**SQlite ile Veri Tabanı İşlemleri ;**

Birbiri ile ilişkili dosyaların oluşturduğu yapıya **veritabanı (database)** denir. Databaseler, birbirleri ile ilişkili, düzenli kayıtlar toplluğudur.

Bir database'in ana çalışma sayfası tablodur. Bir tablodaki sütunlar, alanlara, satırlar ise kayıtlara karşılık gelir. Çoğu uygulama için bilgisayara girelen veya elde edilen verileri, depolama birimlerinde saklamak ve onlar üzerinde işlem yapmak gerekir. Bu işlemler yapılırken klasik dosyalama sistemleri bazı dezavantajlara sahiptir ; Veri tekrarı, Verinin birkaç dosyada güncellemesi, Belleğin tekrarlı bilgi nedeniyle israfı, Sadece belli bir dilin kullanılması.

SQlite benzeri veri tabanı yönetim sistemlerinin kullanım sebepleri ise ;

* Veri tekrarını azaltmak,
* Hataları gidermek,
* Verileri daha kolay paylaşmak.

**SQL'e Giriş : Temel SQL Sorguları :**

İlişkisel veri tabanı sistemelrinin yönetiminde SQL (Structured Query Language - Yapısal Sorgulama Dili). SQL sorguları bir tablodaki veriler üzerinde ; yeniden kayıt oluşturma; crate, okuma; read, güncelleme; update, dileme; delete gibi işlemler gerçekleştirir. **Database programları farklı olsada SQL sorgulama dili evrenseldir ve sorgulamaları standarlaşmıştır.**

**CREATE TABLE :** (CREATE TABLE Tablo\_ismi (Alan1 INTEGER PRIMARY KEY, Alan2 TEXT, ... .)) Tablo **oluşturmak** için kullanılır. Alan isimlerinin veri tipleri sqlite3 veri tipi olmalıdır. İstenirse örnekteki gibi PRIMARY KEY seçilebilir. Alan1 ve Alan2 sütun isimleridir.

CREATE TABLE **IF NOT EXISTS** kitaplık (İsim TEXT, Yazar TEXT, Yayınevi TEXT , Sayfa\_Sayısı INT)

SELECT : (SELECT Alan1, Alan2, ... FROM Tablo\_isim) Database kayıtlarını **okumak** ve **listelemek** için **FROM** ile birlikte kullanılır. Koşullu sorgulama yapmak için **WHERE** deyimi kullanılır. SEELCET sorgularıda \* karakteri tüm alanları göster anlamındadır. (SELECT\*FROM Personel WHERE adSoyad="bahadır") , (SELECT no, adSoyad, bolum FROM Personel)

**INSERT** : (INSERT INTO Tablo\_ismi (Alan1, Alan2, ...) VALUES (Veri1, Veri2)) Tabloya veri **eklemek** için **INTO** ile birlikte kullanılır. Alanlar ile verilerin veri tiplerinin aynı olmasına dikkat edilmeli.

(INSERT INTO personel (no, adSoyad, bolum) VALUES (3, "FATMA","HUKUK"))

cursor.execute("INSERT INTO kitaplık VALUES(?,?,?,?)",(isim,yazar,yayınevi,sayfa\_sayısı))

**UPDATE** : Verileri **güncellemek** için **SET** ile birlikte kullanılır. (UPDATE personel SET adSoyad="Fazıl Ak" WHERE no=3)

**DELETE** : Verileri silmek için **FROM** ifadesi ile birlikte kullanılır. Koşula bağlı olarak silinecekse **WHERE** kullanılır. (DELETE FROM Personel WHERE bolum="Hukuk")

**DROP** : Databaseden **tabloyu siler**. Koşula bağlı olarak silinicekse **if** deyimi kullanılır. (DROP TABLE if Koşul Tablo\_ismi; )

**SQLite** **: Küçük ölçekli bir ilişkisel veri tabanı yönetim sistemidir**. SQLite ayrı bir database sunucusuna gerek duymaz, yazılan koda gömülür. Her bir database ayrı bir dosya olarak tutulur, küçük boyutlu dosyalar için idealdir. SQLite uygulamaları için Python'da ; **sqlite3** databese modülü kullanılır.

bgln=sqlite3.connect(vt) = Database dosyası (vt) ile **bağlantı kurar**. Artık vt, bgln ismi ile temsil edilir.

imlec=bgln.cursor() = Bir cursor(imlec) nesnesi oluşturulur.

imlec.exucate("SQL sorguları") = Parantez içindeki SQL **sorgusunu** çalıştırır.

imlec.exucatemany("INSERT INTO Tablo VALUES (?,?,?)", "kayitlar") = Parantez içerisindeki **birden fazla** kayıt üzerinde işlem yapıcak SQL sorgusunu çalıştırır.

bgln.commit() = Değişiklikleri veri tabanına işler**, veritabanını günceller.**

bgln.close() = Veritabanını **kapatır**.

sqlite3.version = Sqlite veritabanı sürümünü verir.

sqlite3.sqlite\_version = Sqlite3 kütüphane sürümünü verir.

import sqlite3 # Sqlite'yı dahil ediyoruz

con = sqlite3.connect("kütüphane.db") # Tabloya bağlanıyoruz.

cursor = con.cursor() # cursor isimli değişken veritabanı üzerinde işlem yapmak için kullanacağımız imleç olacak.

def tablo\_oluştur():

cursor.execute("CREATE TABLE IF NOT EXISTS kitaplık (İsim TEXT, Yazar TEXT, Yayınevi TEXT, Sayfa\_Sayısı INT)") # Sorguyu çalıştırıyoruz.

con.commit() # Sorgunun veritabanı üzerinde geçerli olması için commit işlemi gerekli.

tablo\_oluştur()

con.close() # Bağlantıyı koparıyoruz.

**sqlite3 Veri Tipleri ;**

* INTEGER = int
* REAL = float
* TEXT = str
* BLOB = bytes
* NULL = None

**Bazı SQL Sorguları :**

* SELECT \* FROM T : T tablosundaki kayıtları listeler
* SELECT no FROM T : T tablosunda no alanındaki kayıtları listeler
* SELECT DISTINCT ad FROM T WHERE ad LIKE “b%” : T tablosunda adı b ile başlayan ad alanındaki kayıtları listeler
* SELECT \* FROM T WHERE no=1 : T tablosunda no=1 olan tüm kayıtları listeler
* SELECT \* FROM T ORDER BY no DESC : T tablosundaki kayıtları no alanına göre azalan sırada listeler
* DELETE FROM T WHERE ad=”a” : T tablosunda adı a olan kaydı siler
* UPDATE T SET ad=”c” WHERE ad=”b” : T tablosunda adı b olan kaydı c olarak günceller
* INSERT INTO T (no,ad) VALUES (4,”f”) : T tablosuna yazılı kayıtları ekler
* DROP TABLE if exists T : Eğer T tablosu mevcut ise tabloyu siler
* ALTER TABLE T RENAME TO P : T tablosunun ismini P olarak değiştirir.