

بسم الله الرحمن الرحيم

درس : برنامه نویسی موبایل ۲

استاد : استاد پیمان مجیدی

موضوع : SQL در اندروید

## SQL یا پایگاه داده چیست :

پایگاه داده یا بانک اطلاعاتی یا دیتابیس مجموعه ای متشکل از این داده‌های ساخت یافته است که در قالبی ذخیره شده و توسط ابزارهای الکترونیکی به راحتی در دسترس، کنترل و بروزرسانی است. یک دیتابیس یا پایگاه داده، می‌تواند حاوی یک یا چند جدول باشد. هر جدول هم می‌تواند حاوی ستون‌ها و سطرهاى مختلفى باشد که اطلاعات را بصورت سازماندهی شده نگهداری می‌کند.

برای ساخت یک SQLite از یک سایتی به نام

[/https://www.tutorialspoint.com](https://www.tutorialspoint.com)

کمک می‌گیریم

نکته : برای این از سایت بالا کمک می‌گیرم که خودش دستورات رو نوشته و فقط ما کپی میکنیم و لازم نیست خودمون بیاییم و بنویسیم و ی سری تغییرات جزئی میدهم

## مراحل ساخت دیتابیس:

برای ساخت یک دیتابیس SQLite به صورت زیر کد می‌زنیم :

```
SQLiteDatabase mydatabase = openOrCreateDatabase( name: "JabarianDB", MODE_PRIVATE, factory: null);
```

نام آبجکت

نام دیتابیس

نکته : برای اولین بار وقتی دیتابیس رو میسازیم **openOrCreateDatabase** ساخته میشود ولی از دفعه بعدی **open** میشود

خب حالا که ما توانستیم یک دیتابیس بسازیم باید برای این دیتابیس یک جدول بسازیم که مقادیر بتوانند داخل آن استفاده شود  
برای ساخت جدول بصورت زیر عمل میکنیم :



ابتدا یک نام برای جدول انتخاب میکنیم طبق تصویر بالا  
و سپس مقادیر را با نوع آن وارد میکنیم

نکته : مقادیر میتواند بصورت عددی هم باشد ...

حالا بعد از ساخت جدول ما باید مقادیر را در آن درج کنیم برای درج مقدار در جدول پیش فرض بصورت زیر عمل میکنیم :

```
mydatabase.execSQL("INSERT INTO Users VALUES('admin','12345');");
```

نام جدولی که مقادیر در آن درج میشوند

مقادیر جدول

خب ما به این صورت مقادیر را در جدول درج کردیم

در مرحله اول نام جدولی را که ساخته ایم وارد میکنیم ک جدول **Users** را در بالا ساختیم

برای مقداردهی نسبت به فیلد های جدول مقدار وارد میکنیم :

به عنوان مثال در جدول **Users** ما دو نوع مقدار **user and password** تعریف شده بود پس برای درج مقادیر باید دو مقدار را وارد کنیم مانند تصویر بالا

الان که ما توانستیم یک دیتابیس بسازیم و جداول و مقادیر آن را وارد کنیم نوبت آن رسیده که آن را از این دیتابیس و جدول بخوانیم و در لاگ برنامه نشان بدهیم برای اینکار بصورت زیر عمل میکنیم :

```
Cursor resultSet = mydatabase.rawQuery( sql: "Select * from Users", selectionArgs: null);  
resultSet.moveToFirst();  
String username = resultSet.getString( i: 0);  
String password = resultSet.getString( i: 1);
```

خب ابتدا میاییم یک اشاره گر یا **Cursor** درست میکنیم و به ابجکت

**myDatabase** دسترسی پیدا میکنیم و از جدول **Users** تمامی مقادیر را میخوانیم

در خط بعدی از **moveToFirst()** این تابع برای محکم کاری استفاده میکنیم که کوئری هارو بره از اول جدول بخواند میتوانیم این خط را ننویسیم اما کار از محکم کاری عیب نمیکند

در خط بعدی ما میایم مقادیر که که وارد کرده ایم را میخوانیم و آن را پیدا میکنیم در خط سوم از ایندکس ۰ که به **Username** ما در جدول **Users** اشاره میکند استفاده میکنیم و در خط آخر از ایندکس ۱ که به **Password** ما در جدول **Users** اشاره میکند استفاده میکنیم

نکته : اگر جدولی ک ما در بالا ساختیم ی مقدار دیگری به نام شماره تلفن داشت ما برای خواندن ان باید از ایندکس ۳ استفاده میکردیم

خب بعد از اینکه ما **SqlLite** را ساختیم و جدواول و مقادیر را وارد کردیم و سپس آن را خواندیم الان باید آن را نمایش دهیم برای نمایش آن به صورت زیر عمل میکنیم :

```
String TAG = "DATABASE_JABARIAN";
```

```
Log.d(TAG, msg: "username = " + username);
```

برای اینکه بتوانیم در لاگ برنامه مشاهده کنیم مقادیری را که با استفاده از کوئری گرفتیم ابتدا مانند تصویر بالا که در خط شماره ۱ یک متغیری تعریف میکنیم به نام دیتابیس ما و سپس در خط ۲ ان متغیر را در لاگ صدا میزنیم و در مقدار دوم تابع لاگ نامی ک میخاهیم آن را در لاگ نمایش دهد + ان مقدار از

کوئری ک در بالا خوانده بودیم که ما اینجا username وارد کردیم میتوانیم حتی password هم وارد کنیم

خب حالا اگر ما بخوایم به ازای هرباری ک پروژه رو اجرا میکنیم یک بار به هر مقداری ک داخل جدول وارد کرده ایم اضافه شود به این صورت عمل میکنیم :

```
for(int i=0;i< resultSet.getCount(); i++) {  
    String username = resultSet.getString(resultSet.getColumnIndex( 5: "Username"));  
    String password = resultSet.getString( 1);  
    Log.d(TAG, msg: "username = " + username);  
  
    resultSet.moveToNext();  
}
```

خب! ما اومدیم مقادیری را که از جدول دریافت کردیم چه بصورت ایندکس یا چه بصورت نام آن فیلد داخل یک حلقه ریختیم ...

این حلقه مقدار اولیش برابر ۰ است و مقدار نهاییش برابر تمامیفیلد هایی هست که داخل جدول ما قرار دارد که اینا ۱ هست یعنی دو مقدار ...

و هر بار ک این حلقه تکرار میشود مثلا ++ میشود یعنی ۱ بار دیگر از اون فیلد و مقادیر آن تکرار میشود

برای مثال : اولین بار ک بخایم پروژه را ران کنیم ۲ مقدار داریم

اما برای دومین بار ۴ مقدار داریم ک از مقدار اول ۲ بار و از مقدار دوم هم ۲ بار تکرار شده است

# SQLITE

خب ما تا اینجا کار توانستیم که دسترسی پیدا کنیم ب مقدار های SqlLite

حالا ما میخواهیم دیتابیس رو از پروژه اندرویدی برداریم و آن را داخل یک نرم افزار به اسم DB Browser SqlLite بارگزاری کنیم

این نرم افزار تمام قابلیت های mySql را داخل خود دارد اما بصورت تک فایل دیتابیس ها را میگیرد و اجرا میکند

برای دانلود این نرم افزار میتوانید از لینک زیر استفاده کنید

[/https://sqlitebrowser.org/dl](https://sqlitebrowser.org/dl)

نکته: توجه داشته باشید برای دانلود نرم افزار باید از فیلتر شکن استفاده کنید

بعد از اینکه ما نرم افزار را دانلود کردیم باید فایل دیتابیس خود را که داخل پروژه ساختیم پیدا کنیم و از آن درون این نرم افزار استفاده کنیم برای اینکار:

دیتابیس > اسم پروژه > data > data > DeviceFileExplorer

به این صورت عمل میکنیم و فایل دیتابیس را ذخیره میکنیم

بعد از اینکه نرم افزار SQLite را نصب کردیم فایل دیتابیس را داخل آن وارد میکنیم

با استفاده از این نرم افزار میتوانیم دسترسی پیدا کنیم به تمام جداول و مقادیری که درون دیتابیس داخل پروژه ساختیم ...

حتی میتوانیم خودمان یک جدول دیگر بصورت دستی بسازیم و مقادیر را به جداول اضافه کنیم

نکته : توجه داشته باشید پس از هر اعمالی که در نرم افزار SQLite انجام میدهید برای ذخیره آن :

File > Right changes

خب حالا ما توانستیم یک دیتابیس را از پروژه برداریم و درون نرم افزار استفاده کنیم

اما حالا میخواهیم که جدولی را که داخل این نرم افزار ساختیم را درون پروژه استفاده کنیم برای اینکار فایل دیتابیس را درونه مسیر

دیتابیس > اسم پروژه > data > data > DeviceFileExplorer

اپلود میکنیم به همین راحتی!

خب حالا برای اینکه دسترسی داشته باشیم به جدول جدیدی که ساختیم و آن را داخل لاگ پروژه نمایش بدیم بصورت زیر عمل میکنی :

```
Cursor resultSet = mydatabase.rawQuery( sql: "Select * from Categories", selectionArgs: null);
resultSet.moveToFirst();
for(int i=0;i< resultSet.getCount(); i++) {

    Log.d(TAG, msg: resultSet.getString( i 1) + " " + resultSet.getString( i 2));

    resultSet.moveToNext();
}
```

اومدیم داخل جدول Categories و تمام مقادیری که داخل این جدول هستند را برای ما برمیگرداند ...

داخل لاگ هم اومدیم مقادیر ۱ و ۲ ایندکس را در خروجی نمایش میدهیم

نکته : مقدار اول ۰ است و چون در بالا نیامده پس نمایش داده نمیشد چون این جدول ما سه فیلد داشت !

**DataBase-Helper:**



پروژه ما در طول توسعه نیاز به تغییراتی دارد که ما با استفاده از کپسوله سازی یا کلاس در اندروید میایم و تغییراتی رو اعمال میکنیم این تغییرات رو یک بار اعمال میکنیم و کار ما را ساده تر میکند

مثلا نیازی نیست ما هربار کوئری بگیریم یک بار ان را مینویسیم و در دفعات بعدی از یک نمونه میسازیم

خب بعد از اینکه ما با **DataBase-Helper** آشنا شدیم برای استفاده از این باید یک کلاسی را در جاوا بسازیم  
بعد از ساخت کلاس باید دیتابیس ک داخل پروژه ساختیم را به این کلاس ربط بدیم که از extend استفاده میکنیم



بعد از ساخت کلاس و ربط دادن آن به دیتابیس به یک اخطار برمیخوریم ک در تصویر بالا مشاهده میکنیم که به ما میگه تابع سازنده را بسازید  
خب برای ساخت تابع سازنده بصورت زیر عمل میکنیم :

```

public class DBHelper extends SQLiteOpenHelper {

    public DBHelper(Context context, int version) {
        super(context, name: null, factory: null, version);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {

    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

    }
}

```

الان کلاس ما آمادس اما یک نکته خیلی مهم وجود دارد اونم Version هست که در تصویر بالا مشاهده میکنید

ممکن است دیتابیس ما در طول توسعه تغییراتی داشته باشد و با نسخه قبلی سازگار باشد اما با نسخه جدید نه پس بهتر است که از Version استفاده کنیم

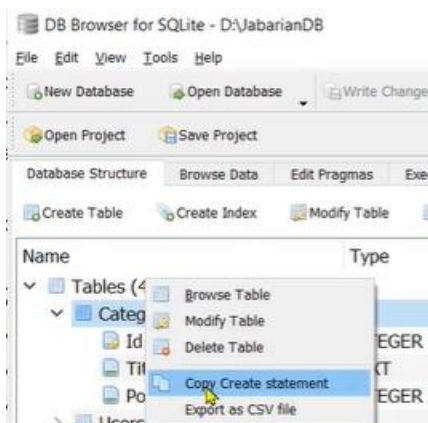
خب بعد از انجام این مراحل حالا نوبت این رسیده ک پروژه خود را به کلاس تبدیل کنیم و بصورت ساده تر از آن استفاده کنیم برای اینکار نیاز داریم ک اسم دیتابیس و جدولی ک میخاییم داشته باشیم را بصورت سراسری تعریف کنیم :

```

public final String DATABASE_NAME = "JabarianDB";
public final String CATEGORY = "Categories";

```

خب برای ساخت جدول برای اینکه کد نویسی نکنیم از نرم افزار SQLite استفاده میکنیم



برای اینکار طبق تصویر فوق بر روی جدول کلیک راست میکنیم و سپس گزینه مورد نظر داخل عکس را کلیک میکنیم بعد از این کار نوت پد ویندوز را باز کرده و کد را داخل آن میگذاریم و کد ساخت جدول را بما میدهد

خب بعد از انجام اینکار توسط این نرم افزار الان نوبت آن رسیده ک جدول را بسازیم برای ساخت جدول به این صورت عمل میکنیم :

```
public void onCreate(SQLiteDatabase db) {
    String createCategory = "CREATE TABLE \"Categories\" (\n" +
        "\t\"Id\" \tINTEGER,\n" +
        "\t\"Title\" \tTEXT,\n" +
        "\t\"Point\" \tINTEGER,\n" +
        "\tPRIMARY KEY(\"Id\" AUTOINCREMENT)\n" +
        ")";
}
```

که یک متغیری تعریف کرده و جدول ساخته شده را به آن پاس میدیم

خب همنجوری که بالا مشاهده میکنید ما مقادیر را بصورت رشته ای داریم

خب این کارا سخت میکند چون شاید یک اشتباه تایپی داشته باشیم و آن را پیدا نکنیم و کل پروژه به مشکل بخوره برای رفع این مشکلا ما نیاز داریم که دوتا کلاس جدید طراحی کنیم

یک کلاس شامل id , title , point

```
public class CategoryHelper {  
    public static String Id = "Id";  
    public static String Title = "Title";  
    public static String Point = "Point";  
}
```

که نام آن هست CategoryHelper

یک کلاس دیگر با نام Category ایجاد میکنیم و مقادیر زیر را برایش اعمال میکنیم همچنین سه تا تابع سازنده برایش میسازیم ک مقادیر را از ورودی دریافت کند بصورت متفاوت تابع اول شامل سه تا فیلد تابع دوم شامل دو تا فیلد و تابع اخر شامل یک فیلد برای اینکار بصورت زیر عمل میکنی :

```
public class Category {  
    public int Id;  
    public String Title;  
    public int Point;  
  
    public Category(int id, String title, int point) {  
        Id = id;  
        Title = title;  
        Point = point;  
    }  
  
    public Category(String title, int point) {  
        Title = title;  
        Point = point;  
    }  
  
    public Category() {  
        |  
    }  
}
```

خب بعد از اینکه اینکارو کردیم میایم و و از اون کلاس ها داخل پروژه دقیقا اونجایی که از رشته استفاده کرده بودیم برای ساخت جدول و مقادیر را بهش پاس میدیم بصورت زیر :

```
@Override
public void onCreate(SQLiteDatabase db) {
    String createCategory = "CREATE TABLE \"" + CATEGORY.toString() + "\" (\n" +
        "\t\"" + CategoryHelper.Id + "\"\tINTEGER,\n" +
        "\t\"" + CategoryHelper.Title + "\"\tTEXT,\n" +
        "\t\"" + CategoryHelper.Point + "\"\tINTEGER,\n" +
        "\tPRIMARY KEY(\" + CategoryHelper.Id + "\" AUTOINCREMENT)\n" +
        ")";

    db.execSQL(createCategory);
}
```



خب تقریبا کار تمامه اما در تصویر بالا اگر فلش قرمز را را ببینید این دستور داره به متغیر یا استرینگ بالا اشاره میکنه و اون رو از db یا تابع سازنده میخونه

خب اگر در بالا دیده باشید ما دوتا تابع ساختیم یکی برای ساخت دیتابیس و یکی برای اپدیت آن چرا اینکارو کردیم ؟ به این دلیل که اگر ک موقع دیتابیس ما اپدیت شد به ورژن رجوع کنه و آن را اپدیت کند  
برای اینکار داخل تابع خودش به این صورت عمل میکنیم :

```
@Override
public void onUpgrade(SQLiteDatabase db, int i, int i1) {
    db.execSQL("DROP TABLE IF EXISTS " + CATEGORY);
    onCreate(db);
}
```

خب همانطور که در بالا مشاهده میکنید این تابع میره از جدلی ک ما ساختیم و متغیرش را تعریف کردیم دیتابیس رو اپدیت میکنه و مقادیر جدید را میخونه

خب الان نوبت آن رسیده که مقادیر جدول را وارد کنیم ما باید یک تابع ای داشته باشیم ک از نوع Boolean باشه و بخونه یا insert کنه از جدولی ک ما میخایم ما در اینجا جدول Category را میخاییم بعد از انجام اینکار ما باید مقادیر جدید را در دیتابیس بنویسیم

برای اینکه این هارو به دیتابیس اضافه کنیم از put استفاده میکنیم

و در اخر مقدار که جدول را در ان ذخیره کرده به دیتابیس پاس میدهیم و مقدار بازگشتی آن را برابر درست قرار میدهیم چرا ک تابع ما یک تابع صحیح غلط یا بولین بود

مانند تصویر زیر عمل میکنیم :

```
public boolean insert(Category category)
{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(CategoryHelper.Title, category.Title);
    contentValues.put(CategoryHelper.Point, category.Point);

    db.insert(CATEGORY, nullColumnHack: null, contentValues);
    return true;
}
```

خب بعد از اینکه ما مقادیر را داخل جدول نمایش دادیم الان لازمه که ما بتوانیم تابعی بنویسیم ک این مقادیر را اپدیت کند دقیقا طبق توضیحات بالا عمل میکنیم

```

public boolean update(int id, Category category)
{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(CategoryHelper.Title, category.Title);
    contentValues.put(CategoryHelper.Point, category.Point);

    db.update(CATEGORY, contentValues, whereClause: "id = ? ", new String[] { Integer.toString(id) } );
    return true;
}

```

خب مشاهده میکنید که داخل تابع اپدیت یک متغیری تعریف شده که این متغیر به اون فیلد درون دیتابیس داره اشاره میکنه چون باید دیتابیس بدونه کدوم فیلد رو داره اپدیت میکنه ...

خب برای دلیت کردن از دیتابیس ما باید با یک ایدی بگیم که داریم کدوم فیلد یا مقدار رو از دیتابیس پاک میکنیم چرا که ما نوع یا تایپ آن را نمیدونیم و باید بصورت تکی حذف کنیم مانند تصویر زیر :

```

public Integer deleteCategory(int id) {
    SQLiteDatabase db = this.getWritableDatabase();
    return db.delete(CATEGORY,
        whereClause: "id = ? ",
        new String[] { Integer.toString(id) });
}

```

خب الان نوبت آن رسیده که تمامی اطلاعات که درج کردیم را بخوانیم و از آن استفاده کنیم برای اینکار ما نیاز به یک لیست داریم از جدولی که ساختیم مقادیر را بخوانیم و یکی یکی مقادیری که در کلاس CategoryHelper استفاده کردیم

یکی یکی بخوانیم و آن را لیست این تابع اد کنیم یا اضافه کنیم

سپس مقدار بازگشتی را نمایش دهیم

نکته مهم این هست که نوع داده point ما از نوت عددی است پس باید از getint استفاده کنیم

مانند تصویر زیر این تابع را نیز میسازیم با توجه به توضیحات بالا :

```
public ArrayList<Category> getAllCategories() {
    ArrayList<Category> array_list = new ArrayList<Category>();

    //hp = new HashMap();
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor res = db.rawQuery( sql: "SELECT * FROM " + CATEGORY, selectionArgs: null );
    res.moveToFirst();

    while(res.isAfterLast() == false){
        Category category = new Category();
        category.Title = res.getString(res.getColumnIndex(CategoryHelper.Title));
        category.Point = res.getInt(res.getColumnIndex(CategoryHelper.Point));
        array_list.add(category);
        res.moveToNext();
    }
    return array_list;
}
```

خب بعد از اینکه مقادیر را خواندیم حتما باید تعداد مقادیر جدول را بتوانیم بدست بیاوریم برای اینکار طبق تصویر زیر عمل میکنیم :

```
public int count(){
    SQLiteDatabase db = this.getReadableDatabase();
    int numRows = (int) DatabaseUtils.queryNumEntries(db, CATEGORY);
    return numRows;
}
```

و بعد از این که تعداد الان را بدست آوردیم الان باید دیتا را بخوانیم و مقدار هارو بدست بیاوریم که مانند تصویر زیر عمل میکنیم :

```
public Cursor getCategory(int id) {
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor res = db.rawQuery( sql: "SELECT * FROM " + CATEGORY + " WHERE " + CategoryHelper.Id + "=" + id + "", selectionArgs: null );
    return res;
}
```

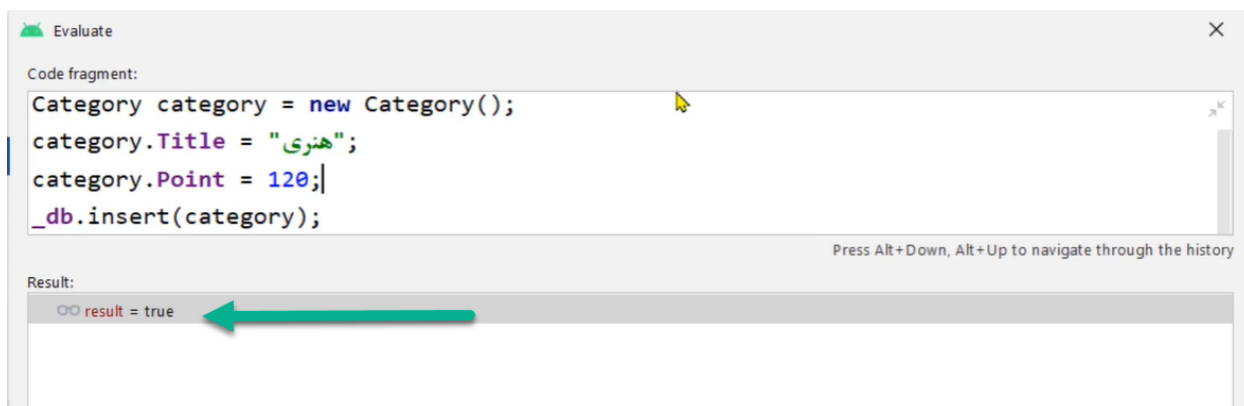


و کار ما تمومه و توانستیم که توسعه بدیم با استفاده از کلاس ها در اندروید  
الان میاییم و داخل MainActivity تمام چیز هایی را ک نوشتیم پاک میکنیم و  
یک نمونه از کلاس هایی ک ساختیم رو میسازیم  
خب برای این کار به این صورت عمل میکنیم :

```
DBHelper _db;
```

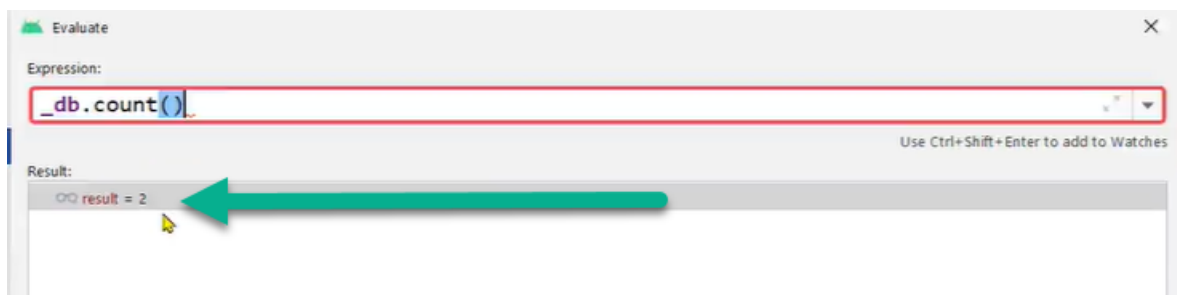
```
_db = new DBHelper( context: this, version: 1);|
```

خب بعد از تمامی این کارهایی ک انجام دادیم وقت آن رسیده که ببینیم دستوراتی  
ک نوشتیم به درستی کار میکنن و دیباگش کنیم برای این کار ما وارد بخش  
Evaluate میشیم که این امکان رو به ما میده ک بصورت بر خط کد بزنینم  
خب برای افزودن یک فیلد به دیتابیس بصورت زیر عمل میکنیم :



همانطور که میبینید ما از Category یک نمونه ساختیم و عنوان و امتیازش رو  
دادیم و سپس به دیتابیسمون اضافه کردیم به این صورت میتونیم ی فیلد را به  
دیتابیس اضافه کنیم

خب برای اینکه بتوانیم تعداد فیلد هایی ک اضافه کردیم را مشاهده کنیم از دستور زیر استفاده میکنیم:



اما اگر بخواهیم یک لیستی از تمامی فیلد هایی ک به جدول اضافه کردیم را بدست بیاریم باید طبق دستور زیر عمل کنیم :

