



**COMSATS University**  
**Islamabad**  
**Abbottabad Campus**

### **Assignment 3**

<b>SUBMITTED BY</b>	<b>SYED MUHAMMAD SHAHEER ALI SHAH, FAZEELA REHMAN</b>
<b>SUBMITTED TO</b>	<b>MISS SUNDAS SHUJAH</b>
<b>REGISTRATION NO.</b>	<b>FA20-BCS-079, FA20-BCS-090</b>
<b>SUBJECT</b>	<b>DATA BASE</b>
<b>DEPARTMENT</b>	<b>COMPUTER SCIENCE</b>
<b>DATE</b>	<b>06-07-2022</b>



## Project Title:

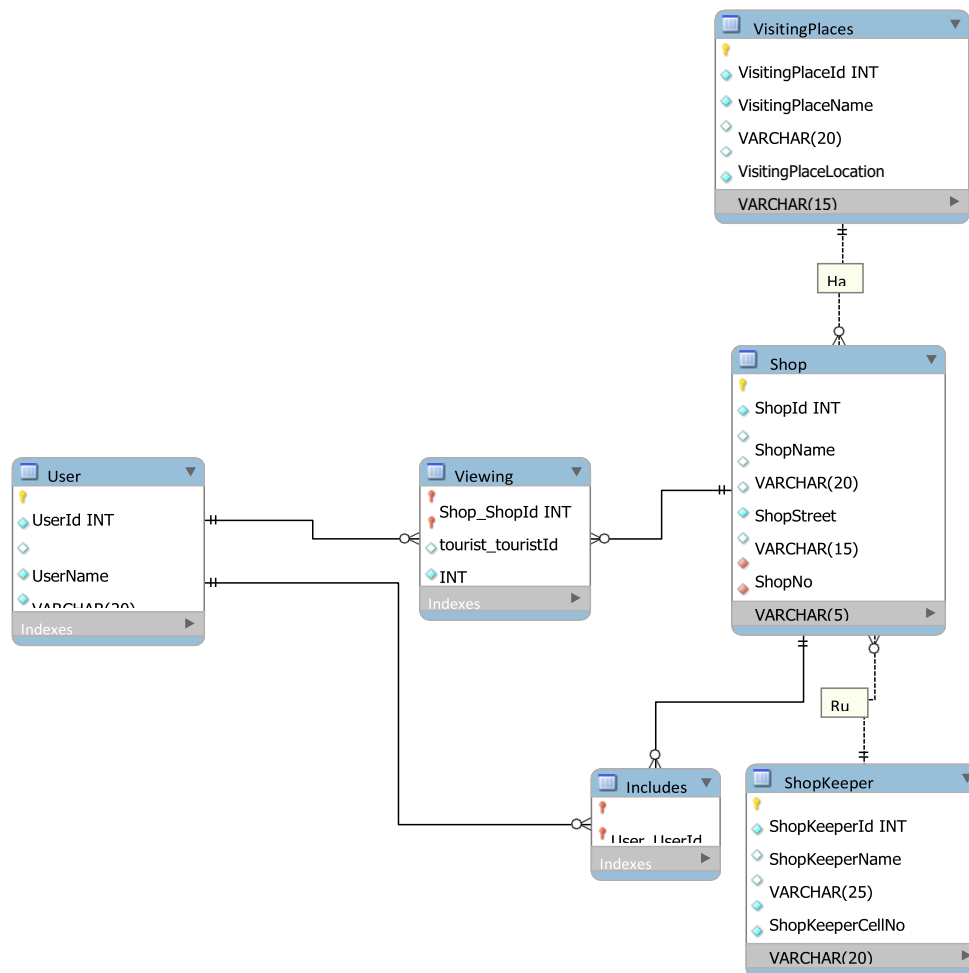
### *Shop Finder*

## Project Overview:

This mobile app is about helping the Users to find shops around places they want to visit (shopping mall, Historical Place and Markets). If they are intending to visit and exploring the various places this application will assist them about a particular place and shops around it. This application also facilitates the user to find local vendors quite easily.

The vendors can register their shops and provide details of shop at a particular place. There will be admins that manages information about visiting places.

## ER Diagram:



## Create Queries:

```
public void onCreate(SQLiteDatabase db) {
    String queryUsers = "create table Users(UserId integer primary key
Autoincrement,UserName varchar(40) not null," +
        " UserCellNo varchar(20) , userEmail varchar(45) not null
unique,\n" +
        "UserPassword Varchar(15) not null)";

    String queryShopKeepers = "create table ShopKeepers(ShopKeeperId integer
primary key Autoincrement,ShopKeeperName text not null"
        + ", ShopKeeperCellNo text, ShopKeeperEmail text unique,
ShopKeeperPassword text not null)";

    String queryVisitingPlaces = "create table VisitingPlaces(PlaceId integer
primary key Autoincrement,PlaceName mediumtext unique not null," +
        "location mediumtext not null unique ,City varchar(20),AboutPlace
mediumtext default'',PlaceType Varchar(15) not null)";

    String queryIncludes="create table Includes(UserId integer ,ShopsId
integer , primary key(UserId,ShopsId),constraint usersincludes_fk foreign
key(UserId) references Users(UserId),\n" +
        "constraint Shopsincludes_fk foreign key(ShopsID) references
Shops(ShopsID) ) ";

    String queryViewing="create table Viewing(ShopsId integer ,UserId integer
,comments varchar(100) default'', ViewDate Date not null
default(CURRENT_DATE)," +
        "primary key (ShopsId,UserId),constraint Userview_fk foreign
key(UserId) references Users(UserId)\n" +
        ",constraint Shopsview_fk foreign key(ShopsId) references
Shops(ShopsId) ) ";

    String queryShops="create table Shops(ShopsId integer primary key
Autoincrement,Shopsname varchar(40) not null, ShopsNum Varchar(5),ShopsStreet
varchar(15)" +
        ", ShopsCity varchar(20),ShopsArea mediumtext,ShopsType
Varchar(15) not null, ShopsRating decimal default 0, ShopkeeperId int,PlaceId
int,\n" +
        "constraint Shopkeepers_shops_fk foreign key(ShopkeeperId)
references ShopKeepers(ShopkeeperId),\n" +
        "constraint VisitingPlaces_shops_fk foreign key(PlaceId)
references VisitingPlaces(PlaceId) )";

    db.execSQL(queryUsers);
    db.execSQL(queryShopKeepers);
    db.execSQL(queryVisitingPlaces);
    db.execSQL(queryShops);
    db.execSQL(queryIncludes);
    db.execSQL(queryViewing);
}

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS Users");
```

```

        db.execSQL("DROP TABLE IF EXISTS ShopKeepers");
        db.execSQL("DROP TABLE IF EXISTS VisitingPlaces");
        db.execSQL("DROP TABLE IF EXISTS Shops");
        db.execSQL("DROP TABLE IF EXISTS Includes");
        db.execSQL("DROP TABLE IF EXISTS Viewing");

        onCreate(db);
    }

```

## Insert Query:

```

public boolean insertIntoShops(String ShopName, String ShopNum, String
ShopStreet,String ShopCity, String ShopArea,String ShopType,int ShopkeeperId,
int PlaceId) {

```

```

    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

    values.put("Shopsname", ShopName);
    values.put("ShopsNum", ShopNum);
    values.put("ShopsStreet", ShopStreet);
    values.put("ShopsCity", ShopCity);
    values.put("ShopsArea", ShopArea);
    values.put("ShopsType", ShopType);
    values.put(" ShopkeeperId", ShopkeeperId);
    values.put("PlaceId", PlaceId);
    long res=db.insert("Shops", null, values);
    db.close();
    if(res!=-1){
        return false;
    }
    else{
        return true;
    }
}

```

```

public boolean insertIntoViewing(int ShopsId, int UserId, String comments) {

```

```

    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

    values.put("ShopsId", ShopsId);
    values.put("UserId", UserId);
    values.put("comments", comments);
    long res=db.insert("Viewing", null, values);
    db.close();
    if(res!=-1){
        return false;
    }
    else{
        return true;
    }
}

```

```

public boolean insertIntoIncludes(int UserId, int ShopsId) {

```

```

        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("UserId", UserId);
        values.put("ShopsId", ShopsId);
        long res=db.insert("Includes", null, values);
        db.close();
        if(res==-1){
            return false;
        }
        else{
            return true;
        }
    }
}

```

```

public boolean insertIntoVisitingPlaces(String PlaceName, String location,
String City,String AboutPlace, String PlaceType) {

```

```

        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("PlaceName", PlaceName);
        values.put("location", location);
        values.put("City", City);
        values.put("AboutPlace", AboutPlace);
        values.put("PlaceType", PlaceType);
        long res = db.insert("VisitingPlaces", null, values);
        db.close();
        if (res == -1) {
            return false;
        } else {
            return true;
        }
    }
}

```

```

public boolean insertIntoShopkeeper(String shName, String shcellNo, String
shEmail, String shPass) {

```

```

        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put("ShopKeeperName", shName);
        cv.put("ShopKeeperCellNo", shcellNo);
        cv.put("ShopKeeperEmail", shEmail);
        cv.put("ShopKeeperPassword", shPass);
        long res=db.insert("ShopKeepers", null, cv);
        db.close();
        if(res==-1){
            return false;
        }
        else{
            return true;
        }
    }
}

```

```

public boolean insertIntoUsers(String usName, String uscellNo, String
usEmail, String usPass) {

```

```

        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();

```

```

values.put("UserName", usName);
values.put("UserCellNo", uscellNo);
values.put("UserEmail", userEmail);
values.put("UserPassword", usPass);
long res=db.insert("Users", null, values);
db.close();
if(res!=-1){
    return false;
}
else{
    return true;
}
}

```

## Login Query:

```

public Cursor getUserLoginEmailPass(String email,String pass){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="select * from Users where UserEmail='"+email+"' and
UserPassword = '"+pass+"'";
    Cursor crs = db.rawQuery(qry,null,null);
    crs.moveToFirst();
    return crs;
}

public Cursor getShopKeeperLoginEmailPass(String email,String pass){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="select * from ShopKeepers where ShopKeeperEmail='"+email+"'
and ShopKeeperPassword = '"+pass+"'";
    Cursor crs = db.rawQuery(qry,null,null);
    crs.moveToFirst();
    return crs;
}

public Cursor getUserLoginEmail(String email){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="select * from Users where UserEmail='"+email+"'";
    Cursor crs = db.rawQuery(qry,null,null);
    crs.moveToFirst();
    return crs;
}

public Cursor getShopKeeperLoginEmail(String email){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="select * from ShopKeepers where ShopKeeperEmail='"+email+"'";
    Cursor crs = db.rawQuery(qry,null,null);
    crs.moveToFirst();
    return crs;
}

public int getNumberOfAccountsUser(String email){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry = "select count(*) from Users where UserEmail='"+email+"'";
    Cursor crs = db.rawQuery(qry,null,null);
    Log.e(TAG, qry+" ");
}

```

```

        Log.e(TAG, qry+" ");
        while(crs.moveToNext()) {
            Log.e(TAG, crs.getString(0)+" ");
        }
        crs.moveToFirst();
        return Integer.parseInt(crs.getString(0)+"");
    }

    public int getNumberOfAccountsShopKeeper(String email){
        SQLiteDatabase db = this.getReadableDatabase();
        String qry = "select count(*) from ShopKeepers where
ShopKeeperEmail='"+email+"'";
        Cursor crs = db.rawQuery(qry,null,null);
        Log.e(TAG, qry+" ");
        Log.e(TAG, qry+" ");
        while(crs.moveToNext()) {
            Log.e(TAG, crs.getString(0)+" ");
        }
        crs.moveToFirst();
        return Integer.parseInt(crs.getString(0)+"");
    }
}

```

## Change Password Query:

```

public void changeUserPassword(int id,String pass){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("UserPassword", pass);
    db.update("Users",values,"Userid=?",new String[]{String.valueOf(id)});
}

public void changeShopKeeperPassword(int id,String pass){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("ShopKeeperPassword", pass);
    db.update("ShopKeepers",values,"ShopKeeperid=?",new
String[]{String.valueOf(id)});
}

```

## Shops of Shopkeeper Query:

```

public Cursor getShopKeeperShop(int id){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="Create view ShopKeeperShopView As select * from Shops where
ShopkeeperId="+id;
    Cursor crs=db.rawQuery(qry,null,null);
    crs.moveToFirst();

    String qry1 ="select * from ShopKeeperShopView";
    Cursor crs1=db.rawQuery(qry1,null,null);
    crs1.moveToFirst();
    return crs1;
}

public void dropShopKeeperShopView() {
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="drop view ShopKeeperShopView";
    Cursor crs=db.rawQuery(qry,null,null);
}

```

```

        crs.moveToFirst();
    }
    public float getShopKeeperShopRating(int id){
        SQLiteDatabase db = this.getReadableDatabase();
        String qry="select ShopsRating from ShopKeeperShopView where
ShopsId="+id;
        Cursor crs =db.rawQuery(qry,null,null);
        crs.moveToFirst();
        return Float.parseFloat(crs.getString(0));
    }
    public String getShopAddress(int id){
        SQLiteDatabase db = this.getReadableDatabase();
        String qry ="select ShopsNum,ShopsStreet,PlaceId,ShopsArea,ShopsCity from
shops where ShopsId="+id;
        Cursor crs=db.rawQuery(qry,null,null);
        crs.moveToFirst();
        String address="Shop# "+crs.getString(0)+" Street# "+crs.getString(1)+"
\n"+crs.getString(3)+" "+
            getVisitingPlaceName(Integer.parseInt(crs.getString(2)))+
"+crs.getString(4);
        return address;
    }
    public Cursor getShopkeeperShopComments(int id){
        SQLiteDatabase db = this.getReadableDatabase();
        String qry="select Users.UserName,Viewing.comments,Viewing.ViewDate from
Users INNER JOIN Viewing on Users.UserId=Viewing.UserId and ShopsId =" +id;
        Cursor crs=db.rawQuery(qry,null,null);
        crs.moveToFirst();
        return crs;
    }
    public void deleteShopkeeperShop(String id){
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete("Includes", "ShopsId=?", new String[]{id});
        db.delete("Viewing", "ShopsId=?", new String[]{id});
        db.delete("Shops", "ShopsId=?", new String[]{id});
        db.close();
    }
    public String getShopKeeperName(int id){
        SQLiteDatabase db = this.getReadableDatabase();
        String qry="select * from ShopKeepers where ShopkeeperId="+id;
        Cursor crs = db.rawQuery(qry,null,null);
        crs.moveToFirst();

        return crs.getString(1);
    }
}

```

## Change Detail of Shop Query:

```

public void changeShopStreet(int id,String street){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("ShopsStreet", street);
    db.update("Shops",values,"ShopsId=?",new String[]{String.valueOf(id)});
}
public void changeShopCity(int id,String city){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

```



```

        values.put("ShopsCity", city);
        db.update("Shops", values, "ShopsId=?", new String[]{String.valueOf(id)});
    }
    public void changeShopArea(int id, String area) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("ShopsArea", area);
        db.update("Shops", values, "ShopsId=?", new String[]{String.valueOf(id)});
    }
    public void changeShopType(int id, String type) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("ShopsType", type);
        db.update("Shops", values, "ShopsId=?", new String[]{String.valueOf(id)});
    }
    public void changeShopvsPlace(int id, String vsPlace) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();

        values.put("PlaceId", getIdVisitingPlace(vsPlace));
        db.update("Shops", values, "ShopsId=?", new String[]{String.valueOf(id)});
    }

    public void changeShopNumber(int id, String number) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("ShopsNum", number);
        db.update("Shops", values, "ShopsId=?", new String[]{String.valueOf(id)});
    }
    public void changeShopName(int id, String name) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("Shopsname", name);
        db.update("Shops", values, "ShopsId=?", new String[]{String.valueOf(id)});
    }
}

```

## Visiting Place Query:

```

public int getIdVisitingPlace(String VPlace) {
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="select * from VisitingPlaces where PlaceName='"+VPlace+"'";
    Cursor crs = db.rawQuery(qry, null, null);
    crs.moveToFirst();
    return Integer.parseInt(crs.getString(0));
}
public String getVisitingPlaceName(int id) {
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="select PlaceName from VisitingPlaces where PlaceId="+id;
    Cursor crs = db.rawQuery(qry, null, null);
    crs.moveToFirst();
    return crs.getString(0);
}

```

## User Operations Query:

```
public String getUsername(int id){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="select * from Users where UserId="+id;
    Cursor crs = db.rawQuery(qry,null,null);
    crs.moveToFirst();

    return crs.getString(1);
}
public Cursor getIncludedShops(int id){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry ="Create view viewIncludedShopsUser As select * from Shops
inner join Includes on Shops.ShopsId=Includes.ShopsId and
Includes.Userid="+id;
    Cursor crs=db.rawQuery(qry,null,null);
    crs.moveToFirst();

    String qry1="select * from viewIncludedShopsUser";
    Cursor crs1=db.rawQuery(qry1,null,null);
    crs1.moveToFirst();
    return crs1;
}
public void dropIncludedShops(){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="drop view viewIncludedShopsUser";
    Cursor crs=db.rawQuery(qry,null,null);
    crs.moveToFirst();
}
public float getUserShopRating(int id){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="select ShopsRating from viewIncludedShopsUser where
ShopsId="+id;
    Cursor crs =db.rawQuery(qry,null,null);
    crs.moveToFirst();
    return Float.parseFloat(crs.getString(0));
}
public Cursor getUserShopComments(int id){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="select Users.UserName,Viewing.comments,Viewing.ViewDate from
Users INNER JOIN Viewing on Users.UserId=Viewing.UserId and comments !='' and
ShopsId =" +id;
    Cursor crs=db.rawQuery(qry,null,null);
    crs.moveToFirst();
    return crs;
}

public String getShopAddress(int id){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry ="select ShopsNum,ShopsStreet,PlaceId,ShopsArea,ShopsCity from
shops where ShopsId="+id;
    Cursor crs=db.rawQuery(qry,null,null);
    crs.moveToFirst();
    String address="Shop# "+crs.getString(0)+" Street# "+crs.getString(1)+"
\n"+crs.getString(3)+" "+
        getVisitingPlaceName(Integer.parseInt(crs.getString(2)))+"
```

```

"+crs.getString(4);
    return address;
}

public void deleteShopFromList(String userid,String shopid){
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete("Includes", "UserId=? and ShopsId=?", new
String[]{userid,shopid});
    db.close();
}

```

## User Search Operation Query:

```

public Cursor userSearchShopByVsPlace(String search){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="select * from shops where PlaceId in (select placeId from
visitingplaces where PlaceName like '%" +search+" ' or PlaceName like
 '%" +search+"%' or PlaceName "
        + "like '" +search+"%' or city like '%" +search+" ' or city like
 '%" +search+"%' or city like '" +search+"%' or PlaceType like '%" +search+" ' or
PlaceType like "
        + "' '%" +search+"%' or PlaceType like '" +search+"%' or location
like '%" +search+" ' or location like '%" +search+"%' or location like
 '%" +search+"%'");
    Cursor crs=db.rawQuery(qry,null,null);
    crs.moveToFirst();
    return crs;
}

public Cursor userSearchShopByShop(String search){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="select * from shops where shopsname like '%" +search+" ' or
shopsname like '%" +search+"%' or shopsname like '" +search+"%' or ShopsCity
like '%" +search+" ' \n" +
        "or ShopsCity like '%" +search+"%' or ShopsCity like '" +search+"%'
or ShopsArea like '%" +search+" ' or ShopsArea like '%" +search+"%' or ShopsArea
like '" +search+"%' \n" +
        " or ShopsType like '%" +search+" ' or ShopsType like
 '%" +search+"%' or ShopsType like '" +search+"%' ; ";
    Cursor crs=db.rawQuery(qry,null,null);
    crs.moveToFirst();
    return crs;
}

public Cursor userSearchShopByVsPlaceAndShop(String search1,String search2){
    SQLiteDatabase db = this.getReadableDatabase();
    String qry="select * from shops,visitingPlaces where
shops.placeId=visitingPlaces.placeId and (PlaceName like '%" +search1+" ' or
PlaceName like '%" +search1+"%' "
        + "or PlaceName like '" +search1+"%' or city like '%" +search1+" '
or city like '%" +search1+"%' or city like '" +search1+"%' or PlaceType like
 '%" +search1+" ' "
        + " or PlaceType like '%Shooping mall' or PlaceType like
'Shooping mall%' or location like '%Shooping mall' or location like
 '%" +search1+"%' "
        + "or location like '" +search1+"%') and (shopsname like
 '%" +search2+" ' or shopsname like '%" +search2+"%' or shopsname like

```

```

        '"+search2+"%" ' "
            + "or ShopsCity like '%" +search2+"'" or ShopsCity like
        '"+search2+"%" or ShopsCity like '"+search2+"%" or ShopsArea like
        '"+search2+"'" "
            + "or ShopsArea like '%" +search2+"%" or ShopsArea like
        '"+search2+"%" or ShopsType like '%" +search2+"'" or ShopsType like
        '"+search2+"%" ' "
            + "or ShopsType like '"+search2+"%"' " ";
        Cursor crs=db.rawQuery(qry,null,null);
        crs.moveToFirst();
        return crs;
    }

    public boolean isViewAlreadyViewed(int userid,int shopid){
        SQLiteDatabase db = this.getReadableDatabase();
        String qry ="select * from Viewing where UserId="+userid+" and ShopsId
    =" +shopid;
        Cursor crs=db.rawQuery(qry,null,null);
        crs.moveToFirst();
        return crs.getCount()==0;
    }

    public void updateRating(float Rate,int shopid){
        SQLiteDatabase db = this.getReadableDatabase();
        String qryOldRating ="select ShopsRating from Shops where
    ShopsId="+shopid;
        Cursor crsOldRating=db.rawQuery(qryOldRating,null,null);
        crsOldRating.moveToFirst();
        float oldRating=Float.parseFloat(crsOldRating.getString(0));

        String qryCount ="select * from Viewing where ShopsId="+shopid;
        Cursor crsCount=db.rawQuery(qryCount,null,null);
        int count=crsCount.getCount()-1;
        float newRate=((oldRating*count)+Rate)/(count+1);

        SQLiteDatabase dbWrite = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("ShopsRating", newRate);
        dbWrite.update("Shops",values,"ShopsId=?",new
    String[]{String.valueOf(shopid)});
    }

    public boolean isAlreadySave(int userid,int shopid){
        SQLiteDatabase db = this.getReadableDatabase();
        String qry ="select * from Includes where UserId="+userid+" and ShopsId
    =" +shopid;
        Cursor crs=db.rawQuery(qry,null,null);
        crs.moveToFirst();
        return crs.getCount()==0;
    }

```