

sqllitedatabaseactivity.java

```
package com.example.sqlitedatabaseactivity;

import static android.widget.Toast.makeText;

import android.annotation.SuppressLint;
import android.database.Cursor;
import android.os.Bundle;

import com.google.android.material.snackbar.Snackbar;

import androidx.appcompat.app.AppCompatActivity;

import android.text.Editable;
import android.view.View;

import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;

import com.example.sqlitedatabaseactivity.databinding.ActivityMainBinding;

import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class SQLiteDatabaseActivity extends AppCompatActivity {

    private DbAdapter dbAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sqlite);

        dbAdapter = new DbAdapter(this);
        dbAdapter.open();

        EditText eTName=findViewById(R.id.edtTextName);
        EditText eTAge=findViewById(R.id.edtTextAge);
        Button btnInsert = findViewById(R.id.btnInsert);
        Button btnRetrieve = findViewById(R.id.btnRetrieve);
        Button btnUpdate = findViewById(R.id.btnUpdate);
        Button btnDelete = findViewById(R.id.btnDelete);
        Button
        btnRetrieveParticularRecord=findViewById(R.id.btnRetrieveRecord);
        EditText eTrowId=findViewById(R.id.edtTextRowId);
        TextView tvRecordResult=findViewById(R.id.tvRecordResult);

        btnInsert.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```

        String name=eTName.getText().toString();
        String ageString=eTAge.getText().toString();
        int age=Integer.valueOf(ageString);
        // Example: Insert a record
        long rowId = dbAdapter.insertRecord(name,age);
        if (rowId != -1) {
            showToast("Record inserted with ID: " + rowId);
        } else {
            showToast("Failed to insert record");
        }
    }
});

btnRetrieve.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Example: Retrieve all records
        Cursor cursor = dbAdapter.getAllRecords();
        if (cursor != null && cursor.getCount() > 0) {
            // Handle retrieved records
            showToast("Number of records: " + cursor.getCount());
        } else {
            showToast("No records found");
        }
    }
});

btnUpdate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Editable editTextRowId=eTrowId.getText();

        String idString = editTextRowId.toString();

        // Example: Update a record
        if(!idString.isEmpty()) {
            long id = Long.parseLong(idString);

            boolean success = dbAdapter.updateRecord(id, "Updated
Name", 30);

            if (success) {
                showToast("Record updated successfully");
            } else {
                showToast("Failed to update record");
            }
        } else {
            showToast("Id is Empty");
        }
    }
});

btnDelete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Editable editTextRowId=eTrowId.getText();

        String idString = editTextRowId.toString();

        // Example: Delete a record

```

```

        if(!idString.isEmpty()) {
            long id = Long.parseLong(idString);

            boolean success = dbAdapter.deleteRecord(id);
            if (success) {
                showToast("Record deleted successfully");
            } else {
                showToast("Failed to delete record");
            }
        } else {
            showToast("Id is Empty");
        }
    }
});

btnRetrieveParticularRecord.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Editable editTextRowId=eTrowId.getText();

        String idString = editTextRowId.toString();
        if (!idString.isEmpty()) {
            try {
                // Parse the string to a long
                long recordId = Long.parseLong(idString);

                // Call getRecord to retrieve the record
                Cursor cursor = dbAdapter.getRecord(recordId);

                if (cursor != null && cursor.moveToFirst()) {
                    // Retrieve data from the cursor
                    @SuppressWarnings("Range") String name =
cursor.getString(cursor.getColumnIndex(DbAdapter.KEY_NAME));
                    @SuppressWarnings("Range") int age =
cursor.getInt(cursor.getColumnIndex(DbAdapter.KEY_AGE));

                    // Display the result in the TextView
                    String resultText = "Name: " + name + ", Age: "
+ age;

                    tvRecordResult.setText(resultText);
                } else {
                    showToast("Record not found");
                }
            } catch (NumberFormatException e) {
                showToast("Invalid ID format");
            }
        } else {
            showToast("Please enter a Record ID");
        }
    }
});

}

private void showToast(String message) {
    Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
}

```

```

@Override
protected void onDestroy() {
    super.onDestroy();
    dbAdapter.close();
}
}

```

Activity_sqlite.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    android:padding="16dp"
    android:gravity="center"
    tools:context=".SQLiteDatabaseActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Enter Name:"
        android:textSize="20dp"/>

    <EditText
        android:id="@+id/edtTextName"
        android:layout_width="match_parent"
        android:textSize="20dp"
        android:layout_height="wrap_content"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Enter Age:"
        android:textSize="20dp"/>

    <EditText
        android:id="@+id/edtTextAge"
        android:layout_width="match_parent"
        android:textSize="20dp"
        android:layout_height="wrap_content"/>

    <Button
        android:id="@+id/btnInsert"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Insert Record" />

    <Button
        android:id="@+id/btnRetrieve"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Total Records Count" />

```

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Enter rowId:"
    android:textSize="20dp"/>

<EditText
    android:id="@+id/edtTextRowId"
    android:layout_width="match_parent"
    android:textSize="20dp"
    android:layout_height="wrap_content"/>

<Button
    android:id="@+id/btnUpdate"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Update Record" />

<Button
    android:id="@+id/btnDelete"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Delete Record" />

<Button
    android:id="@+id/btnRetrieveRecord"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Retrieve Particular Record" />

<TextView
    android:id="@+id/tvRecordResult"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="" />

</LinearLayout>

```

DbAdapter:

```

package com.example.sqlitedatabaseactivity;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DbAdapter {
    // Database constants
    private static final String DATABASE_NAME = "your_database";
    private static final int DATABASE_VERSION = 1;

    // Table constants
    private static final String TABLE_NAME = "your_table";

```

```

public static final String KEY_ID = "_id";
public static final String KEY_NAME = "name";
public static final String KEY_AGE = "age";

// Database creation SQL statement
private static final String DATABASE_CREATE =
    "create table " + TABLE_NAME + " ("
        + KEY_ID + " integer primary key autoincrement, "
        + KEY_NAME + " text not null, "
        + KEY_AGE + " integer not null);";

private final Context context;
private DatabaseHelper DBHelper;
private SQLiteDatabase db;

public DbAdapter(Context ctx) {
    this.context = ctx;
    DBHelper = new DatabaseHelper(context);
}

// Helper class to manage database creation and version management.
private static class DatabaseHelper extends SQLiteOpenHelper {
    DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        try {
            db.execSQL(DATABASE_CREATE);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        // Implement if you need to handle database upgrades
    }
}

// Open the database
public DbAdapter open() throws SQLException {
    db = DBHelper.getWritableDatabase();
    return this;
}

// Close the database
public void close() {
    DBHelper.close();
}

// Insert a record into the database
public long insertRecord(String name, int age) {
    ContentValues initialValues = new ContentValues();
    initialValues.put(KEY_NAME, name);
    initialValues.put(KEY_AGE, age);
    return db.insert(TABLE_NAME, null, initialValues);
}

```

```

    }

    // Retrieve all records from the database
    public Cursor getAllRecords() {
        return db.query(TABLE_NAME, new String[]{KEY_ID, KEY_NAME, KEY_AGE},
            null, null, null, null, null);
    }

    // Retrieve a specific record based on ID
    public Cursor getRecord(long rowId) throws SQLException {
        Cursor cursor = db.query(true, TABLE_NAME,
            new String[]{KEY_ID, KEY_NAME, KEY_AGE},
            KEY_ID + "=" + rowId,
            null, null, null, null, null);
        if (cursor != null) {
            cursor.moveToFirst();
        }
        return cursor;
    }

    // Update a record in the database
    public boolean updateRecord(long rowId, String name, int age) {
        ContentValues args = new ContentValues();
        args.put(KEY_NAME, name);
        args.put(KEY_AGE, age);
        return db.update(TABLE_NAME, args, KEY_ID + "=" + rowId, null) > 0;
    }

    // Delete a record from the database
    public boolean deleteRecord(long rowId) {
        return db.delete(TABLE_NAME, KEY_ID + "=" + rowId, null) > 0;
    }
}

```

OUTPUT :

