

## DERS İÇERİĞİ:

- SQLite ve ORMLite

### SQLite

Android SDK'da uygulamalarınızda kullanabilmeniz için **SQLite** veritabanı kütüphanesini sunar. SQLite kullanarak kendi uygulamanıza özel relational (ilişkisel) veritabanı tasarımı yapabilirsiniz. Veritabanınız uygulamanıza özel olduğu için başka herhangi bir uygulama kayıtlı bilgilerinize erişim sağlayamayacaktır. Diğer veritabanı sistemlerinden farkı serverless(sunucusuz) olmasıdır. Uygulamanız içerisinde, farklı bir işlem gerektirmeksizin çalışan, gömülü bir veritabanı yönetim sistemidir.

### ORMLite

İlişkisel Nesne Haritalama (**O**bject **R**elational **M**apping) karmaşık veri tabanı işlemleri için basit hafif işlevsellik sağlamaktadır. Android'de sıkıcı ve uzun SQL ifadeleri yazmak çoğu zaman geliştiriciyi projeden soğutmaktadır. **ORMLite** bu durum için geliştirilmiş ORM kütüphanelerinden biridir. ORM kütüphaneleri genel olarak veri tabanı işlemlerini çok basit ifadeler kullanarak yapmanıza yardımcı kütüphanelerdir. Veri tabanından kullanılacak nesneler için bir sınıf oluşturularak bu sınıfla veri tabanı işlemleri arasında ilişki kurulur.

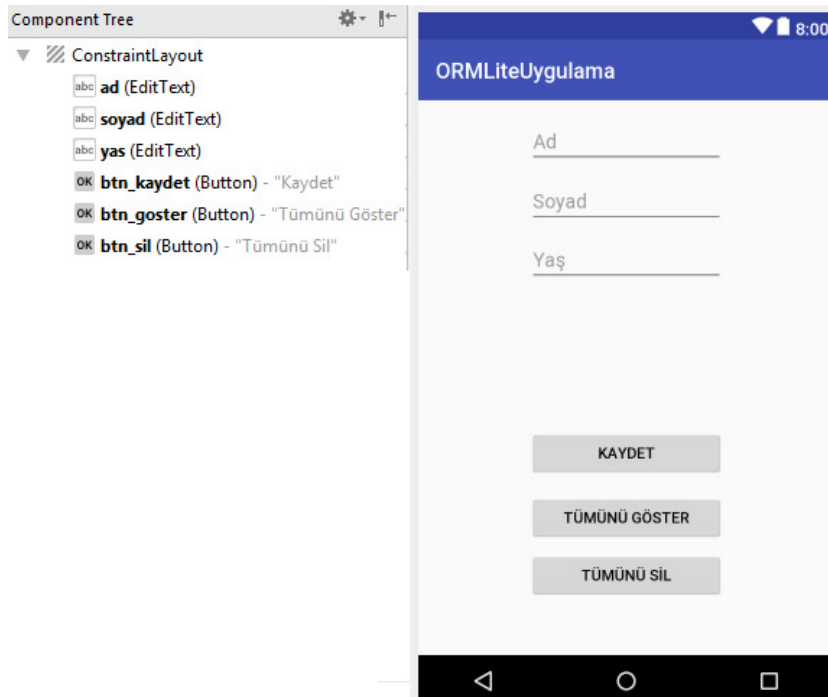
**OrmLiteSqliteOpenHelper**: Veri tabanı oluşturma, tabloları belirleme, versiyon belirleme ve veritabanı versiyon değişikliği (upgrade) işlemleri için bu sınıf kullanılmaktadır.

**OpenHelperManager**: Bağlantı paylaşımını optimize etmek, veritabanı bağlantılarını düzenlemek ve bu bağlantılara ulaşma işlemleri için bu sınıf kullanılmaktadır.

## Uygulama

**Amaç: (Bu kısmı uygulamanın sonunda siz dolduracaksınız!!!)**

1. Start New Android Project, Appliation Name: **ORMLiteUygulama**, Min. SDK: API20, Empty Activity, Finish.
2. Aşağıdaki gibi tasarımı oluşturun.



3. Size verilen ormlite jar dosyalarını libs kalsörüne kopyalayın.
4. **Ogrenci** adında bir sınıf oluşturun ve aşağıdaki kodları bu sınıfa ekleyin.

```
import com.j256.ormlite.field.DatabaseField;

public class Ogrenci {
    @DatabaseField (generatedId = true)
    private int id;
    @DatabaseField
    private String ad;
    @DatabaseField
    private String soyad;
    @DatabaseField
    private int yas;

    public Ogrenci() {

    }

    public Ogrenci(String ad, String soyad, int yas) {
        this.ad = ad;
        this.soyad = soyad;
        this.yas = yas;
    }

    public Ogrenci(int id, String ad, String soyad, int yas) {
        this.id = id;
        this.ad = ad;
        this.soyad = soyad;
        this.yas = yas;
    }

    public int getId() {return id;}
    public void setId(int id) {this.id = id;}
    public String getAd() {return ad;}
    public void setAd(String ad) {this.ad = ad;}
    public String getSoyad() {return soyad;}
    public void setSoyad(String soyad) {this.soyad = soyad;}
    public int getYas() {return yas;}
    public void setYas(int yas) {this.yas = yas;}
}
```

5. Bu sınıfın genel işlevinin ne olduğunu hangi eylemleri yerine getirdiğini arkadaşınızla tartışıp aşağıya yazınız.

.....

.....

.....

.....

.....

.....

6. DataHelper adında OrmLiteSqliteOpenHelper sınıfından türeyen sınıfın kodlarını aşağıdaki gibi yazın.

```
import com.j256.ormlite.android.apptools.OrmLiteSqliteOpenHelper;
import com.j256.ormlite.dao.RuntimeExceptionDao;
import com.j256.ormlite.support.ConnectionSource;
import com.j256.ormlite.table.TableUtils;

public class DataHelper extends OrmLiteSqliteOpenHelper {

    private Context context;
    private static final String DATABASE_NAME = "Ogrenci";
    private static final int DATABASE_VERSION = 1;
    private RuntimeExceptionDao<Ogrenci,Integer> studRuntimeDAO=null;
    public DataHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        this.context = context;
    }
    @Override
    public void onCreate(SQLiteDatabase sqLDb, ConnectionSource cS) {
        try {
            TableUtils.createTable(cS, Ogrenci.class);
        } catch (SQLException e){
            e.printStackTrace();
        } catch (java.sql.SQLException e) {
            e.printStackTrace();
        }
    }
    @Override
    public void onUpgrade(SQLiteDatabase sqLDb, ConnectionSource cS, int i, int i2) {
        try {
            TableUtils.dropTable(cS, Ogrenci.class, true);
            onCreate(sqLDb,cS);
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (java.sql.SQLException e) {
            e.printStackTrace();
        }
    }
    public RuntimeExceptionDao<Ogrenci, Integer> getStudRuntimeExceptionDao(){
        if(studRuntimeDAO == null){
            studRuntimeDAO = getRuntimeExceptionDao(Ogrenci.class);
        }
        return studRuntimeDAO;
    }
}
```

7. Yukarıdaki kodların genel işlevini metotlarını inceleyerek arkadaşınızla tartışın ve aşağıya yazın.

.....

.....

.....

.....

.....

8. Son olarak MainActivity sınıfına aşağıdaki kodları ekleyin.

```

import com.j256.ormlite.android.apptools.OpenHelperManager;
import com.j256.ormlite.dao.RuntimeExceptionDao;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button kaydet = (Button) findViewById(R.id.btn_kaydet);
        Button tumunuGoster = (Button) findViewById(R.id.btn_goster);
        Button sil = (Button) findViewById(R.id.btn_sil);
        EditText ad = (EditText) findViewById(R.id.ad);
        EditText soyad = (EditText) findViewById(R.id.soyad);
        EditText yas = (EditText) findViewById(R.id.yas);
        innit();
    }

    public void innit(){

        DataHelper dbHelper = (DataHelper) OpenHelperManager.getHelper(this,DataHelper.class);
        final RuntimeExceptionDao <Ogrenci,Integer> studDao = dbHelper.getRuntimeExceptionDao(Ogrenci.class);

        kaydet.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

if(ad.getText().toString().length()>0&&soyad.getText().toString().length()>0&&yas.getText().toString().length()>0){
    studDao.create(new Ogrenci(ad.getText().toString(), soyad.getText().toString(),
Integer.parseInt(yas.getText().toString())));
    ad.setText("");
    soyad.setText("");
    yas.setText("");
    Toast.makeText(MainActivity.this, "Ogrenci Kaydedildi", Toast.LENGTH_SHORT).show();
}
else{
    Toast.makeText(MainActivity.this, "Bütün Bilgileri Giriniz", Toast.LENGTH_SHORT).show();
}
}
});

        tumunuGoster.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String display = "";
                List<Ogrenci> ogrencis = studDao.queryForAll();
                for(Ogrenci holder : ogrencis){
                    display += holder.getAd()+" "+holder.getSoyad()+" "+holder.getYas()+" \n";
                }
                Toast.makeText(MainActivity.this,"Öğrenciler:" + display, Toast.LENGTH_SHORT).show();
            }
        });

        sil.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                List<Ogrenci> ogrencis = studDao.queryForAll();
                studDao.delete(ogrencis);
                Toast.makeText(MainActivity.this,"Silindi", Toast.LENGTH_SHORT).show();
            }
        });
    }

    public void onDestroy(){
        super.onDestroy();
        OpenHelperManager.releaseHelper();
    }
}

```

9. Yukarıdaki kodlarda innit() metodu içerisinde gerçekleştirilen işlemlerin neler olduğunu arkadaşınızla tartışarak aşağıya yazınız.

.....

.....

.....

.....

.....