



# Android SQLite Class Example Code

## 1. Database Helper Class: DbHelper.java (CRUD Logic)

```
package com.example.myfirstdemoapp;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;

// DbHelper manages database creation, version control, and all CRUD operations.
public class DbHelper extends SQLiteOpenHelper {

    // Database and Table Constants
    private static final String DATABASE_NAME = "ContactDb";
    private static final int DATABASE_VERSION = 1;
    private static final String CONTACT_TABLE = "contact_table";

    // Column Names
    private static final String KEY_ID = "id"; // Primary Key
    private static final String KEY_NAME = "name";
    private static final String KEY_PHONE_NUMBER = "phone_number";

    public DbHelper(@Nullable Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
}
```

@Override

```
public void onCreate(SQLiteDatabase db) {  
    String CREATE_CONTACT_TABLE = "CREATE TABLE " + CONTACT_TABLE + "("  
        + KEY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "  
        + KEY_NAME + " TEXT,"  
        + KEY_PHONE_NUMBER + " TEXT" + ")";  
    db.execSQL(CREATE_CONTACT_TABLE);  
}
```

@Override

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

// --- CREATE Operation (Existing) ---

```
public void insertData(String name, String number) {  
    SQLiteDatabase db = getWritableDatabase();  
    ContentValues value = new ContentValues();  
    value.put(KEY_NAME, name);  
    value.put(KEY_PHONE_NUMBER, number);  
    db.insert(CONTACT_TABLE, null, value);  
    db.close();  
}
```

// --- READ Operation (Existing) ---

```
public Cursor readData() {  
    SQLiteDatabase db = getReadableDatabase();  
    return db.rawQuery("SELECT * FROM " + CONTACT_TABLE, null);  
}
```

```

// --- UPDATE Operation (NEW) ---
/**
 * Updates an existing contact record based on the ID.
 * @param id The ID of the record to update (string format as read from cursor).
 * @param name New name.
 * @param number New phone number.
 * @return boolean: true if at least one row was updated.
 */
public boolean updateData(String id, String name, String number) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();

    // New values
    contentValues.put(KEY_ID, id);
    contentValues.put(KEY_NAME, name);
    contentValues.put(KEY_PHONE_NUMBER, number);

    // Update the row where ID matches the given id
    int affectedRows = db.update(CONTACT_TABLE, contentValues, KEY_ID + " = ?",
        new String[] { id });
    db.close();
    return affectedRows > 0;
}

// --- DELETE Operation (NEW) ---
/**
 * Deletes a contact record based on the ID.
 * @param id The ID of the record to delete.
 * @return int: Number of rows deleted.
 */
public int deleteData(String id) {

```

```

        SQLiteDatabase db = this.getWritableDatabase();

        // Delete the row where ID matches the given id
        int deletedRows = db.delete(CONTACT_TABLE, KEY_ID + " = ?",
            new String[] { id });

        db.close();

        return deletedRows;
    }
}

```

## 2. Layout File: activity\_main.xml (UI Design)

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="20dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/ediID"
        android:hint="Enter ID (for Update/Delete)"
        android:inputType="number"
        android:layout_marginTop="10dp"/>

    <EditText

        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```

```
android:id="@+id/ediName"  
android:hint="Enter your name"/>
```

```
<EditText
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:id="@+id/ediNumber"  
    android:inputType="number"  
    android:hint="Enter your number"/>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="20sp"  
    android:orientation="horizontal">
```

```
<Button
```

```
    android:layout_width="0dp"  
    android:layout_height="wrap_content"  
    android:text="ADD"  
    android:layout_weight="1"  
    android:id="@+id/btn1"  
    android:layout_marginEnd="5dp"/>
```

```
<Button
```

```
    android:layout_width="0dp"  
    android:layout_height="wrap_content"  
    android:text="SHOW"  
    android:layout_weight="1"  
    android:id="@+id/btn2"  
    android:layout_marginStart="5dp"/>
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="10sp"  
    android:orientation="horizontal">
```

```
<Button
```

```
    android:layout_width="0dp"  
    android:layout_height="wrap_content"  
    android:text="UPDATE"  
    android:layout_weight="1"  
    android:id="@+id/btnUpdate"  
    android:layout_marginEnd="5dp"/>
```

```
<Button
```

```
    android:layout_width="0dp"  
    android:layout_height="wrap_content"  
    android:text="DELETE"  
    android:layout_weight="1"  
    android:id="@+id/btnDelete"  
    android:layout_marginStart="5dp"/>
```

```
</LinearLayout>
```

```
<TextView
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="30sp"  
    android:id="@+id/result"  
    android:textSize="16sp"
```

```
android:textColor="@android:color/black"/>
```

```
</LinearLayout>
```

### 3. Activity File: MainActivity.java (Main Logic)

```
package com.example.myfirstdemoapp;

import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    // UI components declaration
    Button btnAdd, btnShow, btnUpdate, btnDelete; // Buttons
    EditText ediName, ediNumber, ediID; // Input Fields (ediName, ediNumber existing)
    TextView result1;
    DbHelper dbHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
```

```

// --- Initialization ---

ediID = findViewById(R.id.ediID);    // New: For ID input

ediName = findViewById(R.id.ediName);

ediNumber = findViewById(R.id.ediNumber);


btnAdd = findViewById(R.id.btn1);

btnShow = findViewById(R.id.btn2);

btnUpdate = findViewById(R.id.btnUpdate); // New: Update Button

btnDelete = findViewById(R.id.btnDelete); // New: Delete Button


result1 = findViewById(R.id.result);


dbHelper = new DbHelper(MainActivity.this);


// --- Event Listeners Setup ---

AddData();

ViewAll();

UpdateData(); // NEW

DeleteData(); // NEW
}


// Existing: CREATE (Add) operation
public void AddData() {

    btnAdd.setOnClickListener(view -> {

        dbHelper.insertData(ediName.getText().toString(), ediNumber.getText().toString());

        Toast.makeText(MainActivity.this, "Data inserted successfully", Toast.LENGTH_SHORT).show();

        ediName.setText("");

        ediNumber.setText("");

    });

}

```



```

// Existing: READ (Show) operation

public void ViewAll() {

    btnShow.setOnClickListener(view -> {

        Cursor cursor = dbHelper.readData();

        if (cursor.getCount() == 0) {

            result1.setText("Data not found");

            cursor.close();

            return;

        }

        StringBuilder sb = new StringBuilder();

        while(cursor.moveToNext()){

            sb.append("ID: ").append(cursor.getInt(0))

            .append("\nName: ").append(cursor.getString(1))

            .append("\nNumber: ").append(cursor.getString(2))

            .append("\n-----\n");

        }

        result1.setText(sb.toString());

        cursor.close();

    });

}

```

```

// NEW: UPDATE operation

public void UpdateData() {

    btnUpdate.setOnClickListener(view -> {

        String id = ediID.getText().toString();

        String name = ediName.getText().toString();

        String number = ediNumber.getText().toString();

        if (id.isEmpty()) {

```

```

        Toast.makeText(MainActivity.this, "Please enter ID for update!",
Toast.LENGTH_SHORT).show();

        return;
    }

    if (name.isEmpty() || number.isEmpty()) {

        Toast.makeText(MainActivity.this, "Please enter Name and Number for update!",
Toast.LENGTH_SHORT).show();

        return;
    }

    boolean isUpdate = dbHelper.updateData(id, name, number);

    if (isUpdate) {

        Toast.makeText(MainActivity.this, "Data Updated!", Toast.LENGTH_LONG).show();
    } else {

        Toast.makeText(MainActivity.this, "Data Not Updated (Check ID)",
Toast.LENGTH_LONG).show();
    }

    });
}

// NEW: DELETE operation
public void DeleteData() {

    btnDelete.setOnClickListener(view -> {

        String id = ediID.getText().toString();

        if (id.isEmpty()) {

            Toast.makeText(MainActivity.this, "Please enter ID to delete!",
Toast.LENGTH_SHORT).show();

            return;
        }
    }

```

```
int deletedRows = dbHelper.deleteData(id);

if (deletedRows > 0) {
    Toast.makeText(MainActivity.this, deletedRows + " Record(s) Deleted!",
Toast.LENGTH_LONG).show();

    ediID.setText(""); // Clear ID field

    ViewAll(); // Refresh the displayed data
} else {
    Toast.makeText(MainActivity.this, "Deletion Failed (Check ID)",
Toast.LENGTH_LONG).show();
}
});
}
}
```