# Lab 22

Instructor: Sidra Khatoon

E-mail: skhatoon[@uit.edu](mailto:ad@uit.edu)

# Objective

# The objective of lab is creating a SQLite database in a Flutter app to store and retrieve data.

**Student Information**

|  |  |
| --- | --- |
| **Student Name** |  |
| **Student ID** |  |
| **Date** |  |

**Assessment**

|  |  |
| --- | --- |
| **Marks Obtained** |  |
| **Remarks** |  |
| **Signature** |  |

# Objective

# The objective of lab is creating a SQLite database in a Flutter app to store and retrieve data

# Instructions

You have to perform the following tasks yourselves. Raise your hand if you face any difficulty in understanding and solving these tasks. **Plagiarism** is an abhorrent practice and you should not engage in it.

# How to Submit?

Submit lab work using Teams.

# Storing Data in SQLite database in Flutter

In Flutter apps we need persistent data storage. This means storing data/information within the app itself, even when the app is closed or the device is turned off. Here’s where SQLite comes in.

# What is SQLite?

SQLite is a lightweight relational database management system (RDBMS) that’s embedded within the application.

# Why Use SQLite in Flutter?

Here are some key benefits of using SQLite with Flutter:

* Lightweight and Fast: SQLite boasts a small footprint and efficient performance, perfect for mobile apps where resources are limited.
* Serverless and Easy to Use: There’s no need to set up or manage a separate server. SQLite integrates seamlessly into your Flutter project.
* Cross-Platform Compatibility: SQLite works across Android, iOS, and even web platforms where Flutter is applicable.

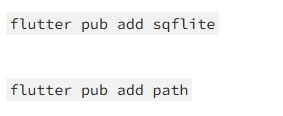
**Getting Started with SQLite in Flutter**

To leverage SQLite in your Flutter applications, you’ll utilize the popular sqflite package. This package provides a Dart API for interacting with the SQLite database. It offers functionalities for:

* Creating and opening databases
* Defining tables and columns
* Performing CRUD operations (Create, Read, Update, Delete) on data
* Querying the database to retrieve specific information

**Step#1: Adding dependencies**

In order to use SQLite in Flutter we need two dependencies i.e. sqflite and path. Add these dependencies in the pubspec.yaml file.



**Step#2: Create database\_service.dart in lib folder**

After adding dependencies we will create a Database Service in database\_service.dart. The Database Service will provide the following services:

* Create database
* Create tables
* Add/insert data in tables
* Retrieve data from tables
* Update data in tables
* Delete data from tables

**Step-3: User interface for using Database Service (Adding and retrieving data from the SQLite database)**

Create UI in main.dart

**Assessment:**

**Basics of SQLite**

1. What is SQLite, and why is it suitable for mobile applications?
2. Explain the difference between SQLite and other relational databases like MySQL or PostgreSQL.
3. Mention three features of SQLite.

**SQLite in Flutter**

1. How does Flutter support SQLite database integration?
2. What is the role of the sqflite package in Flutter?
3. Explain the lifecycle of a database in a Flutter app.

**SQL Commands**

1. Write the SQL command to:
2. Create a table named users with columns id (INTEGER, PRIMARY KEY), name (TEXT), and age (INTEGER).
3. Insert a row into the users table.
4. Retrieve all users above the age of 25.
5. Update the name of a user where id = 1.
6. Delete a user where id = 3.