



الفصل الرابع عشر: الخدمات Services

الصفحة	العنوان
3	1. قواعد البيانات المحمولة SQLite
4	قواعد البيانات SQLite
4	إنشاء قاعدة البيانات
4	إنشاء جدول في قاعدة البيانات
5	مثال تعليمي: نظام إدارة طلاب بسيط

الكلمات المفتاحية:

قاعدة بيانات، جدول، استعلام.

ملخص:

نستعرض في هذا الفصل أساسيات التعامل مع قواعد البيانات المحمولة SQLite.

أهداف تعليمية:

يتعرف الطالب في هذا الفصل على:

- إنشاء قاعدة بيانات.
- إنشاء الجداول.
- إضافة/تعديل/حذف التسجيلات.
- الاستعلام عن التسجيلات

المخطط:

التعامل مع قواعد البيانات المحمولة

- 1 وحدة (Learning Objects)

1. قواعد البيانات المحمولة SQLite

الأهداف التعليمية:

- قواعد البيانات المحمولة SQLite

قواعد البيانات SQLite

- يوفر الصف SQLiteDatabase من الحزمة android.database.sqlite والصف Cursor من الحزمة android.database جميع الوظائف المطلوبة لإنشاء جداول قاعدة البيانات وإجراء الاستعلامات عليها.

إنشاء قاعدة البيانات

- يتم إنشاء قاعدة بيانات باستخدام الطريقة openOrCreateDatabase كما يُبين المثال التالي:

```
SQLiteDatabase db;
db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE, null);
```

- تقوم الطريقة openOrCreateDatabase بفتح قاعدة البيانات StudentDB إن كانت موجودة سابقاً، وإلا فإنها تقوم بإنشاء قاعدة بيانات جديدة باسم StudentDB.
- يُحدّد المعامل الأول للطريقة اسم قاعدة البيانات المطلوب إنشائها أو فتحها.
- يُحدّد المعامل الثاني Context.MODE_PRIVATE بأن قاعدة البيانات يُمكن الوصول لها فقط من قبل التطبيق المستدعي أو أي تطبيق آخر له نفس مُعرّف المستخدم user ID.
- يكون المعامل الثالث عبارة عن غرض من النوع Cursor factory ويُمكن تركه null إذا كان غير مطلوباً.

إنشاء جدول في قاعدة البيانات

- يُمكن استخدام الطريقة execSQL التي يوفرها الصف SQLiteDatabase لتنفيذ الأوامر SQL على قاعدة البيانات. نقوم في المثال التالي بإنشاء جدول الطلاب student والذي يتألف من ثلاثة أعمدة:
 - رقم تسجيل الطالب rollno
 - اسم الطالب name
 - علامة الطالب marks

```
db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno VARCHAR,name VARCHAR,marks VARCHAR);");
```

مثال تعليمي: نظام إدارة طلاب بسيط

- نقوم في المثال التعليمي التالي بإنشاء نظام إدارة للطلاب بسيط يُمكن المستخدم من إضافة طالب، حذف طالب، تعديل بيانات طالب، معاينة بيانات طالب، معاينة بيانات كل الطلاب.
- يُبين الشكل التالي واجهة نظام إدارة الطلاب البسيط:

- تحوي الواجهة ثلاثة صناديق تحرير نص لإدخال بيانات الطالب: رقم الطالب، اسم الطالب، علامة الطالب.
- كما تحوي الواجهة الأزرار التالية:
 - ADD (إضافة): إضافة بيانات الطالب المُدخلة إلى جدول الطلاب.
 - DELETE (حذف): حذف الطالب المُدخل رقمه في صندوق رقم الطالب.
 - MODIFY (تعديل): تعديل بيانات الطالب (الاسم والعلامة) المُدخل رقمه في صندوق رقم الطالب.

- VIEW (عرض): عرض بيانات الطالب المُدخل رقمه في صندوق رقم الطالب.
- VIEW ALL (عرض الكل): عرض بيانات كل الطلاب.
- SHOW (إظهار): عرض رسالة تعريفية.
- يكون ملف النشاط activity_main.xml الموافق للواجهة السابقة:

```
<AbsoluteLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    android:id="@+id/myLayout"

    android:stretchColumns="0"

    android:layout_width="fill_parent"

    android:layout_height="fill_parent">

    <TextView android:text="@string/title"

        android:layout_x="110dp"

        android:layout_y="10dp"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"/>

    <TextView android:text="@string/roll_no"

        android:layout_x="30dp"

        android:layout_y="50dp"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"/>

    <EditText android:id="@+id/editRollno"

        android:inputType="number"

        android:layout_x="150dp"

        android:layout_y="50dp"

        android:layout_width="150dp"
```

```
        android:layout_height="40dp"/>

<TextView android:text="@string/name"

        android:layout_x="30dp"

        android:layout_y="100dp"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"/>

<EditText android:id="@+id/editName"

        android:inputType="text"

        android:layout_x="150dp"

        android:layout_y="100dp"

        android:layout_width="150dp"

        android:layout_height="40dp"/>

<TextView android:text="@string/marks"

        android:layout_x="30dp"

        android:layout_y="150dp"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"/>

<EditText android:id="@+id/editMarks"

        android:inputType="number"

        android:layout_x="150dp"

        android:layout_y="150dp"

        android:layout_width="150dp"

        android:layout_height="40dp"/>

<Button        android:id="@+id/btnAdd"
```



```
        android:text="@string/add"

        android:layout_x="30dp"

        android:layout_y="200dp"

        android:layout_width="100dp"

        android:layout_height="40dp"

        android:onClick="onClick" />

<Button        android:id="@+id/btnDelete"

        android:text="@string/delete"

        android:layout_x="150dp"

        android:layout_y="200dp"

        android:layout_width="100dp"

        android:layout_height="40dp"

        android:onClick="onClick" />

<Button        android:id="@+id/btnModify"

        android:text="@string/modify"

        android:layout_x="30dp"

        android:layout_y="250dp"

        android:layout_width="100dp"

        android:layout_height="40dp"

        android:onClick="onClick" />

<Button        android:id="@+id/btnView"

        android:text="@string/view"

        android:layout_x="150dp"

        android:layout_y="250dp"
```

```

        android:layout_width="100dp"

        android:layout_height="40dp"

        android:onClick="onClick" />

<Button    android:id="@+id/btnViewAll"

        android:text="@string/view_all"

        android:layout_x="30dp"

        android:layout_y="300dp"

        android:layout_width="100dp"

        android:layout_height="40dp"

        android:onClick="onClick" />

<Button    android:id="@+id/btnShowInfo"

        android:text="@string/show_info"

        android:layout_x="150dp"

        android:layout_y="300dp"

        android:layout_width="100dp"

        android:layout_height="40dp"

        android:onClick="onClick" />

</AbsoluteLayout >

```

- يتم أولاً في ملف النشاط MainActivity.java التصريح عن المتحولات الموافقة لعناصر التحكم، والتصريح عن المتحول db من النوع SQLiteDatabase للتعامل مع قاعدة البيانات:

```

public class MainActivity extends Activity {

    EditText editRollno,editName,editMarks;

    Button btnAdd,btnDelete,btnModify,btnView,btnViewAll,
        btnShowInfo;

    SQLiteDatabase db;

```

- نقوم في الإجرائية onCreate بالوصول إلى عناصر التحكم (صناديق تحرير النصوص والأزرار) وإنشاء قاعدة البيانات StudentDB (أو فتحها إن كانت منشأة سابقاً).

```

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    // Initializing controls

    editRollno=(EditText)findViewById(R.id.editRollno);

    editName=(EditText)findViewById(R.id.editName);

    editMarks=(EditText)findViewById(R.id.editMarks);

    btnAdd=(Button)findViewById(R.id.btnAdd);

    btnDelete=(Button)findViewById(R.id.btnDelete);

    btnModify=(Button)findViewById(R.id.btnModify);

    btnView=(Button)findViewById(R.id.btnView);

    btnViewAll=(Button)findViewById(R.id.btnViewAll);

    btnShowInfo=(Button)findViewById(R.id.btnShowInfo);


    // Creating database and table

    db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE,
null);

    db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno
VARCHAR,name VARCHAR,marks VARCHAR);");

}

```

- تم ربط جميع أزرار الواجهة (عن طريق الخاصية onClick) مع الإجراءات التالية:

```

public void onClick(View view)

{

    // Adding a record

    if(view==btnAdd)

    {

```

```
// Checking empty fields

if(editRollno.getText().toString().trim().length()==0||
    editName.getText().toString().trim().length()==0||
    editMarks.getText().toString().trim().length()==0)
{
    showMessage("Error", "Please enter all values");
    return;
}

// Inserting record
db.execSQL("INSERT INTO student
VALUES('"+editRollno.getText()+"', '"+editName.getText()+"
        '"+editMarks.getText()+"');");

showMessage("Success", "Record added");
clearText();
}

// Deleting a record
if(view==btnDelete)
{
    // Checking empty roll number
    if(editRollno.getText().toString().trim().length()==0)
    {
        showMessage("Error", "Please enter Rollno");
        return;
    }

    // Searching roll number
```

```
Cursor c=db.rawQuery("SELECT * FROM student WHERE  
rollno='"+editRollno.getText()+"'", null);  
  
if(c.moveToFirst())  
{  
  
    // Deleting record if found  
  
    db.execSQL("DELETE FROM student WHERE  
rollno='"+editRollno.getText()+"'");  
  
    showMessage("Success", "Record Deleted");  
  
}  
  
else  
{  
  
    showMessage("Error", "Invalid Rollno");  
  
}  
  
clearText();  
  
}  
  
// Modifying a record  
if(view==btnModify)  
{  
  
    // Checking empty roll number  
  
    if(editRollno.getText().toString().trim().length()==0)  
    {  
  
        showMessage("Error", "Please enter Rollno");  
  
        return;  
  
    }  
  
    // Searching roll number  
  
    Cursor c=db.rawQuery("SELECT * FROM student WHERE
```

```
rollno='"+editRollno.getText()+"'", null);

if(c.moveToFirst())
{
    // Modifying record if found

    db.execSQL("UPDATE student SET
name='"+editName.getText()+"',marks='"+editMarks.getText()+
        "' WHERE rollno='"+editRollno.getText()+"'");

    showMessage("Success", "Record Modified");
}

else
{
    showMessage("Error", "Invalid Rollno");
}

clearText();
}

// Viewing a record
if(view==btnView)
{
    // Checking empty roll number

    if(editRollno.getText().toString().trim().length()==0)
    {
        showMessage("Error", "Please enter Rollno");

        return;
    }

    // Searching roll number
```

```
Cursor c=db.rawQuery("SELECT * FROM student WHERE  
rollno='"+editRollno.getText()+"'", null);  
  
if(c.moveToFirst())  
{  
    // Displaying record if found  
    editName.setText(c.getString(1));  
    editMarks.setText(c.getString(2));  
}  
else  
{  
    showMessage("Error", "Invalid Rollno");  
    clearText();  
}  
}  
  
// Viewing all records  
if(view==btnViewAll)  
{  
    // Retrieving all records  
    Cursor c=db.rawQuery("SELECT * FROM student", null);  
    // Checking if no records found  
    if(c.getCount()==0)  
    {  
        showMessage("Error", "No records found");  
        return;  
    }  
}
```

```
// Appending records to a string buffer

StringBuffer buffer=new StringBuffer();

while(c.moveToNext())
{
    buffer.append("Rollno: "+c.getString(0)+"\n");
    buffer.append("Name: "+c.getString(1)+"\n");
    buffer.append("Marks: "+c.getString(2)+"\n\n");
}

// Displaying all records

showMessage("Student Details", buffer.toString());
}

// Displaying info

if(view==btnShowInfo)
{
    showMessage("Student Management Application", "Developed By
BK");
}
}
```


- نبين فيما يلي الأجزاء الأساسية من الإجراءات السابقة:

إضافة بيانات طالب

```
db.execSQL("INSERT INTO student
VALUES('"+editRollno.getText()+"', '"+editName.getText()+"
        '"+editMarks.getText()+"');");
```

- لاحظ استخدام الطريقة `execSQL` لتنفيذ عبارة SQL الموافقة.

حذف طالب

```
// Searching roll number

Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+editRollno.getText()+"'", null);

if(c.moveToFirst())
{
    // Deleting record if found

    db.execSQL("DELETE FROM student WHERE
        rollno='"+editRollno.getText()+"'");

    showMessage("Success", "Record Deleted");
}

else
{
    showMessage("Error", "Invalid Rollno");
}

clearText();
```

- لاحظ أننا نقوم أولاً نقوم بإجراء استعلام عن الطالب ذو الرقم المدخل في صندوق تحرير النص الموافق.
- تُستخدم الطريقة `rawQuery` لتنفيذ استعلام يُعيد مجموعة من التسجيلات. تُعيد هذه الطريقة مؤشر (`Cursor`) على التسجيلات المعادة. تقوم الطريقة (`moveToFirst()`) بنقل المؤشر على أول تسجيله وتُعيد `true` في حال وجود تسجيلات وإلا فإنها تُعيد `false`.

- في حال عدم وجود طالب بالرقم المدخل نقوم بإظهار رسالة خطأ موافقة. وإلا نقوم بحذف تسجيله الطالب عن طريق تنفيذ عبارة SQL:

```
db.execSQL("DELETE FROM student WHERE
```

```
rollno='"+editRollno.getText()+"");
```

تعديل بيانات طالب

```
Cursor c=db.rawQuery("SELECT * FROM student WHERE
    rollno='"+editRollno.getText()+"'", null);

if(c.moveToFirst())
{
    // Modifying record if found

    db.execSQL("UPDATE student SET
name='"+editName.getText()+"',marks='"+editMarks.getText()+"'
    WHERE rollno='"+editRollno.getText()+"");

    showMessage("Success", "Record Modified");
}

else
{
    showMessage("Error", "Invalid Rollno");
}
```

- لتعديل بيانات طالب، نقوم أولاً بالاستعلام عن الطالب المدخل رقمه. وفي حال عدم وجود طالب بهذا الرقم تظهر رسالة خطأ موافقة، وإلا يتم تعديل بيانات الطالب لتصبح البيانات الظاهرة (الاسم والعلامة) هي البيانات الجديدة للطالب:

```
db.execSQL("UPDATE student SET
name='"+editName.getText()+"',marks='"+editMarks.getText()+"'
    WHERE rollno='"+editRollno.getText()+"");
```

إظهار بيانات طالب

```
Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+editRollno.getText()+"'", null);
```

```

if(c.moveToFirst())
{
    // Displaying record if found
    editName.setText(c.getString(1));
    editMarks.setText(c.getString(2));
}
else
{
    showMessage("Error", "Invalid Rollno");
    clearText();
}

```

- نقوم باستخدام الطريقة `rawQuery` مع عبارة SQL تستعلم عن الطالب وفق الرقم المدخل في صندوق تحرير رقم الطالب. تُعيد هذه الطريقة مؤشر (`Cursor`) على التسجيلات المعادة. في حال كانت نتيجة الاستعلام لا تُعيد أي تسجيلة فإن الطريقة (`moveToFirst()`) على المؤشر تُعيد `false`. أما في حال وجود تسجيلة، فلاحظ أننا نستخدم الطريقة `getString` للوصول إلى قيمة العمود المطلوب (تأخذ الطريقة معامل هو رقم العمود في الجدول).

إظهار بيانات كل الطلاب

```

// Retrieving all records
Cursor c=db.rawQuery("SELECT * FROM student", null);

// Checking if no records found
if(c.getCount()==0)
{
    showMessage("Error", "No records found");
    return;
}

```

```
// Appending records to a string buffer
StringBuffer buffer=new StringBuffer();

while(c.moveToNext())
{
    buffer.append("Rollno: "+c.getString(0)+"\n");
    buffer.append("Name: "+c.getString(1)+"\n");
    buffer.append("Marks: "+c.getString(2)+"\n\n");
}

// Displaying all records
showMessage("Student Details", buffer.toString());
```

- نقوم أولاً باستخدام الطريقة rawQuery للاستعلام عن جميع الطلاب والتي تُعيد مؤشر على التسجيلات المعادة.
- لاحظ أن الطريقة getCount() تُعيد عدد التسجيلات.
- للدوران على مجموعة التسجيلات، نستخدم الطريقة moveToNext() والتي تُعيد false عند الوصول لنهاية التسجيلات.
- نستخدم في الإجراءات السابقة الإجراءات التالية لإظهار الرسائل التنبيهية:

```
public void showMessage(String title,String message)
{
    Builder builder=new Builder(this);
    builder.setCancelable(true);
    builder.setTitle(title);
    builder.setMessage(message);
    builder.show();
}
```

- والإجراءية التالية لمسح محتوى صناديق تحرير النص:

```
public void clearText()
{
    editRollno.setText("");
    editName.setText("");
    editMarks.setText("");
    editRollno.requestFocus();
}
```