**Main.dart**

import 'package:database/database\_helper.dart';

import 'package:flutter/material.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.deepOrange,

),

home: const MyHomePage(title: 'DataBase'),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({Key? key, required this.title}) : super(key: key);

final String title;

@override

State<MyHomePage> createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text("Database"),

),

body: Center(

child: Column(

children: [

ElevatedButton(onPressed: () async{

int i = await DatabaseHelper.instance.insert(

{DatabaseHelper.columnName: "SIES"}

);

print("Inserted id is: $i");

},child: Text("Insert Record")),

ElevatedButton(onPressed: () async{

List<Map<String,dynamic>> rows = await DatabaseHelper.instance.showAll();

print(rows);

},child: Text("Show all Records")),

ElevatedButton(onPressed: () async{

int updateId = await DatabaseHelper.instance.update({

DatabaseHelper.columnId:2,

DatabaseHelper.columnName:"SIESCOMS"

});

print("Updated Id is: $updateId");

},child: Text("Update Record")),

ElevatedButton(onPressed: () async{

int rowsAffected = await DatabaseHelper.instance.delete(3);

print(rowsAffected);

},child: Text("Delete Record")),

],

),

)

);

}

}

**Database\_helper.dart**

import 'dart:io';

import 'package:path\_provider/path\_provider.dart';

import 'package:sqflite/sqflite.dart';

import 'package:path/path.dart';

class DatabaseHelper{

static final \_dbName = "student.db";

static final \_dbVersion = 1;

static final \_tabName = "student";

static final columnId = "\_id";

static final columnName = "name";

DatabaseHelper.\_privateConstructor();

static final DatabaseHelper instance = DatabaseHelper.\_privateConstructor();

static Database? \_database;

Future<Database?> get database async{

\_database??= await \_initiateDatabase();

return \_database;

}

\_initiateDatabase() async{

Directory direc = await getApplicationDocumentsDirectory();

String path = join(direc.path, \_dbName);

return await openDatabase(path,version: \_dbVersion, onCreate: \_onCreate);

}

Future \_onCreate(Database db,int version) async{

db.execute(

'''

CREATE TABLE $\_tabName(

$columnId INTEGER PRIMARY KEY,

$columnName TEXT NOT NULL

)

'''

);

}

Future<int> insert(Map<String,dynamic> row) async{

Database? db = await instance.database;

return await db!.insert(\_tabName,row);

}

Future<List<Map<String,dynamic>>> showAll() async{

Database? db = await instance.database;

return await db!.query(\_tabName);

}

Future<int> update(Map<String,dynamic> row) async{

Database? db = await instance.database;

int id = row[columnId];

return await db!.update(\_tabName, row, where: '$columnId = ?',whereArgs: [id]);

}

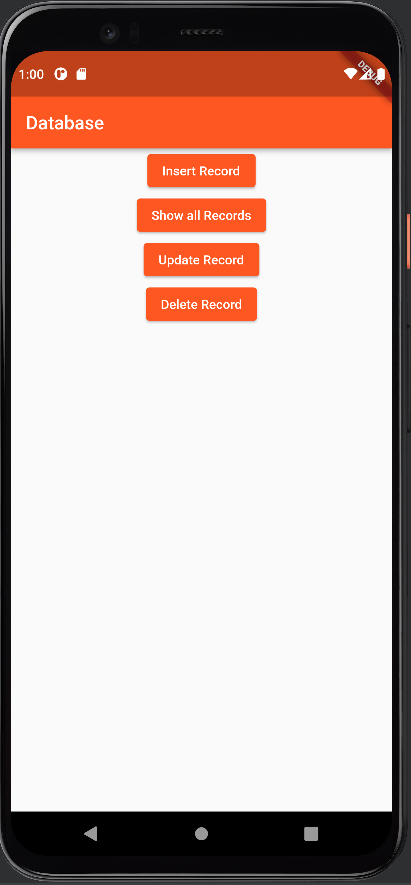
Future<int> delete(int id) async{

Database? db = await instance.database;

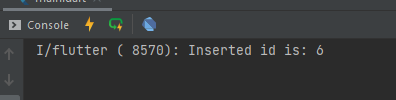
return await db!.delete(\_tabName, where: '$columnId = ?',whereArgs: [id]);

}

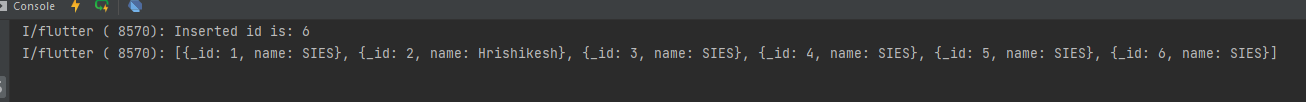
}

**Output:**

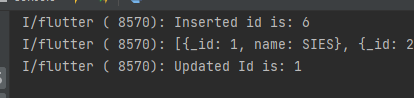
Insert Record:



Show All Records:



Update Record:



Delete Record:

