

Temel Veri Tabanı

Kasım ADALAN

Elektronik ve Haberleşme Mühendisi

Software Developer

İçindekiler

- Tablo yapısı
- Tablo tasarlama
- İlişkisel Tablo Modeli
- Birincil Anahtar - PRIMARY KEY
- Dış Anahtar - FOREIGN KEY
- SQL ifadeler
- DB Browser ile çalışma
- SQLite ile çalışma

Tablo Yapısı

- Tüm Veritabanları içerisinde bilgiler tablolarda
Alan

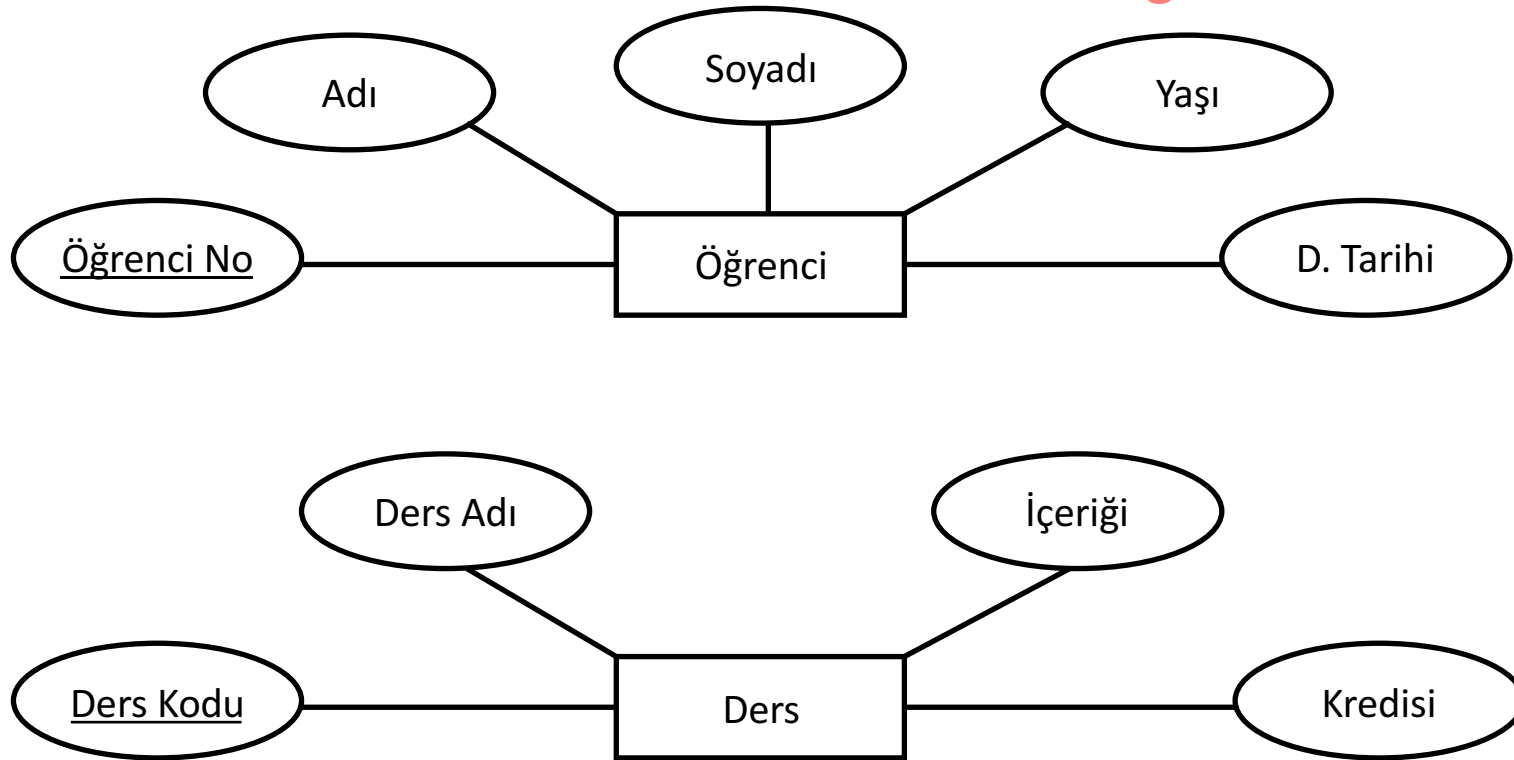
Veri

not_id	ogrenci_adi	ders_adi	not1	not2
1	Mehmet ERSOY	Tarih	50	60
2	Mehmet ERSOY	Fizik	70	80
3	Zeynep GÜR	Tarih	70	90
4	Cemal GELİR	Tarih	30	50

→ **Kayıt**

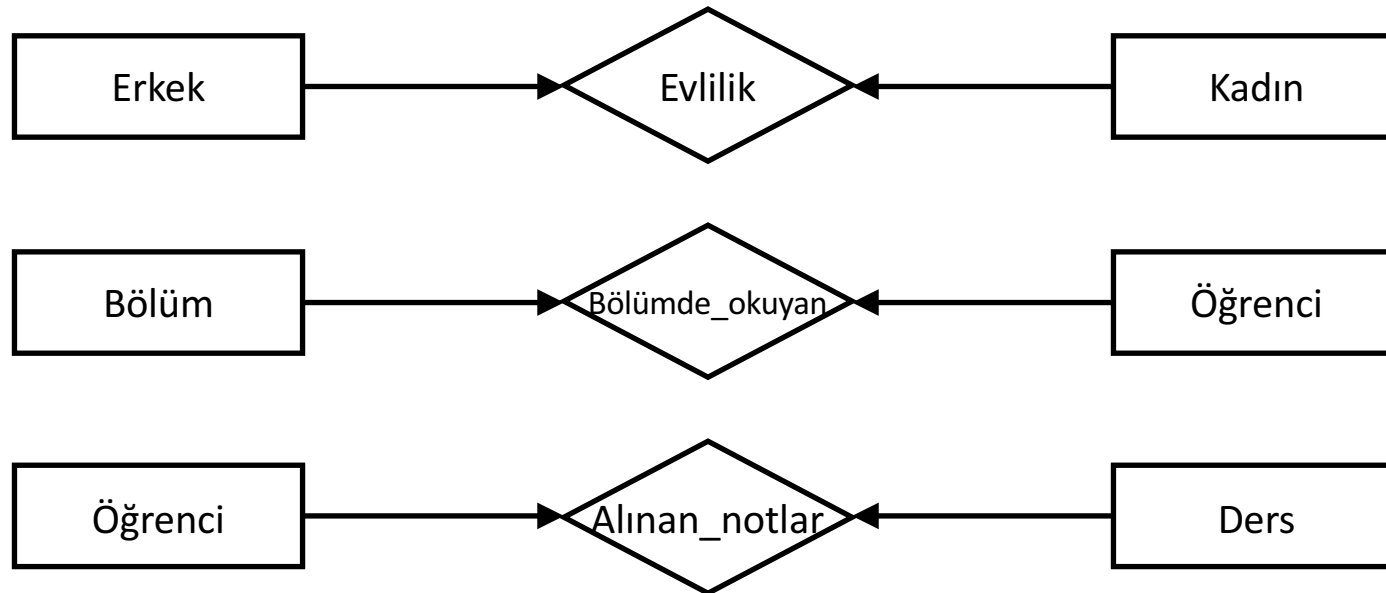
Tablo Tasarlama

- Öncelikle her tablo için uygun alanların belirlenmelidir aynı sınıflardaki özelliklerin belirlenmesi gibi.

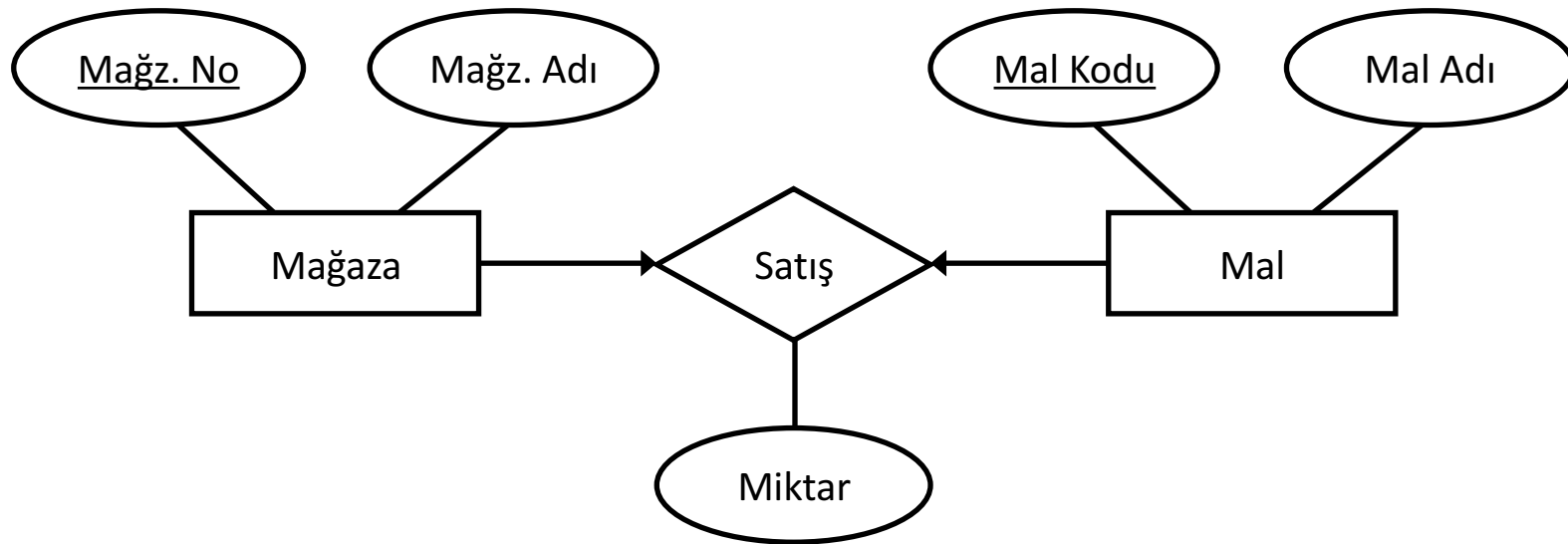


İlişkisel Tablo Modeli

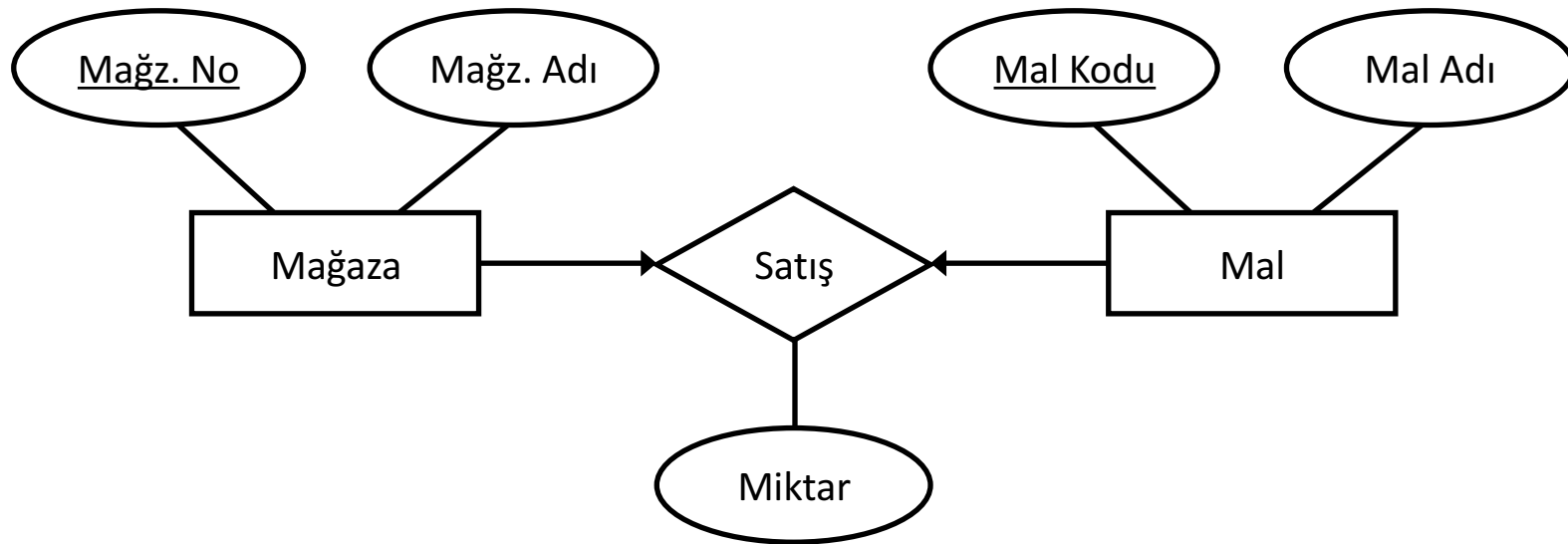
- Tablolardaki gereksiz veri tekrarlarını önlemek için kullanılan bir yapıdır.



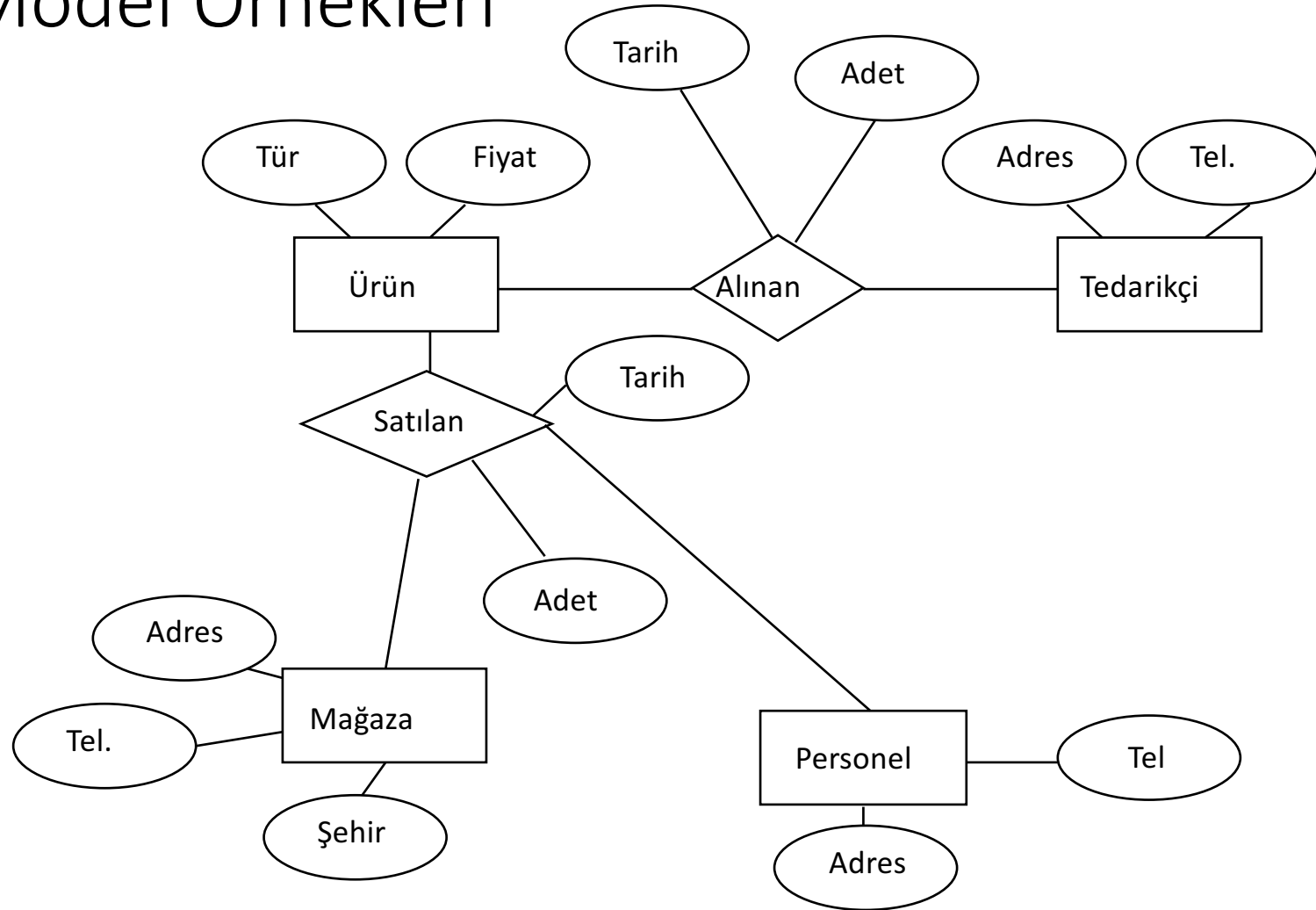
İlişkisel Model Örnekleri



İlişkisel Model Örnekleri



İlişkisel Model Örnekleri

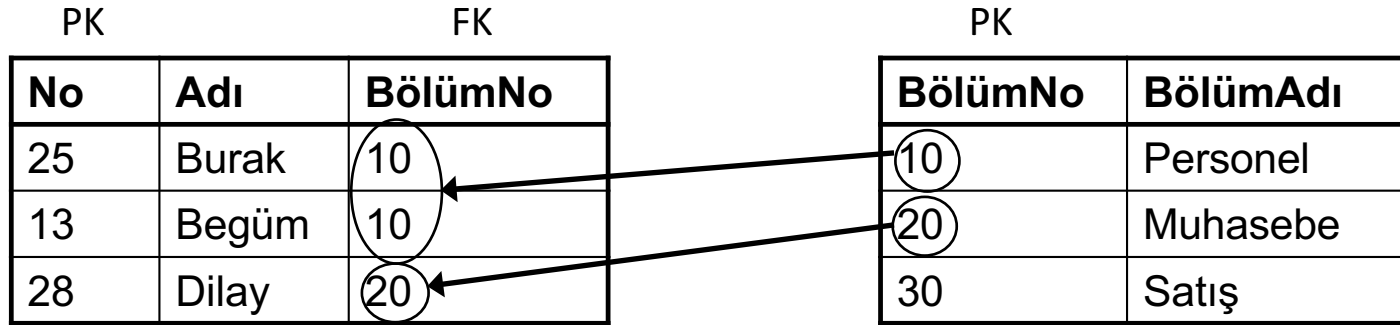


Birincil Anahtar – PRIMARY KEY

- İlişkisel veri tabanında bir tablonun benzer değerler içermeyen (tekil : unique) bir sütunu ya da birkaç sütunu birlikte **birincil anahtar** (primary key - PK) olarak tanımlanabilir.
- Birincil anahtar bir aday anahtardır ve söz konusu varlığın kayıtlarını en iyi biçimde karakterize eden anahtardır.
- Birincil anahtar tanımlandığında şu şekilde bir sınırlama konulmuş olacaktır:
 - Birincil anahtar NULL (boş) değerleri veya birbirinin aynı değerleri içeremez.

Dış Anahtar – FOREIGN KEY

- Bir dış anahtar (foreign key - FK) bir sütun veya çok sayıdaki sütunların birleşiminden oluşur.
- Dış anahtar aynı tablo ya da başka bir tablodaki bir birincil anahtar ile eşleştirilir.



Filmler Uygulaması

- Bu senaryo için temel bir veri tabanı modeli düşünelim.

film_id	film_adi	film_yil	film_resim	kategori_ad	yonetmen_ad
3	Inception	2010	inception	Bilim kurgu	C. Nolan
5	Django	2008	django	Dram	Q. Tarantino
9	The Hateful Eight	2010	thehatefuleight	Dram	Q. Tarantino

Filmler Uygulaması

- Bu senaryo için temel bir veri tabanı modeli düşünelim.

film_id	film_ad	film_yil	film_resim	kategori_id	yonetmen_id
3	Inception	2010	inception	11	88
5	Django	2008	django	22	90
9	The Hateful Eight	2010	thehatefuleight	22	90



kategori_id	kategori_adi
11	Bilim kurgu
22	Dram
44	Komedi

yonetmen_id	yonetmen_adi
88	C. Nolan
90	Q. Tarantino
130	Yılmaz Erdoğan

Okul Uygulaması

- Bu senaryo için temel bir veri tabanı modeli düşünelim.

not_id	ogrenci_adi	ders_adi	not1	not2
1	Mehmet ERSOY	Tarih	50	60
2	Mehmet ERSOY	Fizik	70	80
3	Zeynep GÜR	Tarih	70	90
4	Cemal GELİR	Tarih	30	50

Okul Uygulaması

- Bu senaryo için olması gereken veri tabanı modeli.

not_id	ogrenci_id	ders_id	not1	not2
1	1	1	50	60
2	1	2	70	80
3	2	1	70	90
4	4	1	30	50

ogrenci_id	ogrenci_adi
1	Mehmet ERSOY
2	Zeynep GÜR
4	Cemal GELİR

ders_id	ders_adi
1	Tarih
2	Fizik
3	Kimya

Sipariş Girme Uygulaması

- Kategorilere ayrılmış ürünlerden adet girerek sipariş oluşturma.


siparis_id	siparis_adet	urun_adi	kategori_adi
1	2	Pizza	Yiyecekler
2	1	Baklava	Tatlılar
3	1	Sütlaç	Tatlılar
4	3	Ayran	İçecekler

Sipariş Girme Uygulaması

- Bu senaryo için olması gereken veri tabanı modeli.

siparis_id	siparis_adet	urun_adi
1	2	1
2	1	2
3	1	3
4	3	5

kategori_id	kategori_adi
1	Yiyecekler
2	Tatlılar
3	İçecekler



urun_id	urun_adi	kategori_id
1	Pizza	1
2	Baklava	2
3	Sütlac	2
5	Ayran	3

Hastane Uygulaması

- Bu senaryo için temel bir veri tabanı modeli düşünelim.

randevu_id	bolum_adi	doktor_adi	hasta_adi	randevu_tarihi
1	Dermotoloji	Sedat AK	Mehmet ERSOY	12/02/2018 10:30
2	Kardiyoloji	Ceyda MERMER	Kemal ALTAY	10/04/2018 12:30
3	Dermotoloji	Sedat AK	Zeynep GÜR	07/01/2018 16:00
4	Dermotoloji	Sedat AK	Cemal GELİR	26/02/2018 9:30

Hastane Uygulaması

- Bu senaryo için olması gereken veri tabanı modeli.

randevu_id	doktor_id	hasta_adi	randevu_tarihi
1	1	Mehmet ERSOY	12/02/2018 10:30
2	2	Kemal ALTAY	10/04/2018 12:30
3	1	Zeynep GÜR	07/01/2018 16:00
4	1	Cemal GELİR	26/02/2018 9:30

bolum_id	bolum_adi
1	Dermatoloji
2	Kardiyoloji
3	Fizik Tedavi
4	Göz Hastalıkları

doktor_id	doktor_adi	bolum_id
1	Sedat AK	1
2	Ceyda MERMER	2
3	Ahmet Zıyan	1
4	Ece Kasırğa	3

SQL İFADELER

TABLO OLUŞTURMA

```
CREATE TABLE yonetmenler  
( yonetmen_id INTEGER PRIMARY KEY AUTOINCREMENT,  
  yonetmen_ad TEXT )
```

yonetmenler	Türü	KEY
yonetmen_id	int	Primary, Auto Increment
yonetmen_ad	String	

Filmler Uygulaması

- Bu senaryo için temel bir veri tabanı modeli düşünelim.

film_id	film_adi	film_yil	film_resim	kategori_id	yonetmen_id
3	Inception	2010	inception	11	88
5	Django	2008	django	22	90
9	The Hateful Eight	2010	thehatefuleight	22	90



kategori_id	kategori_adi
11	Bilim kurgu
22	Dram
44	Komedi

yonetmen_id	yonetmen_adi
88	C. Nolan
90	Q. Tarantino
130	Yılmaz Erdoğan

Örnek Veri Tabanı Modeli

kategoriler	Türü	KEY
kategori_id	int	Primary, Auto Increment
kategori_ad	String	

yonetmenler	Türü	KEY
yonetmen_id	int	Primary, Auto Increment
yonetmen_ad	String	

filmler	Türü	KEY
film_id	int	Primary, Auto Increment
film_ad	String	
film_yil	int	
film_resim	String	
kategori_id	int	FOREIGN
yonetmen_id	int	FOREIGN



Örnek Veri Tabanı SQLite yapısı.

```
CREATE TABLE yonetmenler  
( yonetmen_id INTEGER PRIMARY KEY AUTOINCREMENT,  
  yonetmen_ad TEXT )
```

yonetmenler	Türü	KEY
yonetmen_id	int	Primary, Auto Increment
yonetmen_ad	String	

Örnek Veri Tabanı SQLite yapısı.

```
CREATE TABLE kategoriler  
( kategori_id INTEGER PRIMARY KEY AUTOINCREMENT,  
  kategori_ad TEXT )
```

kategoriler	Türü	KEY
kategori_id	int	Primary, Auto Increment
kategori_ad	String	

Örnek Veri Tabanı SQLite yapısı.

filmler	Türü	KEY
film_id	int	Primary, Auto Increment
film_ad	String	
film_yil	int	
film_resim	String	
kategori_id	int	FOREIGN
yonetmen_id	int	FOREIGN

CREATE TABLE filmler

(film_id INTEGER PRIMARY KEY AUTOINCREMENT,

film_ad TEXT,

film_yil INTEGER,

film_resim TEXT,

kategori_id INTEGER,

yonetmen_id INTEGER,

FOREIGN KEY (kategori_id) REFERENCES kategoriler (kategori_id),

FOREIGN KEY (yonetmen_id) REFERENCES yonetmenler (yonetmen_id))

Veri Kaydı - Insert

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

SQLite sorgu karşılığı ;

```
INSERT INTO kisiler (kisi_ad,kisi_tel,kisi_yas,kisi_boy) VALUES ('mehmet','05362912932',23,1.69) ;
```

Update - Güncelleme

SQLite sorgu karşılığı ;

UPDATE kisiler **SET** kisi_ad = 'zeynep' **WHERE** kisi_no = 3 ;

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

Delete – Veri Silme

SQLite sorgu karşılığı ;

DELETE FROM kisiler **WHERE** kisi_no = 4 ;

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

SELECT – Seçim Yapma

- Tüm verileri getir.

`SELECT * FROM` kisiler

- İstenilen alanları getir.

`SELECT` kisi_ad,kisi_yas `FROM` kisiler

- Kayıt kontrol.

`SELECT` count(*) as toplam `FROM` kisiler

- Foreign key li tablolardan verileri alma.

`SELECT *FROM` kisiler,adres `WHERE` kisiler.adres_id = adres.adres_id

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

WHERE – Şart Oluşturma

Not : Şart yazarken yazı mı ? Sayısal veri mi? Buna dikkat edilmelidir.
Sayısal veriler normal yazılırken yazısal ifadeler " veya ' ile yazılır.

- Kisiler tablosundaki ahmet isimli kişi ve kişileri getir.

```
SELECT * FROM kisiler WHERE kisi_ad = "ahmet"
```

- Kisiler tablosundan yaşı 18 olanları getir.

```
SELECT * FROM kisiler WHERE kisi_yas = 18
```

- Kisiler tablosundan yaşı 18 den büyük olanları getir.

```
SELECT * FROM kisiler WHERE kisi_yas > 18
```

- Kisiler tablosundan yaşı 18 den büyük ve boyu 2.00 dan büyü

```
SELECT * FROM kisiler WHERE kisi_yas > 18 and kisi_boy > 2.00
```

- Kisiler tablosundan yaşı 18 den büyük veya boyu 2.00 dan büyük olanları getir.

```
SELECT * FROM kisiler WHERE kisi_yas > 18 or kisi_boy > 2.00
```

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

ORDER BY - SIRALAMA

ASC : Artan
DESC : Azalan

- Kisiler tablosundaki kişileri harf sırasına göre artan şekilde getir.

```
SELECT * FROM kisiler ORDER BY kisi_ad ASC
```

- Kisiler tablosundan yaşı 18 olanları getir.

```
SELECT * FROM kisiler ORDER BY kisi_ad ASC , kisi_yas DESC
```

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

LIKE – BENZERLİK ARAMA

- Kisiler tablosundaki kişilerdeki kisi adı içerisinde as olanları getir

```
SELECT * FROM kisiler WHERE kisi_ad like '%as%'
```

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

LIMIT– SINIRLI SAYIDA VERİ GETİR

- Kisiler tablosundan 5 kayıt getir.

```
SELECT * FROM kisiler LIMIT 5
```

- Kisiler tablosundan yaşı 18 den büyük olan 5 kayıt getir.

```
SELECT * FROM kisiler WHERE kisi_yas > 18 LIMIT 5
```

- Kisiler tablosundan RASGELE 5 kayıt getir.

```
SELECT * FROM kisiler ORDER BY RANDOM() LIMIT 5
```

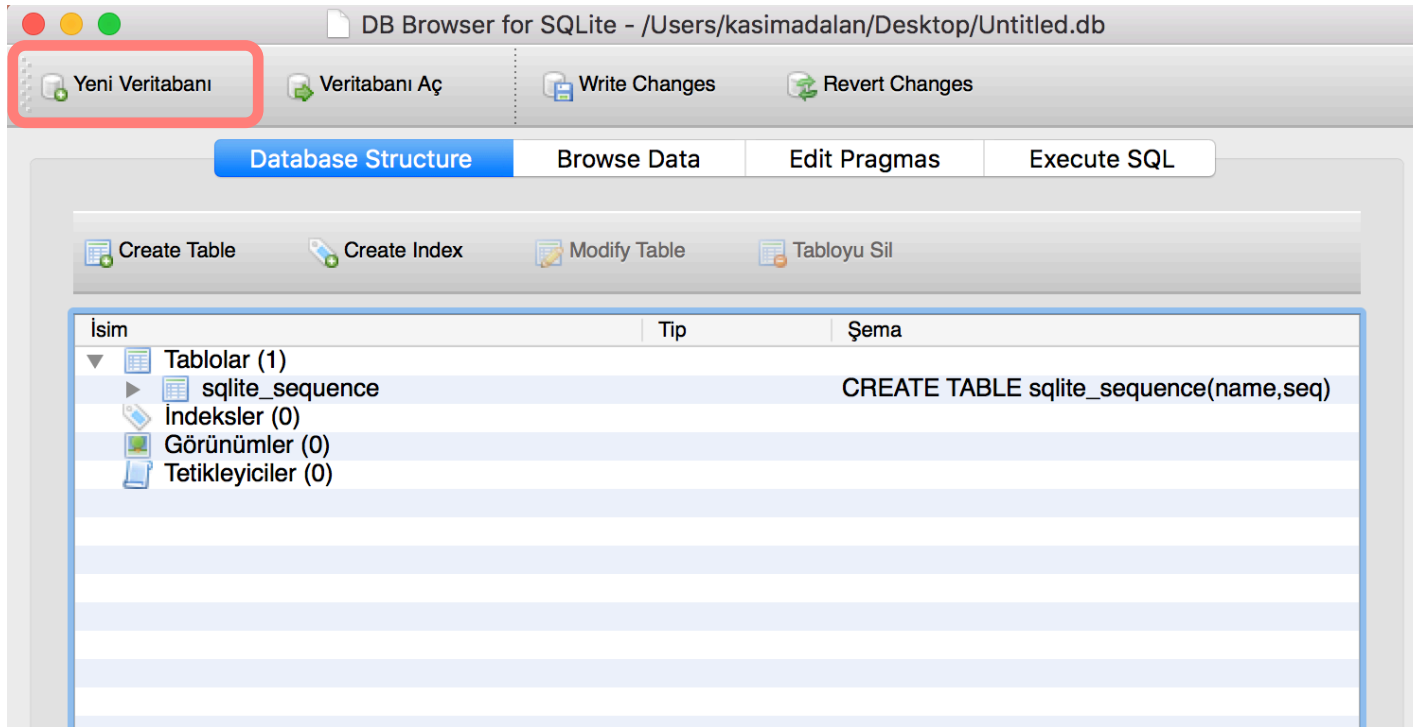
kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

DB Browser ile Çalışma

- <http://sqlitebrowser.org/>
- SQLite için geliştirilmiş veri tabanı yönetim uygulamasıdır.



Veri tabanı oluřturma



Tablo Oluřturma

Tablo tanımlarını düzenle

Tablo

tablo_adi

▼ Geliřmi

Alanlar

Alan Ekle Alanı kaldır Alanı yukarı taşı Alanı aşağı taşı

İsim	Tip	NULL Olamaz	Birincil Anahtar	Otomatik Arttırma	Benzersiz	Varsayılan	Kontrol	Yabancı Anahtar
id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			

```
1 CREATE TABLE `tablo_adi` (  
2   `id` INTEGER PRIMARY KEY AUTOINCREMENT  
3 );
```

Cancel OK

Foreign Key ile Bağlantı

Tablo

kisiler

▼ Gelişmi

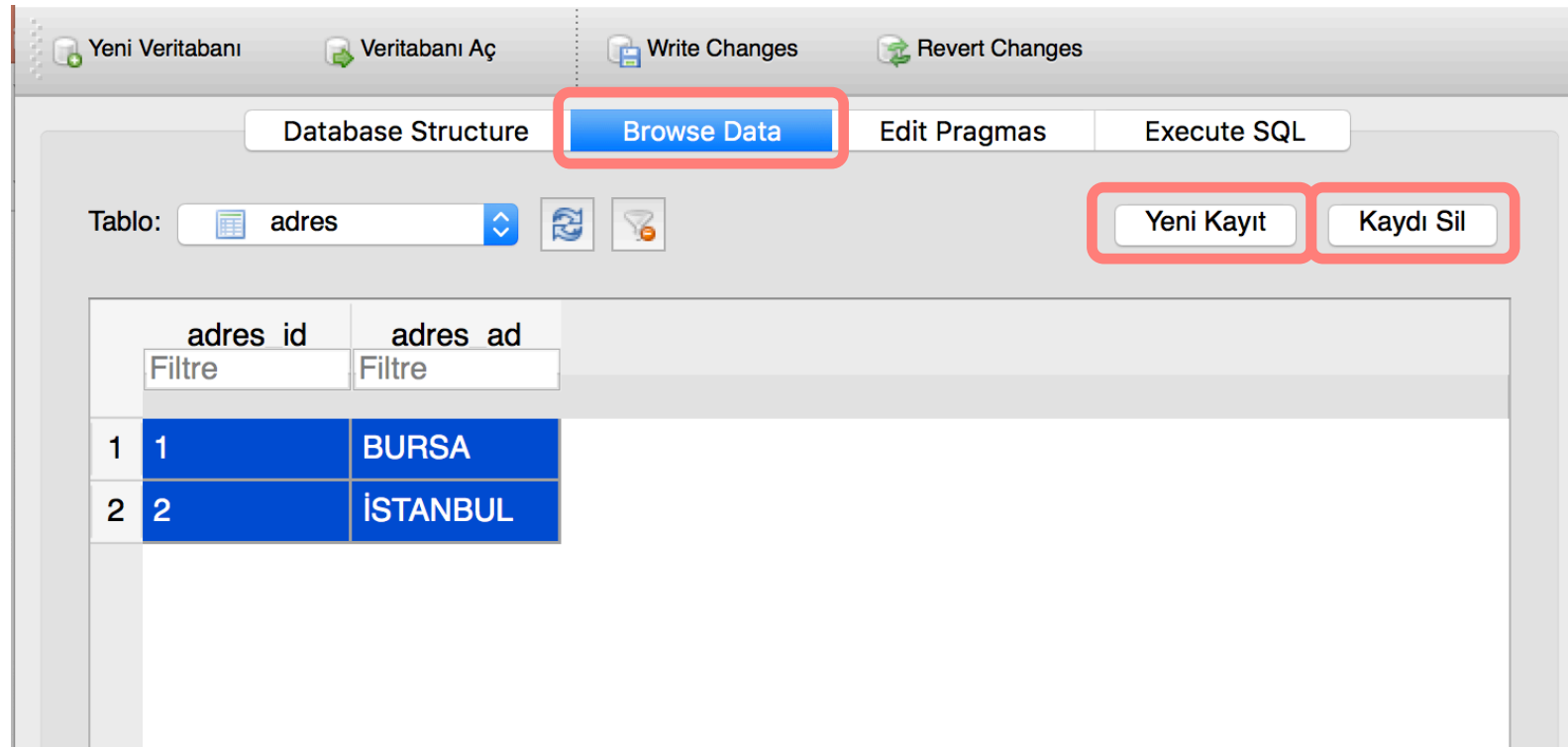
Alanlar

Alan Ekle Alanı kaldır Alanı yukarı taşı Alanı aşağı taşı

İsim	Tip	NULL Olamaz	Birincil Anahtar	Otomatik Arttırma	Benzersiz	Varsayıla	Kontrol	Yabancı Anahtar
kisi_id	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
kisi_ad	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
adres_id	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			<code>`adres`(`adres_id`)</code>

```
1 CREATE TABLE `kisiler` (  
2   `kisi_id` INTEGER PRIMARY KEY AUTOINCREMENT,  
3   `kisi_ad` TEXT,  
4   `adres_id` INTEGER,  
5   FOREIGN KEY(`adres_id`) REFERENCES `adres`(`adres_id`)  
6 );
```

Tablo kayıtları işlemleri

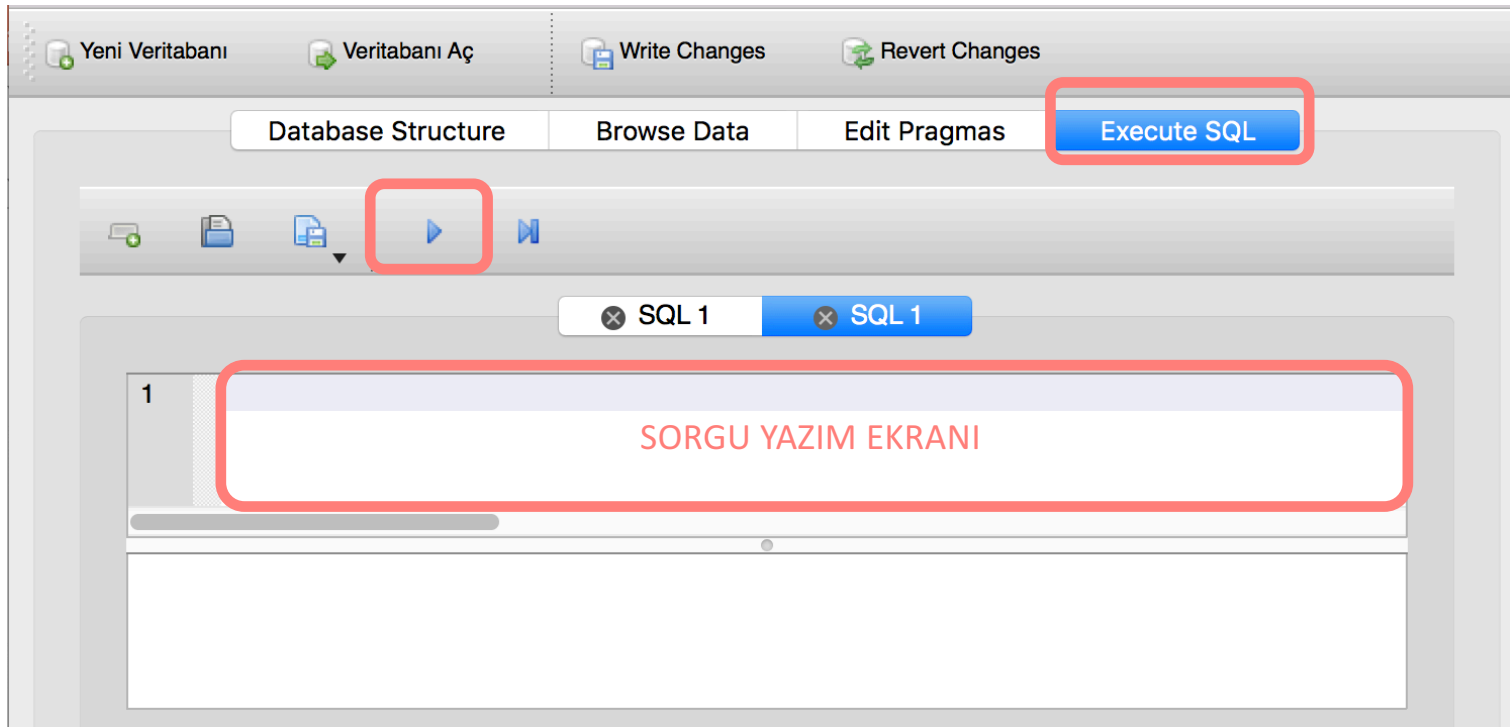


The screenshot displays a database management interface with the following elements:

- Top toolbar: **Yeni Veritabanı**, **Veritabanı Aç**, **Write Changes**, and **Revert Changes**.
- Navigation tabs: **Database Structure**, **Browse Data** (highlighted with a red box), **Edit Pragmas**, and **Execute SQL**.
- Table selection: **Tablo:** **adres** (with a dropdown arrow).
- Actions: **Yeni Kayıt** and **Kayı Sil** (both highlighted with red boxes).
- Data table:

	adres id	adres ad
	Filtre	Filtre
1	1	BURSA
2	2	İSTANBUL

Sorgu alıřtırma



SQLite Veri Tabanı ile Çalışma

Örnek Veri Tabanı Modeli

kategoriler	Türü	KEY
kategori_id	int	Primary, Auto Increment
kategori_ad	String	

yonetmenler	Türü	KEY
yonetmen_id	int	Primary, Auto Increment
yonetmen_ad	String	

filmler	Türü	KEY
film_id	int	Primary, Auto Increment
film_ad	String	
film_yil	int	
film_resim	String	
kategori_id	int	FOREIGN
yonetmen_id	int	FOREIGN



Örnek Veri Tabanı SQLite yapısı.

```
CREATE TABLE yonetmenler  
( yonetmen_id INTEGER PRIMARY KEY AUTOINCREMENT,  
  yonetmen_ad TEXT )
```

yonetmenler	Türü	KEY
yonetmen_id	int	Primary, Auto Increment
yonetmen_ad	String	

Örnek Veri Tabanı SQLite yapısı.

CREATE TABLE kategoriler

(kategori_id INTEGER PRIMARY KEY AUTOINCREMENT,
kategori_ad TEXT)

kategoriler	Türü	KEY
kategori_id	int	Primary, Auto Increment
kategori_ad	String	

Örnek Veri Tabanı SQLite yapısı.

```
CREATE TABLE filmler
( film_id INTEGER PRIMARY KEY AUTOINCREMENT,
  film_ad TEXT,
  film_yil INTEGER,
  film_resim TEXT,
  kategori_id INTEGER,
  yönetmen_id INTEGER,
  FOREIGN KEY (kategori_id) REFERENCES kategoriler (kategori_id),
  FOREIGN KEY (yönetmen_id) REFERENCES yönetmenler (yönetmen_id) )
```

filmler	Türü	KEY
film_id	int	Primary, Auto Increment
film_ad	String	
film_yil	int	
film_resim	String	
kategori_id	int	FOREIGN
yönetmen_id	int	FOREIGN

Veritabanı oluşturmak için sınıf oluşturulmalıdır.

```

public class VeritabaniYardimcisi extends SQLiteOpenHelper{

    private static final int Surum=1;

    private static String veritabaniAdi = "rehber";

    public VeritabaniYardimcisi(Context context) {

        super(context, veritabaniAdi, null, Surum);

    }

    @Override
    public void onCreate(SQLiteDatabase db) {

        db.execSQL("CREATE TABLE kisiler (kisi_no INTEGER PRIMARY KEY AUTOINCREMENT,kisi_ad TEXT,kisi_tel TEXT,kisi_yas INTEGER,kisi_boy DOUBLE);");

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int arg1, int arg2) {

        db.execSQL("DROP TABLE IF EXIST kisiler");

        onCreate(db);

    }

}

```

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

Veritabanı oluşturmak için sınıf oluşturulmalıdır.
Foreign Key'li Veritabanı Sınıfı

```

public class VeritabaniYardimcisi extends SQLiteOpenHelper{

    private static final int Surum=1;

    private static String veritabaniAdi = "filmler";

    public VeritabaniYardimcisi(Context context) {

        super(context, veritabaniAdi, null, Surum);

    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE kategoriler ( kategori_id INTEGER PRIMARY KEY AUTOINCREMENT, kategori_ad TEXT );");
        db.execSQL("CREATE TABLE yonetmenler ( yonetmen_id INTEGER PRIMARY KEY AUTOINCREMENT, yonetmen_ad TEXT );");
        db.execSQL("CREATE TABLE filmler ( " +
            "film_id INTEGER PRIMARY KEY AUTOINCREMENT, " +
            "film_ad TEXT, " +
            "film_yil INTEGER, " +
            "film_resim TEXT, " +
            "kategori_id INTEGER, " +
            "yonetmen_id INTEGER, " +
            "FOREIGN KEY (kategori_id) REFERENCES kategoriler (kategori_id), " +
            "FOREIGN KEY (yonetmen_id) REFERENCES yonetmenler (yonetmen_id) );");

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int arg1, int arg2) {

        db.execSQL("DROP TABLE IF EXIST kategoriler");
        db.execSQL("DROP TABLE IF EXIST yonetmenler");
        db.execSQL("DROP TABLE IF EXIST filmler");

        onCreate(db);

    }

}

```


Veri tabanının tablolarının **Sınıflar** ile
Modellenmesi

```

public class Yonetmenler implements Serializable {
    private int yonetmen_id;
    private String yonetmen_ad;

    public Yonetmenler() {
    }

    public Yonetmenler(int yonetmen_id, String yonetmen_ad) {
        this.yonetmen_id = yonetmen_id;
        this.yonetmen_ad = yonetmen_ad;
    }

    public int getYonetmen_id() {
        return yonetmen_id;
    }

    public void setYonetmen_id(int yonetmen_id) {
        this.yonetmen_id = yonetmen_id;
    }

    public String getYonetmen_ad() {
        return yonetmen_ad;
    }

    public void setYonetmen_ad(String yonetmen_ad) {
        this.yonetmen_ad = yonetmen_ad;
    }
}

```

yonetmenler	Türü	KEY
yonetmen_id	int	Primary, Auto Increment
yonetmen_ad	String	

```

public class Kategoriler implements Serializable{
    private int kategori_id;
    private String kategori_ad;

    public Kategoriler() {
    }

    public Kategoriler(int kategori_id, String kategori_ad) {
        this.kategori_id = kategori_id;
        this.kategori_ad = kategori_ad;
    }

    public int getKategori_id() {
        return kategori_id;
    }

    public void setKategori_id(int kategori_id) {
        this.kategori_id = kategori_id;
    }

    public String getKategori_ad() {
        return kategori_ad;
    }

    public void setKategori_ad(String kategori_ad) {
        this.kategori_ad = kategori_ad;
    }
}

```

kategoriler	Türü	KEY
kategori_id	int	Primary, Auto Increment
kategori_ad	String	

```

public class Filmler implements Serializable {
    private int film_id;
    private String film_ad;
    private int film_yil;
    private String film_resim;
    private Kategoriler kategori;
    private Yonetmenler yonetmen;

    public Filmler() {
    }

    public Filmler(int film_id, String film_ad, int film_yil, String film_resim, Kategoriler kategori, Yonetmenler yonetmen) {
        this.film_id = film_id;
        this.film_ad = film_ad;
        this.film_yil = film_yil;
        this.film_resim = film_resim;
        this.kategori = kategori;
        this.yonetmen = yonetmen;
    }

    /*
    *
    * SETTER VE GETTER METODLAR BURADA OLUCAK
    *
    * */
}

```

filmler	Türü	KEY
film_id	int	Primary, Auto Increment
film_ad	String	
film_yil	int	
film_resim	String	
kategori_id	int	FOREIGN
yonetmen_id	int	FOREIGN

Veri tabanı tabloları üzerinde işlemler

SQLite Execute işlemleri

- Insert – Veri Kaydı
- Update - Veri Güncelleme
- Delete – Veri Silme

Insert – Veri Kaydı

SQLite sorgu karşılığı ;

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

INSERT INTO kisiler (kisi_ad,kisi_tel,kisi_yas,kisi_boy) **VALUES** ('mehmet','05362912932',23,1.69) ;

```
public void kisiEkle(VeritabaniYardimcisi vt,String kisi_ad,String kisi_tel,int kisi_yas,Double kisi_boy){  
    SQLiteDatabase dbx=vt.getWritableDatabase();  
    ContentValues values=new ContentValues();  
    values.put("kisi_ad",kisi_ad );  
    values.put("kisi_tel",kisi_tel );  
    values.put("kisi_yas",kisi_yas );  
    values.put("kisi_boy",kisi_boy );  
    dbx.insertOrThrow("kisiler", null, values);  
    dbx.close();  
}
```

Update - Güncelleme

SQLite sorgu karşılığı ;

UPDATE kisiler **SET** kisi_ad = 'zeynep' **WHERE** kisi_no = 3 ;

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

```
public void kisiGuncelle(VeritabaniYardimcisi vt,int kisi_no,String kisi_ad,String kisi_tel,int kisi_yas,Double kisi_boy){  
    SQLiteDatabase dbx=vt.getWritableDatabase();  
    ContentValues values=new ContentValues();  
    values.put("kisi_ad",kisi_ad );  
    values.put("kisi_tel",kisi_tel );  
    values.put("kisi_yas",kisi_yas );  
    values.put("kisi_boy",kisi_boy );  
    dbx.update("kisiler", values,"kisi_no=?",new String[] {String.valueOf(kisi_no)});  
    dbx.close();  
}
```


Delete – Veri Silme

SQLite sorgu karşılığı ;

DELETE FROM kisiler **WHERE** kisi_no = 4 ;

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

```
public void kisiSil(VeritabaniYardimcisi vt,int kisi_no){  
    SQLiteDatabase dbx=vt.getWritableDatabase();  
    dbx.delete("kisiler", "kisi_no=?",new String[] {String.valueOf(kisi_no)} );  
    dbx.close();  
}
```

SQLite veri tabanından veri çekme işlemleri

- Select sorgu işlemleri olarak bilinmektedir.
- Kisiler tablosundaki tüm kisileri getirir.
Örn : `SELECT * FROM kisiler`
- Kisiler tablosundaki ahmet isimli kişi ve kişileri getir.
Örn : `SELECT * FROM kisiler WHERE kisi_ad = 'ahmet'`

Tüm Verilerin alınması

kategoriler	Türü	KEY
kategori_id	int	Primary, Auto Increment
kategori_ad	String	

```
public ArrayList<Kategoriler> tumKategoriler(VeritabaniYardimcisi vt){  
  
    ArrayList<Kategoriler> kategorilerArrayList=new ArrayList<>();  
    SQLiteDatabase dbx=vt.getWritableDatabase();  
    Cursor cursor=dbx.rawQuery("SELECT * FROM kategoriler ", null);  
  
    while(cursor.moveToNext()){  
        Kategoriler kategori = new Kategoriler(  
            cursor.getInt(cursor.getColumnIndex("kategori_id")),  
            cursor.getString(cursor.getColumnIndex("kategori_ad")));  
  
        kategorilerArrayList.add(kategori);  
    }  
    return kategorilerArrayList;  
}
```

Tüm Verilerin Alınması (Foreign Key)

```
public ArrayList<Filmler> tumFilmlerKategoriIdGore(VeritabaniYardimcisi vt, int kategori_id){  
  
    ArrayList<Filmler> filmlerArrayList=new ArrayList<>();  
    SQLiteDatabase dbx=vt.getWritableDatabase();  
    Cursor cursor=dbx.rawQuery("SELECT * FROM filmler,kategoriler,yonetmenler " +  
        "WHERE filmler.kategori_id = kategoriler.kategori_id and filmler.yonetmen_id = yonetmenler.yonetmen_id and filmler.kategori_id = "+kategori_id, null);  
  
    while(cursor.moveToNext()){  
        Kategoriler kategori = new Kategoriler(  
            cursor.getInt(cursor.getColumnIndex("kategori_id")),  
            cursor.getString(cursor.getColumnIndex("kategori_ad")));  
  
        Yonetmenler yonetmen = new Yonetmenler(  
            cursor.getInt(cursor.getColumnIndex("yonetmen_id")),  
            cursor.getString(cursor.getColumnIndex("yonetmen_ad")));  
  
        Filmler film = new Filmler(  
            cursor.getInt(cursor.getColumnIndex("film_id")),  
            cursor.getString(cursor.getColumnIndex("film_ad")),  
            cursor.getInt(cursor.getColumnIndex("film_yil")),  
            cursor.getString(cursor.getColumnIndex("film_resim")),  
            kategori,  
            yonetmen);  
  
        filmlerArrayList.add(film);  
    }  
    return filmlerArrayList;  
}
```

filmler	Türü	KEY
film_id	int	Primary, Auto Increment
film_ad	String	
film_yil	int	
film_resim	String	
kategori_id	int	FOREIGN
yonetmen_id	int	FOREIGN

Veri tabanında kayıt kontrol

```
public int kayitKontrol(VeritabaniYardimcisi vt,String kisi_ad){  
    int sonuc=0;  
  
    SQLiteDatabase dbx=vt.getWritableDatabase();  
  
    Cursor cr=dbx.rawQuery("select count(*) as sonuc from kisiler where kisi_ad='"+kisi_ad+"' ", null);  
  
    if(cr.moveToFirst()){  
        do {  
            sonuc=cr.getInt(cr.getColumnIndex("sonuc"));  
        } while (cr.moveToNext());  
    }  
    return sonuc;  
}
```

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

Kelime içerisinde arama

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

```
public ArrayList<Kisiler> ara(VeritabaniYardimcisi vt,String keyWord){  
  
    ArrayList<Kisiler> kisilerArrayList=new ArrayList<>();  
    SQLiteDatabase dbx=vt.getWritableDatabase();  
    Cursor cursor=dbx.rawQuery("SELECT * FROM kisiler WHERE kisi_ad like '%" +keyWord+"%' ", null);  
  
    while(cursor.moveToNext()){  
        Kisiler kisi = new Kisiler(cursor.getInt(cursor.getColumnIndex("kisi_no")),  
                                    cursor.getString(cursor.getColumnIndex("kisi_ad")),  
                                    cursor.getString(cursor.getColumnIndex("kisi_tel")),  
                                    cursor.getInt(cursor.getColumnIndex("kisi_yas")),  
                                    cursor.getDouble(cursor.getColumnIndex("kisi_boy")));  
        kisilerArrayList.add(kisi);  
    }  
    return kisilerArrayList;  
}
```

Kelime içerisinde arama

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

```
public ArrayList<Kisiler> ara(VeritabaniYardimcisi vt,String keyWord){  
    ArrayList<Kisiler> kisilerArrayList=new ArrayList<>();  
    SQLiteDatabase dbx=vt.getWritableDatabase();  
    //Cursor cursor=dbx.rawQuery("SELECT * FROM kisiler WHERE kisi_ad like '%" +keyWord+"%' ", null);  
    Cursor cursor = dbx.query( table: "kisiler",  
        columns: null,  
        selection: "kisi_ad like ?",  
        new String []{"%" + keyWord+ "%"},  
        groupBy: null,  
        having: null,  
        orderBy: null);  
    while(cursor.moveToNext()){  
        Kisiler kisi = new Kisiler(cursor.getInt(cursor.getColumnIndex( columnName: "kisi_no")),  
            cursor.getString(cursor.getColumnIndex( columnName: "kisi_ad")),  
            cursor.getString(cursor.getColumnIndex( columnName: "kisi_tel")),  
            cursor.getInt(cursor.getColumnIndex( columnName: "kisi_yas")),  
            cursor.getDouble(cursor.getColumnIndex( columnName: "kisi_boy")));  
        kisilerArrayList.add(kisi);  
    }  
    return kisilerArrayList;  
}
```

Rastgele ve Sınırlı veri getirme

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

```
public ArrayList<Kisiler> rasgele5KisiGetir(VeritabaniYardimcisi vt){  
  
    ArrayList<Kisiler> kisilerArrayList=new ArrayList<>();  
    SQLiteDatabase dbx=vt.getWritableDatabase();  
    Cursor cursor=dbx.rawQuery("SELECT * FROM kisiler ORDER BY RANDOM() LIMIT 5", null);  
  
    while(cursor.moveToNext()){  
        Kisiler kisi = new Kisiler(  
            cursor.getInt(cursor.getColumnIndex("kisi_no")),  
            cursor.getString(cursor.getColumnIndex("kisi_ad")),  
            cursor.getString(cursor.getColumnIndex("kisi_tel")),  
            cursor.getInt(cursor.getColumnIndex("kisi_yas")),  
            cursor.getDouble(cursor.getColumnIndex("kisi_boy")));  
        kisilerArrayList.add(kisi);  
    }  
    return kisilerArrayList;  
}
```


Bir tane kayıt getirme

kisiler	Türü
kisi_no	int
kisi_ad	String
kisi_tel	String
kisi_yas	int
kisi_boy	double

```
public Kisiler kisiyiGetir(VeritabaniYardimcisi vt,int kisi_no){  
    Kisiler kisi=new Kisiler();  
    SQLiteDatabase dbx=vt.getWritableDatabase();  
    Cursor cursor=dbx.rawQuery("SELECT * FROM kisiler WHERE kisi_no="+kisi_no, null);  
  
    while(cursor.moveToNext()){  
        Kisiler k = new Kisiler(  
            cursor.getInt(cursor.getColumnIndex("kisi_no")),  
            cursor.getString(cursor.getColumnIndex("kisi_ad")),  
            cursor.getString(cursor.getColumnIndex("kisi_tel")),  
            cursor.getInt(cursor.getColumnIndex("kisi_yas")),  
            cursor.getDouble(cursor.getColumnIndex("kisi_boy")));  
        kisi = k ;  
    }  
    return kisi;  
}
```