılır.*/

/*-----
CONSTRAINTS - KISITLAMALAR
-----

NOT NULL - Bir Sütunun NULL içermemesini garanti eder.

UNIQUE - Bir sütundaki tüm değerlerin BENZERSİZ olmasını garanti eder.

PRIMARY KEY - Bir sütunun NULL içermemesini ve sütundaki verilerin
BENZERSİZ olmasını garanti eder.(NOT NULL ve UNIQUE birleşimi gibi)

FOREIGN KEY - Başka bir tablodaki Primary Key'i referans göstermek için kullanılır.
Böylelikle, tablolar arasında ilişki kurulmuş olur.

DEFAULT - Herhangi bir değer atanmadığında Başlangıç değerinin atanmasını sağlar.*/

CREATE TABLE workers(
id INT PRIMARY KEY,
id_number TEXT NOT NULL UNIQUE,
name TEXT DEFAULT 'NONAME',
salary INTEGER NOT NULL
);

INSERT INTO workers VALUES(1,"12345678910","CAN CANAN", 10000);
-- UNIQUE CONST VIOLATION, unique olduğu için aynı id ile ekleme yapılamaz.
-- INSERT INTO workers VALUES(1,"44456789100","AHMET CANAN", 7000);
INSERT INTO workers VALUES(3,"55547896321", NULL, 9000);
-- name kısmına NONAME yazılır DEFAULT.
INSERT INTO workers (id,id_number, salary) VALUES(4,"55787876321", 9000);

/*vacation_plan tablosunu place_id sütunu PK ve employee_id sütununu
ise FK olarak değiştirilerek vacation_plan2 adında yeni bir tablo oluşturunuz.

```

Bu tablo, employees tablosu ile ilişkili olmalıdır */

```
CREATE TABLE vacation_plan2(
    place_id INT,
    country NVARCHAR(50),
    hotel_name NVARCHAR(100),
    employee_id INTEGER,
    vacation_length INT,
    budget REAL,
    PRIMARY KEY (place_id),
    FOREIGN KEY (employee_id) REFERENCES employees(EmployeeId)
);
```

```
/* Employees tablosundaki EmployeeId'si 1 olan kişi için bir tatil planı giriniz.*/
INSERT INTO vacation_plan2 VALUES(35, "TURKEY","HAPPY NATION", 9, 5, 5000);
-- HATA VERİR
```

/*JOIN işlemi ile 2 tablodan veri çekme*/

```
SELECT e.FirstName, e.LastName, v.length, v.hotel_name,
FROM employees e,
INNER JOIN vacation_plan2 v,
on e.EmployeeId = v.employee_id;
```

```
/*-----
/* ALTER TABLE (ADD, RENAME TO, DROP)
/* SQLITE MODIFY VE DELETE KOMUTLARINI DOĞRUDAN DESTEKLENMEZ
/*-----
```

/*vacation_plan2 tablosuna name adında ve DEFAULT değeri noname olan yeni bir sütun ekleyelim */

```
ALTER TABLE vacation_plan2
ADD name TEXT DEFAULT "ISIMSIZ",
```

```
/* workers tablosunun adını name sütununu siliniz*/
DROP COLUMN
```

/* workers tablosunun adını people olarak değiştiriniz*/

```
ALTER TABLE workers
RENAME to people;
```

```
*-----
/* UPDATE,DELETE
-- SYNTAX
-----
-- UPDATE tablo_adı
-- SET sutun1 = yeni_deger1, sutun2 = yeni_deger2,...
-- WHERE koşul;

--DELETE tablo_adı
--WHERE koşul;
/*-----*/
```

```

/*vacation_plan2 tablosundaki employee_id=1 olan kaydını hotel_name'ini Komagene Hotel olarak
güncelleyiniz.*/

UPDATE vacation_plan2
SET hotel_name = 'Komagene Hotel'
WHERE employee_id = 1;

/* people tablosunda salary sütünü 7000 'den az olanların salary(maaşına)
%10 zam yapacak sorguyu yazınız*/

UPDATE people
SET salary = salary*1.2
WHERE salary < 7000;

/*people tablosundaki tüm kayıtkarın salary sütununu 10000 olarak güncelleyiniz */
UPDATE people
SET salary=10000;

/*vacation_plan2 tablosundaki employee_id=1 olan kaydı siliniz*/
DELETE FROM vacation_plan2
WHERE employee_id=1;

/*=====
DDL COMMANDS (CREATE TABLE, DROP TABLE,ALTER TABLE)
=====*/

/*-----
/* CREATE TABLE
/*-----

/*personel adında bir tablo oluşturunuz. Tabloda first_name, last_name
age(INT) ve hire_date (Date) sütunları bulunmalıdır. */
--yoksa oluşturur varsa hata vermez ama oluşturmaz.
CREATE TABLE personel(
    first_name TEXT,
    last_name TEXT,
    age INT,
    hire_date date
);

/* Aynı isimle yeniden bir veritabanı oluşturulmak istenirse hata verir. Bu hatayı
almamak için IF NOT EXISTS keywordu kullanılabilir */

CREATE TABLE IF NOT EXISTS personel(
    first_name TEXT,
    last_name TEXT,
    age INT,
    hire_date date
);

/*Veritabanında vacation_plan adında yeni bir tablo oluşturunuz. Sütun isimleri
place_id, country, hotel_name, employee_id, vacation_length,budget */

CREATE TABLE vacation_plan(
    place_id INT,

```

```

country NVARCHAR(50),
hotel_name NVARCHAR(100),
employee_id INTEGER,
vacation_length INT,
budget REAL
);

/*-----
/* DROP TABLE
/*-----
/* personel tablosunu siliniz */

DROP TABLE personel;

/* Bir tabloyu silerken tablo bulunamazsa hata verir.
Bu hatayı görmemek için IF EXISTS keywordu kullanılabilir.*/
DROP TABLE IF EXISTS personel;

-- NOT: SQL'de TRUNCATE TABLE komutu bulunmasına karşın SQLite bu komutu
-- desteklememektedir. Truncate komutu bir tabloyu değil içindeki tüm verileri
-- silmek için kullanılır. SQLite'da bu komut yoktur.

INSERT INTO vacation_plan VALUES(34, "TURKEY", "HOTEL İSTANBUL", 1,7, 4000);
-- Birden fazla kez çalıştırsak tekrar tekrar aynı satırları ekler. Bunun önüne geçmek için
-- PRIMARY KEY yapılması gerekir.
INSERT INTO vacation_plan VALUES(42, "TURKEY", "HOTEL MEVLANA", 2, 4, 2000);

/*vacation_plan tablosuna vacation_lenght ve budget sütunlarını eksik olarak veri girişi
yapınız*/

INSERT INTO vacation_plan (place_id, country, hotel_name, employee_id)
VALUES(06, "TURKEY", "HOTEL BAŞKENT", 3);

/* table_name'den sonra sütun isimlerini yazarak sonrasında sütun isimlerine uygun sırayla
VALUES içine değişken tiplerine uygun değerler yazılarak ekleme yapılır.*/

/*-----
CONSTRAINTS - KISITLAMALAR
-----

NOT NULL - Bir Sütunun NULL içermemesini garanti eder.

UNIQUE - Bir sütundaki tüm değerlerin BENZERSİZ olmasını garanti eder.

PRIMARY KEY - Bir sütunun NULL içermemesini ve sütundaki verilerin
BENZERSİZ olmasını garanti eder.(NOT NULL ve UNIQUE birleşimi gibi)

FOREIGN KEY - Başka bir tablodaki Primary Key'i referans göstermek için kullanılır.
Böylelikle, tablolar arasında ilişki kurulmuş olur.

DEFAULT - Herhangi bir değer atanmadığında Başlangıç değerinin atanmasını sağlar.*/

```



```

CREATE TABLE workers(
  id INT PRIMARY KEY,
  id_number TEXT NOT NULL UNIQUE,
  name TEXT DEFAULT 'NONAME',
  salary INTEGER NOT NULL
);

INSERT INTO workers VALUES(1,"12345678910","CAN CANAN", 10000);
-- UNIQUE CONST VIOLATION, unique olduğu için aynı id ile ekleme yapılamaz.
-- INSERT INTO workers VALUES(1,"44456789100","AHMET CANAN", 7000);
INSERT INTO workers VALUES(3,"55547896321", NULL, 9000);
-- name kısmına NONAME yazılır DEFAULT.
INSERT INTO workers (id,id_number, salary) VALUES(4,"55787876321", 9000);

/*vacation_plan tablosunu place_id sutunu PK ve employee_id sutununu
ise FK olarak değiştirerek vacation_plan2 adında yeni bir tablo oluşturunuz.
Bu tablo, employees tablosu ile ilişkili olmalıdır */

CREATE TABLE vacation_plan2(
  place_id INT,
  country NVARCHAR(50),
  hotel_name NVARCHAR(100),
  employee_id INTEGER,
  vacation_length INT,
  budget REAL,
  PRIMARY KEY (place_id),
  FOREIGN KEY (employee_id) REFERENCES employees(EmployeeId)
);

/* Employees tablosundaki EmployeeId'si 1 olan kişi için bir tatil planı giriniz.*/
INSERT INTO vacation_plan2 VALUES(35, "TURKEY","HAPPY NATION", 9, 5, 5000);
-- HATA VERİR

/*JOIN işlemi ile 2 tablodan veri çekme*/

SELECT e.FirstName, e.LastName, v.length, v.hotel_name,
FROM employees e,
INNER JOIN vacation_plan2 v,
on e.EmployeeId = v.employee_id;

/*-----
/* ALTER TABLE (ADD, RENAME TO, DROP)
/* SQLITE MODIFY VE DELETE KOMUTLARINI DOĞRUDAN DESTEKLENMEZ
/*-----

/*vacation_plan2 tablosuna name adında ve DEFAULT değeri noname olan
yeni bir sütun ekleyelim */

ALTER TABLE vacation_plan2
ADD name TEXT DEFAULT "ISIMSIZ",

/* workers tablosunun adını name sütununu siliniz*/
DROP COLUMN

/* workers tablosunun adını people olarak değiştiriniz*/

```

```

ALTER TABLE workers
RENAME to people;

/*-----
/*  UPDATE,DELETE
-- SYNTAX
-----
-- UPDATE tablo_adı
-- SET sutun1 = yeni_deger1, sutun2 = yeni_deger2,...
-- WHERE koşul;

--DELETE tablo_adı
--WHERE koşul;
/*-----*/

/*vacation_plan2 tablosundaki employee_id=1 olan kaydını hotel_name'ini Komagene Hotel olarak
güncelleyiniz.*/

UPDATE vacation_plan2
SET hotel_name = 'Komagene Hotel'
WHERE employee_id = 1;

/* people tablosunda salary sütünü 7000 'den az olanların salary(maaşına)
%10 zam yapacak sorguyu yazınız*/

UPDATE people
SET salary = salary*1.2
WHERE salary < 7000;

/*people tablosundaki tüm kayıtkarın salary sütununu 10000 olarak güncelleyiniz */
UPDATE people
SET salary=10000;

/*vacation_plan2 tablosundaki employee_id=1 olan kaydı siliniz*/
DELETE FROM vacation_plan2
WHERE employee_id=1;

```