Android SQLite Example

## **Project Description**

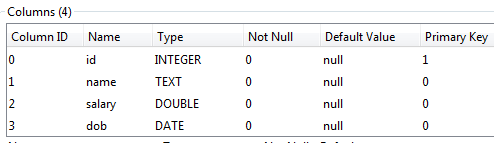
This Android SQLite tutorial explains the following,

* How to create a new database with a table (Employee table) in your application.
* Write SQLite Data Access Object (DAO) class for handling database create, read, update and delete (CRUD) operations on the table.
* Add an Action bar menu item icon to insert employee details into database.
* Retrieve employee details from database and display it in ListView.
* **Custom dialog fragment** to update employee details.
* **Communication** between **dialog fragment** and **list fragment** (**via main activity** using an interface) when employee details are updated.
* Delete employee details on list item long click event.
* Proper **back navigation of fragments** on back key press.
* How to **handle fragment orientation changes** and retain the state of the fragment.

## **Employee Table**

This Android SQLite example uses an **employee** table with the following structure.

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Key | Extra |
| id | int | Primary Key | auto\_increment |
| name | text |  |  |
| salary | double |  |  |
| dob | Date |  |  |

[](http://androidopentutorials.com/wp/wp-content/uploads/2014/04/android-sqlite-table-structure.png?1a8226)

## **Android Project**

Create a new Android project and name it as **AndroidSQLite**.

### **Resources**

#### **colors.xml**

Create a new file **res/values/colors.xml** and copy paste the following content.



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <?xml version="1.0" encoding="utf-8"?>  <resources>      <color name="transparent">#00000000</color>      <color name="list\_item\_bg">#ffffff</color>      <color name="view\_divider\_color">#BABABA</color>  </resources> |

#### **strings.xml**

Open **res/values/strings.xml** and edit to have the content as shown below.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | <?xml version="1.0" encoding="utf-8"?>  <resources>        <string name="app\_name">AndroidSQLite</string>      <string name="add\_emp">Add Employee</string>      <string name="update\_emp">Update Employee</string>      <string name="name">Employee Name</string>      <string name="dob">Employee DOB</string>      <string name="salary">Employee Salary</string>      <string name="update">Update</string>      <string name="add">Add</string>      <string name="reset">Reset</string>      <string name="cancel">Cancel</string>    </resources> |

#### **Action Bar Menu Item (main.xml)**

Open **res/menu/main.xml** and edit to have the content as shown below. This file defines the menu item in action bar for adding an employee. When this menu item is clicked, it displays EmpAddFragment which is used to add an employee record to SQLite database.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | <menu xmlns:android="http://schemas.android.com/apk/res/android" >        <item          android:id="@+id/action\_add"          android:icon="@android:drawable/ic\_menu\_add"          android:orderInCategory="100"          android:showAsAction="ifRoom|withText"          android:title="@string/add\_emp"/>    </menu> |

#### **Layout files**

##### **activity\_main.xml**

This is the main layout file. Here we define a Framelayout which is used to hold all the fragments.  
Open **res/layout/activity\_main.xml** and edit to have the content as shown below.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"      xmlns:tools="http://schemas.android.com/tools"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      tools:context=".MainActivity" >       <FrameLayout          android:id="@+id/content\_frame"          android:layout\_width="match\_parent"          android:layout\_height="match\_parent" />    </RelativeLayout> |

##### **fragment\_add\_emp.xml**

This layout file is used by **EmpAddFragment** to add employee details (name, dob, salary).



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63 | <?xml version="1.0" encoding="utf-8"?>  <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent" >        <RelativeLayout          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_margin="5dp" >            <EditText              android:id="@+id/etxt\_name"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:ems="10"              android:hint="@string/name"              android:nextFocusDown="@+id/etxt\_dob"              android:singleLine="true" />            <EditText              android:id="@+id/etxt\_dob"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:layout\_below="@+id/etxt\_name"              android:hint="@string/dob"              android:nextFocusDown="@+id/etxt\_salary"              android:singleLine="true" />            <EditText              android:id="@+id/etxt\_salary"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:layout\_below="@+id/etxt\_dob"              android:hint="@string/salary"              android:inputType="numberDecimal"              android:singleLine="true" />            <LinearLayout              android:id="@+id/layout\_submit"              android:layout\_width="fill\_parent"              android:layout\_height="wrap\_content"              android:layout\_below="@+id/etxt\_salary"              android:layout\_margin="5dp"              android:orientation="horizontal"              android:weightSum="2" >                <Button                  android:id="@+id/button\_add"                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_weight="1"                  android:text="@string/add" />                <Button                  android:id="@+id/button\_reset"                  android:layout\_width="0dp"                  android:layout\_height="wrap\_content"                  android:layout\_weight="1"                  android:text="@string/reset" />          </LinearLayout>      </RelativeLayout>    </ScrollView> |

##### **fragment\_emp\_list.xml**

This layout file defines a ListView which is used by **EmpListFragment** to display list of employees.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"      xmlns:tools="http://schemas.android.com/tools"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:background="#EDEDED" >        <ListView          android:id="@+id/list\_emp"          android:layout\_width="match\_parent"          android:layout\_height="match\_parent"          android:divider="@color/transparent"          android:dividerHeight="10dp"          android:drawSelectorOnTop="true"          android:footerDividersEnabled="false"          android:padding="10dp"          android:scrollbarStyle="outsideOverlay" />    </RelativeLayout> |

##### **list\_item.xml**

This file defines custom layout for ListView item (employee row) which is used by **EmpListAdapter**.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47 | <?xml version="1.0" encoding="utf-8"?>  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="fill\_parent"      android:layout\_height="wrap\_content"      android:background="@color/list\_item\_bg"      android:descendantFocusability="blocksDescendants" >        <RelativeLayout          android:id="@+id/layout\_item"          android:layout\_width="fill\_parent"          android:layout\_height="wrap\_content" >            <TextView              android:id="@+id/txt\_emp\_id"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:padding="6dp" />            <TextView              android:id="@+id/txt\_emp\_name"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:layout\_toRightOf="@+id/txt\_emp\_id"              android:padding="6dp" />            <TextView              android:id="@+id/txt\_emp\_dob"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:layout\_below="@+id/txt\_emp\_name"              android:padding="6dp" />            <TextView              android:id="@+id/txt\_emp\_salary"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:layout\_below="@+id/txt\_emp\_dob"              android:padding="6dp" />      </RelativeLayout>        <View          android:layout\_width="match\_parent"          android:layout\_height="1dp"          android:layout\_below="@+id/layout\_item"          android:background="@color/view\_divider\_color" />    </RelativeLayout> |

## **Create a SQLite database in Android**

### **DataBaseHelper class**

In src folder, create a new class **DataBaseHelper** in the package **com.androidopentutorials.sqlite.db**. This class extends **SQLiteOpenHelper** which manages database creation and version management.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49 | package com.androidopentutorials.sqlite.db;    import android.content.Context;  import android.database.sqlite.SQLiteDatabase;  import android.database.sqlite.SQLiteOpenHelper;    public class DataBaseHelper extends SQLiteOpenHelper {        private static final String DATABASE\_NAME = "employeedb";      private static final int DATABASE\_VERSION = 1;        public static final String EMPLOYEE\_TABLE = "employee";        public static final String ID\_COLUMN = "id";      public static final String NAME\_COLUMN = "name";      public static final String EMPLOYEE\_DOB = "dob";      public static final String EMPLOYEE\_SALARY = "salary";        public static final String CREATE\_EMPLOYEE\_TABLE = "CREATE TABLE "              + EMPLOYEE\_TABLE + "(" + ID\_COLUMN + " INTEGER PRIMARY KEY, "              + NAME\_COLUMN + " TEXT, " + EMPLOYEE\_SALARY + " DOUBLE, "              + EMPLOYEE\_DOB + " DATE" + ")";        private static DataBaseHelper instance;        public static synchronized DataBaseHelper getHelper(Context context) {          if (instance == null)              instance = new DataBaseHelper(context);          return instance;      }        private DataBaseHelper(Context context) {          super(context, DATABASE\_NAME, null, DATABASE\_VERSION);      }        @Override      public void onOpen(SQLiteDatabase db) {          super.onOpen(db);      }        @Override      public void onCreate(SQLiteDatabase db) {          db.execSQL(CREATE\_EMPLOYEE\_TABLE);      }        @Override      public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {      }  } |

### **Employee DB DAO class**

Create a new class **EmployeeDBDAO** in the package **com.androidopentutorials.sqlite.db**. To write to and read from the database we need a SQLiteDatabase object (that represents the database) which can be obtained by calling getWritableDatabase() on the database helper.

Then we will create an EmployeeDAO class which will extend this class and using SQLiteDatabase object provides methods for SQLite CRUD (Create, Read, Update, Delete) operations.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | package com.androidopentutorials.sqlite.db;    import android.content.Context;  import android.database.SQLException;  import android.database.sqlite.SQLiteDatabase;    public class EmployeeDBDAO {        protected SQLiteDatabase database;      private DataBaseHelper dbHelper;      private Context mContext;        public EmployeeDBDAO(Context context) {          this.mContext = context;          dbHelper = DataBaseHelper.getHelper(mContext);          open();        }        public void open() throws SQLException {          if(dbHelper == null)              dbHelper = DataBaseHelper.getHelper(mContext);          database = dbHelper.getWritableDatabase();      }        /\*public void close() {          dbHelper.close();          database = null;      }\*/  } |

## **Insert a new record in SQLite database**

### **Employee**

In src folder, create a new class **Employee** in the package **com.androidopentutorials.sqlite.to**.

* This class represents an Employee stored in a database.
* This class implements Parcelable interface because we need to bundle the employee object as argument when showing custom dialog fragment to update employee details.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109 | package com.androidopentutorials.sqlite.to;    import java.util.Date;  import android.os.Parcel;  import android.os.Parcelable;    public class Employee implements Parcelable {        private int id;      private String name;      private Date dateOfBirth;      private double salary;        public Employee() {          super();      }        private Employee(Parcel in) {          super();          this.id = in.readInt();          this.name = in.readString();          this.dateOfBirth = new Date(in.readLong());          this.salary = in.readDouble();      }        public int getId() {          return id;      }        public void setId(int id) {          this.id = id;      }        public String getName() {          return name;      }        public void setName(String name) {          this.name = name;      }        public Date getDateOfBirth() {          return dateOfBirth;      }        public void setDateOfBirth(Date dateOfBirth) {          this.dateOfBirth = dateOfBirth;      }        public double getSalary() {          return salary;      }        public void setSalary(double salary) {          this.salary = salary;      }        @Override      public String toString() {          return "Employee [id=" + id + ", name=" + name + ", dateOfBirth="                  + dateOfBirth + ", salary=" + salary + "]";      }        @Override      public int hashCode() {          final int prime = 31;          int result = 1;          result = prime \* result + id;          return result;      }        @Override      public boolean equals(Object obj) {          if (this == obj)              return true;          if (obj == null)              return false;          if (getClass() != obj.getClass())              return false;          Employee other = (Employee) obj;          if (id != other.id)              return false;          return true;      }        @Override      public int describeContents() {          return 0;      }        @Override      public void writeToParcel(Parcel parcel, int flags) {          parcel.writeInt(getId());          parcel.writeString(getName());          parcel.writeLong(getDateOfBirth().getTime());          parcel.writeDouble(getSalary());      }        public static final Parcelable.Creator<Employee> CREATOR = new Parcelable.Creator<Employee>() {          public Employee createFromParcel(Parcel in) {              return new Employee(in);          }            public Employee[] newArray(int size) {              return new Employee[size];          }      };    } |

### **Save Employee**

In src folder, create a new class **EmployeeDAO** in the package **com.androidopentutorials.sqlite.db**. This class extends EmployeeDBDAO and implements database operations such as retrieve, save, update, and delete employee object.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | package com.androidopentutorials.sqlite.db;    import android.content.ContentValues;  import android.content.Context;  import java.text.SimpleDateFormat;  import java.util.Locale;    import com.androidopentutorials.sqlite.to.Employee;    public class EmployeeDAO extends EmployeeDBDAO {            private static final SimpleDateFormat formatter = new SimpleDateFormat(              "yyyy-MM-dd", Locale.ENGLISH);        public EmployeeDAO(Context context) {          super(context);      }            ...      public long save(Employee employee) {          ContentValues values = new ContentValues();          values.put(DataBaseHelper.NAME\_COLUMN, employee.getName());          values.put(DataBaseHelper.EMPLOYEE\_DOB, formatter.format(employee.getDateOfBirth()));          values.put(DataBaseHelper.EMPLOYEE\_SALARY, employee.getSalary());            return database.insert(DataBaseHelper.EMPLOYEE\_TABLE, null, values);      }            ....  } |

Here we are formatting ‘date of birth’ as ‘yyyy-MM-dd’ by using SimpleDateFormatter to be compatible with SQLite date type. To insert a new row in SQLite, we use **ContentValues** which define key-value pairs where key represents table column name and the value represents the data for the column for a particular record.

### **Add Employee Fragment**

In src folder, create a new class **EmpAddFragment** in the package **com.androidopentutorials.sqlite.fragment**.

* This fragment displays a form to enter employee details such as name, date of birth and salary. DatePickerDialog opens when date of birth EditText is clicked.
* We execute a background task (AsyncTask) to save the employee record by calling save() method of EmployeeDAO class.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175 | package com.androidopentutorials.sqlite.fragment;    import java.lang.ref.WeakReference;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  import java.util.Locale;    import android.app.Activity;  import android.app.DatePickerDialog;  import android.app.DatePickerDialog.OnDateSetListener;  import android.os.AsyncTask;  import android.os.Bundle;  import android.support.v4.app.Fragment;  import android.text.InputType;  import android.view.LayoutInflater;  import android.view.View;  import android.view.View.OnClickListener;  import android.view.ViewGroup;  import android.widget.Button;  import android.widget.DatePicker;  import android.widget.EditText;  import android.widget.Toast;    import com.androidopentutorials.sqlite.R;  import com.androidopentutorials.sqlite.db.EmployeeDAO;  import com.androidopentutorials.sqlite.to.Employee;    public class EmpAddFragment extends Fragment implements OnClickListener {        // UI references      private EditText empNameEtxt;      private EditText empSalaryEtxt;      private EditText empDobEtxt;      private Button addButton;      private Button resetButton;        private static final SimpleDateFormat formatter = new SimpleDateFormat(              "yyyy-MM-dd", Locale.ENGLISH);        DatePickerDialog datePickerDialog;      Calendar dateCalendar;        Employee employee = null;      private EmployeeDAO employeeDAO;      private AddEmpTask task;        public static final String ARG\_ITEM\_ID = "emp\_add\_fragment";        @Override      public void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          employeeDAO = new EmployeeDAO(getActivity());      }        @Override      public View onCreateView(LayoutInflater inflater, ViewGroup container,              Bundle savedInstanceState) {          View rootView = inflater.inflate(R.layout.fragment\_add\_emp, container,                  false);            findViewsById(rootView);            setListeners();            //For orientation change.          if (savedInstanceState != null) {              dateCalendar = Calendar.getInstance();              if (savedInstanceState.getLong("dateCalendar") != 0)                  dateCalendar.setTime(new Date(savedInstanceState                          .getLong("dateCalendar")));          }            return rootView;      }        private void setListeners() {          empDobEtxt.setOnClickListener(this);          Calendar newCalendar = Calendar.getInstance();          datePickerDialog = new DatePickerDialog(getActivity(),                  new OnDateSetListener() {                        public void onDateSet(DatePicker view, int year,                              int monthOfYear, int dayOfMonth) {                          dateCalendar = Calendar.getInstance();                          dateCalendar.set(year, monthOfYear, dayOfMonth);                          empDobEtxt.setText(formatter.format(dateCalendar                                  .getTime()));                      }                    }, newCalendar.get(Calendar.YEAR),                  newCalendar.get(Calendar.MONTH),                  newCalendar.get(Calendar.DAY\_OF\_MONTH));            addButton.setOnClickListener(this);          resetButton.setOnClickListener(this);      }        protected void resetAllFields() {          empNameEtxt.setText("");          empSalaryEtxt.setText("");          empDobEtxt.setText("");      }        private void setEmployee() {          employee = new Employee();          employee.setName(empNameEtxt.getText().toString());          employee.setSalary(Double.parseDouble(empSalaryEtxt.getText()                  .toString()));          if (dateCalendar != null)              employee.setDateOfBirth(dateCalendar.getTime());      }        @Override      public void onResume() {          getActivity().setTitle(R.string.add\_emp);          getActivity().getActionBar().setTitle(R.string.add\_emp);          super.onResume();      }        @Override      public void onSaveInstanceState(Bundle outState) {          if (dateCalendar != null)              outState.putLong("dateCalendar", dateCalendar.getTime().getTime());      }        private void findViewsById(View rootView) {          empNameEtxt = (EditText) rootView.findViewById(R.id.etxt\_name);          empSalaryEtxt = (EditText) rootView.findViewById(R.id.etxt\_salary);          empDobEtxt = (EditText) rootView.findViewById(R.id.etxt\_dob);          empDobEtxt.setInputType(InputType.TYPE\_NULL);            addButton = (Button) rootView.findViewById(R.id.button\_add);          resetButton = (Button) rootView.findViewById(R.id.button\_reset);      }        @Override      public void onClick(View view) {          if (view == empDobEtxt) {              datePickerDialog.show();          } else if (view == addButton) {              setEmployee();                task = new AddEmpTask(getActivity());              task.execute((Void) null);          } else if (view == resetButton) {              resetAllFields();          }      }        public class AddEmpTask extends AsyncTask<Void, Void, Long> {            private final WeakReference<Activity> activityWeakRef;            public AddEmpTask(Activity context) {              this.activityWeakRef = new WeakReference<Activity>(context);          }            @Override          protected Long doInBackground(Void... arg0) {              long result = employeeDAO.save(employee);              return result;          }            @Override          protected void onPostExecute(Long result) {              if (activityWeakRef.get() != null                      && !activityWeakRef.get().isFinishing()) {                  if (result != -1)                      Toast.makeText(activityWeakRef.get(), "Employee Saved",                              Toast.LENGTH\_LONG).show();              }          }      }  } |

## **Fetch records from SQLite database**

### **EmployeeDAO class**

In the above mentioned EmployeeDAO class, add the below getEmployees() method to get all records from database. This class uses query method which accepts the following parameters and returns a Cursor over the result set,

*public Cursor query (String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy)*



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | package com.androidopentutorials.sqlite.db;    import java.util.ArrayList;  import java.util.Date;  import android.content.Context;  import android.database.Cursor;    import com.androidopentutorials.sqlite.to.Employee;    public class EmployeeDAO extends EmployeeDBDAO {            ...      public ArrayList<Employee> getEmployees() {          ArrayList<Employee> employees = new ArrayList<Employee>();            Cursor cursor = database.query(DataBaseHelper.EMPLOYEE\_TABLE,                  new String[] { DataBaseHelper.ID\_COLUMN,                          DataBaseHelper.NAME\_COLUMN,                          DataBaseHelper.EMPLOYEE\_DOB,                          DataBaseHelper.EMPLOYEE\_SALARY }, null, null, null,                  null, null);            while (cursor.moveToNext()) {              Employee employee = new Employee();              employee.setId(cursor.getInt(0));              employee.setName(cursor.getString(1));              try {                  employee.setDateOfBirth(formatter.parse(cursor.getString(2)));              } catch (ParseException e) {                  employee.setDateOfBirth(null);              }              employee.setSalary(cursor.getDouble(3));                employees.add(employee);          }          return employees;      }          ...  } |

### **Employee List Adapter**

In src folder, create a new class **EmpListAdapter** in the package **com.androidopentutorials.sqlite.adapter**. This is the custom list adapter class which displays employee details – id, name, date of birth and salary.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 | package com.androidopentutorials.sqlite.adapter;    import java.text.SimpleDateFormat;  import java.util.List;  import java.util.Locale;    import android.app.Activity;  import android.content.Context;  import android.view.LayoutInflater;  import android.view.View;  import android.view.ViewGroup;  import android.widget.ArrayAdapter;  import android.widget.TextView;    import com.androidopentutorials.sqlite.R;  import com.androidopentutorials.sqlite.to.Employee;    public class EmpListAdapter extends ArrayAdapter<Employee> {        private Context context;      List<Employee> employees;        private static final SimpleDateFormat formatter = new SimpleDateFormat(              "yyyy-MM-dd", Locale.ENGLISH);        public EmpListAdapter(Context context, List<Employee> employees) {          super(context, R.layout.list\_item, employees);          this.context = context;          this.employees = employees;      }        private class ViewHolder {          TextView empIdTxt;          TextView empNameTxt;          TextView empDobTxt;          TextView empSalaryTxt;      }        @Override      public int getCount() {          return employees.size();      }        @Override      public Employee getItem(int position) {          return employees.get(position);      }        @Override      public long getItemId(int position) {          return 0;      }        @Override      public View getView(int position, View convertView, ViewGroup parent) {          ViewHolder holder = null;          if (convertView == null) {              LayoutInflater inflater = (LayoutInflater) context                      .getSystemService(Activity.LAYOUT\_INFLATER\_SERVICE);              convertView = inflater.inflate(R.layout.list\_item, null);              holder = new ViewHolder();                holder.empIdTxt = (TextView) convertView                      .findViewById(R.id.txt\_emp\_id);              holder.empNameTxt = (TextView) convertView                      .findViewById(R.id.txt\_emp\_name);              holder.empDobTxt = (TextView) convertView                      .findViewById(R.id.txt\_emp\_dob);              holder.empSalaryTxt = (TextView) convertView                      .findViewById(R.id.txt\_emp\_salary);                convertView.setTag(holder);          } else {              holder = (ViewHolder) convertView.getTag();          }          Employee employee = (Employee) getItem(position);          holder.empIdTxt.setText(employee.getId() + "");          holder.empNameTxt.setText(employee.getName());          holder.empSalaryTxt.setText(employee.getSalary() + "");            holder.empDobTxt.setText(formatter.format(employee.getDateOfBirth()));            return convertView;      }        @Override      public void add(Employee employee) {          employees.add(employee);          notifyDataSetChanged();          super.add(employee);      }        @Override      public void remove(Employee employee) {          employees.remove(employee);          notifyDataSetChanged();          super.remove(employee);      }  } |

The **add()** and **remove()** method adds and removes item from ListView and notifies the Listview adapter.

### **Display Database Records in ListView – Employee List Fragment**

In src folder, create a new class **EmpListFragment** in the package **com.androidopentutorials.sqlite.fragment**.

* This fragment class executes a background task (AsyncTask) where we call getEmployees() method in EmployeeDAO and display list of employees in a ListView.
* When a list item is clicked we display a custom dialog fragment to update employee details.
* On list item long click event we delete a particular employee record from the database and remove it from the list adapter.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144 | package com.androidopentutorials.sqlite.fragment;    import java.lang.ref.WeakReference;  import java.util.ArrayList;    import android.app.Activity;  import android.os.AsyncTask;  import android.os.Bundle;  import android.support.v4.app.Fragment;  import android.util.Log;  import android.view.LayoutInflater;  import android.view.View;  import android.view.ViewGroup;  import android.widget.AdapterView;  import android.widget.AdapterView.OnItemClickListener;  import android.widget.AdapterView.OnItemLongClickListener;  import android.widget.ListView;  import android.widget.Toast;    import com.androidopentutorials.sqlite.R;  import com.androidopentutorials.sqlite.adapter.EmpListAdapter;  import com.androidopentutorials.sqlite.db.EmployeeDAO;  import com.androidopentutorials.sqlite.to.Employee;    public class EmpListFragment extends Fragment implements OnItemClickListener,          OnItemLongClickListener {        public static final String ARG\_ITEM\_ID = "employee\_list";        Activity activity;      ListView employeeListView;      ArrayList<Employee> employees;        EmpListAdapter employeeListAdapter;      EmployeeDAO employeeDAO;        private GetEmpTask task;        @Override      public void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          activity = getActivity();          employeeDAO = new EmployeeDAO(activity);      }        @Override      public View onCreateView(LayoutInflater inflater, ViewGroup container,              Bundle savedInstanceState) {          View view = inflater.inflate(R.layout.fragment\_emp\_list, container,                  false);          findViewsById(view);            task = new GetEmpTask(activity);          task.execute((Void) null);            employeeListView.setOnItemClickListener(this);          employeeListView.setOnItemLongClickListener(this);          // Employee e = employeeDAO.getEmployee(1);          // Log.d("employee e", e.toString());          return view;      }        private void findViewsById(View view) {          employeeListView = (ListView) view.findViewById(R.id.list\_emp);      }        @Override      public void onResume() {          getActivity().setTitle(R.string.app\_name);          getActivity().getActionBar().setTitle(R.string.app\_name);          super.onResume();      }        @Override      public void onItemClick(AdapterView<?> list, View arg1, int position,              long arg3) {          Employee employee = (Employee) list.getItemAtPosition(position);            if (employee != null) {              Bundle arguments = new Bundle();              arguments.putParcelable("selectedEmployee", employee);              CustomEmpDialogFragment customEmpDialogFragment = new CustomEmpDialogFragment();              customEmpDialogFragment.setArguments(arguments);              customEmpDialogFragment.show(getFragmentManager(),                      CustomEmpDialogFragment.ARG\_ITEM\_ID);          }      }        @Override      public boolean onItemLongClick(AdapterView<?> parent, View view,              int position, long arg3) {          Employee employee = (Employee) parent.getItemAtPosition(position);            // Use AsyncTask to delete from database          employeeDAO.delete(employee);          employeeListAdapter.remove(employee);          return true;      }        public class GetEmpTask extends AsyncTask<Void, Void, ArrayList<Employee>> {            private final WeakReference<Activity> activityWeakRef;            public GetEmpTask(Activity context) {              this.activityWeakRef = new WeakReference<Activity>(context);          }            @Override          protected ArrayList<Employee> doInBackground(Void... arg0) {              ArrayList<Employee> employeeList = employeeDAO.getEmployees();              return employeeList;          }            @Override          protected void onPostExecute(ArrayList<Employee> empList) {              if (activityWeakRef.get() != null                      && !activityWeakRef.get().isFinishing()) {                  Log.d("employees", empList.toString());                  employees = empList;                  if (empList != null) {                      if (empList.size() != 0) {                          employeeListAdapter = new EmpListAdapter(activity,                                  empList);                          employeeListView.setAdapter(employeeListAdapter);                      } else {                          Toast.makeText(activity, "No Employee Records",                                  Toast.LENGTH\_LONG).show();                      }                  }                }          }      }        /\*       \* This method is invoked from MainActivity onFinishDialog() method. It is       \* called from CustomEmpDialogFragment when an employee record is updated.       \* This is used for communicating between fragments.       \*/      public void updateView() {          task = new GetEmpTask(activity);          task.execute((Void) null);      }  } |

## **Update a row in SQLite database**

### **EmployeeDAO class**

In the above mentioned EmployeeDAO class, add the following method to update a particular record from database.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | package com.androidopentutorials.sqlite.db;    import android.content.ContentValues;  import android.content.Context;  import android.util.Log;    import com.androidopentutorials.sqlite.to.Employee;    public class EmployeeDAO extends EmployeeDBDAO {        private static final String WHERE\_ID\_EQUALS = DataBaseHelper.ID\_COLUMN              + " =?";          private static final SimpleDateFormat formatter = new SimpleDateFormat(              "yyyy-MM-dd", Locale.ENGLISH);          ...      public long update(Employee employee) {          ContentValues values = new ContentValues();          values.put(DataBaseHelper.NAME\_COLUMN, employee.getName());          values.put(DataBaseHelper.EMPLOYEE\_DOB, formatter.format(employee.getDateOfBirth()));          values.put(DataBaseHelper.EMPLOYEE\_SALARY, employee.getSalary());            long result = database.update(DataBaseHelper.EMPLOYEE\_TABLE, values,                  WHERE\_ID\_EQUALS,                  new String[] { String.valueOf(employee.getId()) });          Log.d("Update Result:", "=" + result);          return result;        }          ...  } |

### **CustomEmpDialogFragment class**

In src folder, create a new class **CustomEmpDialogFragment** in the package **com.androidopentutorials.sqlite.fragment**.

* This class receives an employee object in bundle as an argument and displays a dialog with employee details (name, salary, dob) and updates employee details.
* If the update is successful, it calls onFinishDialog() on MainActivity to notify the ListView (EmpListFragment). **All communication between fragments should go via their activity class.**



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118 | package com.androidopentutorials.sqlite.fragment;    import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Locale;    import android.app.AlertDialog;  import android.app.Dialog;  import android.content.DialogInterface;  import android.os.Bundle;  import android.support.v4.app.DialogFragment;  import android.view.LayoutInflater;  import android.view.View;  import android.widget.EditText;  import android.widget.LinearLayout;  import android.widget.Toast;    import com.androidopentutorials.sqlite.MainActivity;  import com.androidopentutorials.sqlite.R;  import com.androidopentutorials.sqlite.db.EmployeeDAO;  import com.androidopentutorials.sqlite.to.Employee;    public class CustomEmpDialogFragment extends DialogFragment {        // UI references      private EditText empNameEtxt;      private EditText empSalaryEtxt;      private EditText empDobEtxt;      private LinearLayout submitLayout;        private Employee employee;        private static final SimpleDateFormat formatter = new SimpleDateFormat(              "yyyy-MM-dd", Locale.ENGLISH);        EmployeeDAO employeeDAO;        public static final String ARG\_ITEM\_ID = "emp\_dialog\_fragment";        public interface EmpDialogFragmentListener {          void onFinishDialog();      }        public CustomEmpDialogFragment() {        }        @Override      public Dialog onCreateDialog(Bundle savedInstanceState) {          employeeDAO = new EmployeeDAO(getActivity());            Bundle bundle = this.getArguments();          employee = bundle.getParcelable("selectedEmployee");            AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());          LayoutInflater inflater = getActivity().getLayoutInflater();            View customDialogView = inflater.inflate(R.layout.fragment\_add\_emp,                  null);          builder.setView(customDialogView);            empNameEtxt = (EditText) customDialogView.findViewById(R.id.etxt\_name);          empSalaryEtxt = (EditText) customDialogView                  .findViewById(R.id.etxt\_salary);          empDobEtxt = (EditText) customDialogView.findViewById(R.id.etxt\_dob);          submitLayout = (LinearLayout) customDialogView                  .findViewById(R.id.layout\_submit);          submitLayout.setVisibility(View.GONE);            setValue();            builder.setTitle(R.string.update\_emp);          builder.setCancelable(false);          builder.setPositiveButton(R.string.update,                  new DialogInterface.OnClickListener() {                      public void onClick(DialogInterface dialog, int id) {                          try {                              employee.setDateOfBirth(formatter.parse(empDobEtxt.getText().toString()));                          } catch (ParseException e) {                              Toast.makeText(getActivity(),                                      "Invalid date format!",                                      Toast.LENGTH\_SHORT).show();                              return;                          }                          employee.setName(empNameEtxt.getText().toString());                          employee.setSalary(Double.parseDouble(empSalaryEtxt                                  .getText().toString()));                          long result = employeeDAO.update(employee);                          if (result > 0) {                              MainActivity activity = (MainActivity) getActivity();                              activity.onFinishDialog();                          } else {                              Toast.makeText(getActivity(),                                      "Unable to update employee",                                      Toast.LENGTH\_SHORT).show();                          }                      }                  });          builder.setNegativeButton(R.string.cancel,                  new DialogInterface.OnClickListener() {                      public void onClick(DialogInterface dialog, int id) {                          dialog.cancel();                      }                  });            AlertDialog alertDialog = builder.create();            return alertDialog;      }        private void setValue() {          if (employee != null) {              empNameEtxt.setText(employee.getName());              empSalaryEtxt.setText(employee.getSalary() + "");              empDobEtxt.setText(formatter.format(employee.getDateOfBirth()));          }      }  } |

## **Delete a row from SQLite database**

### **EmployeeDAO class**

In the above mentioned EmployeeDAO class, add the following method to delete a particular record from database.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | package com.androidopentutorials.sqlite.db;    import android.content.Context;  import com.androidopentutorials.sqlite.to.Employee;    public class EmployeeDAO extends EmployeeDBDAO {          ...      public int delete(Employee employee) {          return database.delete(DataBaseHelper.EMPLOYEE\_TABLE, WHERE\_ID\_EQUALS,                  new String[] { employee.getId() + "" });      }          ...  } |

**EmpListFragment calls this delete() method on list item long click event.**

## **Complete EmployeeDAO class**



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132 | package com.androidopentutorials.sqlite.db;    import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.ArrayList;  import java.util.Locale;    import android.content.ContentValues;  import android.content.Context;  import android.database.Cursor;  import android.util.Log;    import com.androidopentutorials.sqlite.to.Employee;    public class EmployeeDAO extends EmployeeDBDAO {        private static final String WHERE\_ID\_EQUALS = DataBaseHelper.ID\_COLUMN              + " =?";      private static final SimpleDateFormat formatter = new SimpleDateFormat(              "yyyy-MM-dd", Locale.ENGLISH);        public EmployeeDAO(Context context) {          super(context);      }        public long save(Employee employee) {          ContentValues values = new ContentValues();          values.put(DataBaseHelper.NAME\_COLUMN, employee.getName());          Log.d("dob", employee.getDateOfBirth().getTime() + "");          values.put(DataBaseHelper.EMPLOYEE\_DOB, formatter.format(employee.getDateOfBirth()));          values.put(DataBaseHelper.EMPLOYEE\_SALARY, employee.getSalary());            return database.insert(DataBaseHelper.EMPLOYEE\_TABLE, null, values);      }        public long update(Employee employee) {          ContentValues values = new ContentValues();          values.put(DataBaseHelper.NAME\_COLUMN, employee.getName());          values.put(DataBaseHelper.EMPLOYEE\_DOB, formatter.format(employee.getDateOfBirth()));          values.put(DataBaseHelper.EMPLOYEE\_SALARY, employee.getSalary());            long result = database.update(DataBaseHelper.EMPLOYEE\_TABLE, values,                  WHERE\_ID\_EQUALS,                  new String[] { String.valueOf(employee.getId()) });          Log.d("Update Result:", "=" + result);          return result;        }        public int delete(Employee employee) {          return database.delete(DataBaseHelper.EMPLOYEE\_TABLE, WHERE\_ID\_EQUALS,                  new String[] { employee.getId() + "" });      }        //USING query() method      public ArrayList<Employee> getEmployees() {          ArrayList<Employee> employees = new ArrayList<Employee>();            Cursor cursor = database.query(DataBaseHelper.EMPLOYEE\_TABLE,                  new String[] { DataBaseHelper.ID\_COLUMN,                          DataBaseHelper.NAME\_COLUMN,                          DataBaseHelper.EMPLOYEE\_DOB,                          DataBaseHelper.EMPLOYEE\_SALARY }, null, null, null,                  null, null);            while (cursor.moveToNext()) {              Employee employee = new Employee();              employee.setId(cursor.getInt(0));              employee.setName(cursor.getString(1));              try {                  employee.setDateOfBirth(formatter.parse(cursor.getString(2)));              } catch (ParseException e) {                  employee.setDateOfBirth(null);              }              employee.setSalary(cursor.getDouble(3));                employees.add(employee);          }          return employees;      }        //USING rawQuery() method      /\*public ArrayList<Employee> getEmployees() {          ArrayList<Employee> employees = new ArrayList<Employee>();            String sql = "SELECT " + DataBaseHelper.ID\_COLUMN + ","                  + DataBaseHelper.NAME\_COLUMN + ","                  + DataBaseHelper.EMPLOYEE\_DOB + ","                  + DataBaseHelper.EMPLOYEE\_SALARY + " FROM "                  + DataBaseHelper.EMPLOYEE\_TABLE;            Cursor cursor = database.rawQuery(sql, null);            while (cursor.moveToNext()) {              Employee employee = new Employee();              employee.setId(cursor.getInt(0));              employee.setName(cursor.getString(1));              try {                  employee.setDateOfBirth(formatter.parse(cursor.getString(2)));              } catch (ParseException e) {                  employee.setDateOfBirth(null);              }              employee.setSalary(cursor.getDouble(3));                employees.add(employee);          }          return employees;      }\*/        //Retrieves a single employee record with the given id      public Employee getEmployee(long id) {          Employee employee = null;            String sql = "SELECT \* FROM " + DataBaseHelper.EMPLOYEE\_TABLE                  + " WHERE " + DataBaseHelper.ID\_COLUMN + " = ?";            Cursor cursor = database.rawQuery(sql, new String[] { id + "" });            if (cursor.moveToNext()) {              employee = new Employee();              employee.setId(cursor.getInt(0));              employee.setName(cursor.getString(1));              try {                  employee.setDateOfBirth(formatter.parse(cursor.getString(2)));              } catch (ParseException e) {                  employee.setDateOfBirth(null);              }              employee.setSalary(cursor.getDouble(3));          }          return employee;      }  } |

## **MainActivity class**

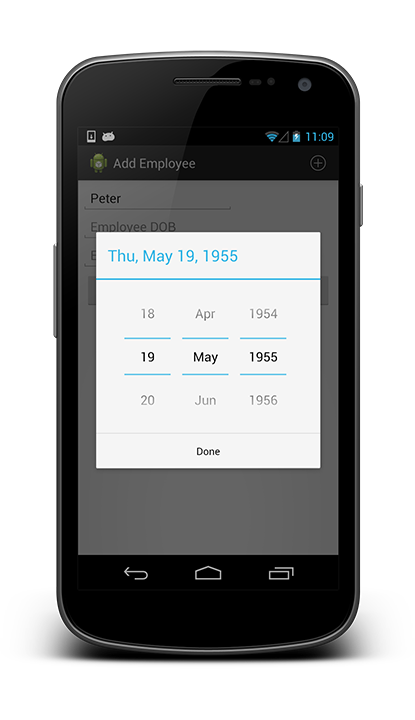
* This is the main activity class.
* When the app starts, we add the EmpListFragment.
* This activity **handles proper back navigation of fragments** on back key pressed by overriding **onBackPressed()**. It shows an alert dialog on quit.
* It also handles **fragment orientation changes** and retain the fragment state by overriding **onSaveInstanceState()**.



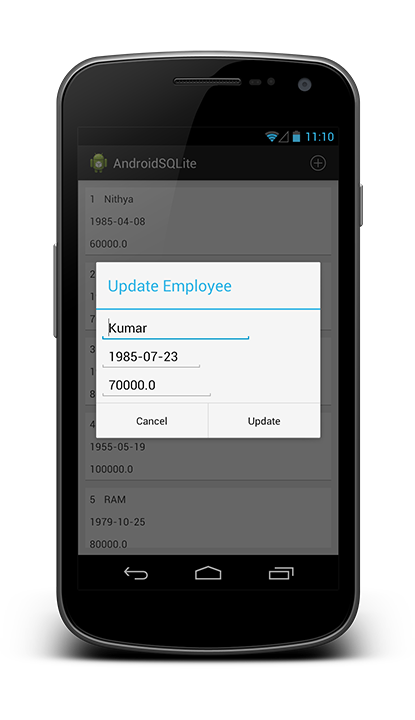
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160 | package com.androidopentutorials.sqlite;    import android.app.AlertDialog;  import android.content.DialogInterface;  import android.os.Bundle;  import android.support.v4.app.Fragment;  import android.support.v4.app.FragmentActivity;  import android.support.v4.app.FragmentManager;  import android.support.v4.app.FragmentTransaction;  import android.view.Menu;  import android.view.MenuItem;    import com.androidopentutorials.sqlite.fragment.CustomEmpDialogFragment.EmpDialogFragmentListener;  import com.androidopentutorials.sqlite.fragment.EmpAddFragment;  import com.androidopentutorials.sqlite.fragment.EmpListFragment;    public class MainActivity extends FragmentActivity implements          EmpDialogFragmentListener {        private Fragment contentFragment;      private EmpListFragment employeeListFragment;      private EmpAddFragment employeeAddFragment;        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);            FragmentManager fragmentManager = getSupportFragmentManager();            /\*           \* This is called when orientation is changed.           \*/          if (savedInstanceState != null) {              if (savedInstanceState.containsKey("content")) {                  String content = savedInstanceState.getString("content");                  if (content.equals(EmpAddFragment.ARG\_ITEM\_ID)) {                      if (fragmentManager                              .findFragmentByTag(EmpAddFragment.ARG\_ITEM\_ID) != null) {                          setFragmentTitle(R.string.add\_emp);                          contentFragment = fragmentManager                                  .findFragmentByTag(EmpAddFragment.ARG\_ITEM\_ID);                      }                  }              }              if (fragmentManager.findFragmentByTag(EmpListFragment.ARG\_ITEM\_ID) != null) {                  employeeListFragment = (EmpListFragment) fragmentManager                          .findFragmentByTag(EmpListFragment.ARG\_ITEM\_ID);                  contentFragment = employeeListFragment;              }          } else {              employeeListFragment = new EmpListFragment();              setFragmentTitle(R.string.app\_name);              switchContent(employeeListFragment, EmpListFragment.ARG\_ITEM\_ID);          }      }        @Override      protected void onSaveInstanceState(Bundle outState) {          if (contentFragment instanceof EmpAddFragment) {              outState.putString("content", EmpAddFragment.ARG\_ITEM\_ID);          } else {              outState.putString("content", EmpListFragment.ARG\_ITEM\_ID);          }          super.onSaveInstanceState(outState);      }        @Override      public boolean onCreateOptionsMenu(Menu menu) {          getMenuInflater().inflate(R.menu.main, menu);          return true;      }        @Override      public boolean onOptionsItemSelected(MenuItem item) {          switch (item.getItemId()) {          case R.id.action\_add:              setFragmentTitle(R.string.add\_emp);              employeeAddFragment = new EmpAddFragment();              switchContent(employeeAddFragment, EmpAddFragment.ARG\_ITEM\_ID);                return true;          }          return super.onOptionsItemSelected(item);      }        /\*       \* We consider EmpListFragment as the home fragment and it is not added to       \* the back stack.       \*/      public void switchContent(Fragment fragment, String tag) {          FragmentManager fragmentManager = getSupportFragmentManager();          while (fragmentManager.popBackStackImmediate())              ;            if (fragment != null) {              FragmentTransaction transaction = fragmentManager                      .beginTransaction();              transaction.replace(R.id.content\_frame, fragment, tag);              // Only EmpAddFragment is added to the back stack.              if (!(fragment instanceof EmpListFragment)) {                  transaction.addToBackStack(tag);              }              transaction.commit();              contentFragment = fragment;          }      }        protected void setFragmentTitle(int resourseId) {          setTitle(resourseId);          getActionBar().setTitle(resourseId);        }        /\*       \* We call super.onBackPressed(); when the stack entry count is > 0. if it       \* is instanceof EmpListFragment or if the stack entry count is == 0, then       \* we prompt the user whether to quit the app or not by displaying dialog.       \* In other words, from EmpListFragment on back press it quits the app.       \*/      @Override      public void onBackPressed() {          FragmentManager fm = getSupportFragmentManager();          if (fm.getBackStackEntryCount() > 0) {              super.onBackPressed();          } else if (contentFragment instanceof EmpListFragment                  || fm.getBackStackEntryCount() == 0) {              //Shows an alert dialog on quit              onShowQuitDialog();          }      }        public void onShowQuitDialog() {          AlertDialog.Builder builder = new AlertDialog.Builder(this);          builder.setCancelable(false);            builder.setMessage("Do You Want To Quit?");          builder.setPositiveButton(android.R.string.yes,                  new DialogInterface.OnClickListener() {                      @Override                      public void onClick(DialogInterface dialog, int id) {                          finish();                      }                  });          builder.setNegativeButton(android.R.string.no,                  new DialogInterface.OnClickListener() {                      public void onClick(DialogInterface dialog, int id) {                          dialog.cancel();                      }                  });          builder.create().show();      }        @Override      public void onFinishDialog() {          if (employeeListFragment != null) {              employeeListFragment.updateView();          }      }  } |

## **Output:**

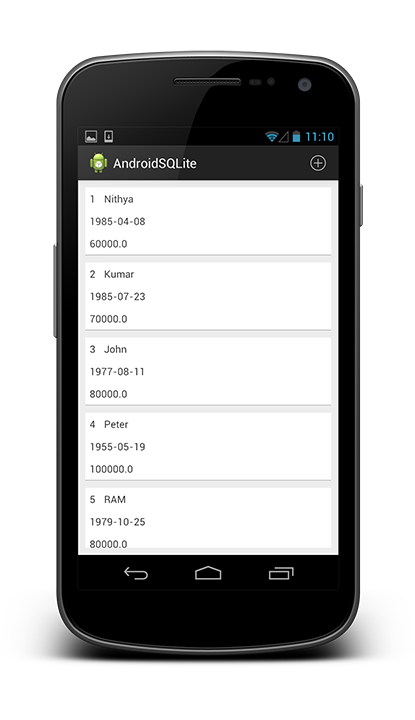
### **Insert a row – EmpAddFragment**

[](http://androidopentutorials.com/wp/wp-content/uploads/2014/03/1.android-sqlite-insert-row.png?1a8226)

### **Update a row – CustomEmpDialogFragment**

[](http://androidopentutorials.com/wp/wp-content/uploads/2014/03/2.android-sqlite-update-row.png?1a8226)

### **Fetch all rows – EmpListFragment**

[](http://androidopentutorials.com/wp/wp-content/uploads/2014/03/3.android-sqlite-read-rows-display-listview.png?1a8226)