# Connecting with SQLite Database

To connect app with SQLite database, we need to create an CLI app with the following command

npx react-native init projectcli

Once the project folder is created, navigate into the project folder:

cd projectcli

and install the SQLite package with the following command:

npm install react-native-sqlite-storage -save

Later, if we want to react native screens in our project, we also need to install the following dependencies:

npm install @react-navigation/native --save

npm install @react-navigation/native-stack --save

npm install react-native-screens react-native-safe-area-context --save

Use the following code to perform operations on SQLite

import React, {useEffect, useState} from 'react';

import { Button, View, Text, FlatList, ActivityIndicator } from 'react-native';

import SQLite from 'react-native-sqlite-storage';

const db = SQLite.openDatabase(

  {

      name: 'MainDB',

      location: 'default',

  },

  () => { },

  error => { console.log(error) }

);

const createTable = () => {

  db.transaction((tx) => {

      tx.executeSql(

          "CREATE TABLE IF NOT EXISTS "

          + "Users "

          + "(ID INTEGER PRIMARY KEY AUTOINCREMENT, Name TEXT, Age INTEGER);"

      )

  })

}

const setData = async () => {

var name = "noman";

var age = "50";

      try {

          await db.transaction(async (tx) => {

              await tx.executeSql(

                  "INSERT INTO Users (Name, Age) VALUES (?,?)",

                  [name, age]

              );

          })

        //  navigation.navigate('Home');

      } catch (error) {

          console.log(error);

      }

}

const getData = () => {

  try {

      db.transaction((tx) => {

          tx.executeSql(

              "SELECT Name, Age FROM Users",

              [],

              (tx, results) => {

                  var len = results.rows.length;

                  if (len > 0) {

                      var userName = results.rows.item(0).Name;

                      var userAge = results.rows.item(0).Age;

                      console.log(userName);

                      console.log(userAge);

                  }

              }

          )

      })

  } catch (error) {

      console.log(error);

  }

}

const updateData = async () => {

   var name="jawad";

      try {

          db.transaction((tx) => {

              tx.executeSql(

                  "UPDATE Users SET Name=?",

                  [name],

                  () => { Alert.alert('Success!', 'Your data has been updated.') },

                  error => { console.log(error) }

              )

          })

      } catch (error) {

          console.log(error);

      }

}

const removeData = async () => {

  try {

      db.transaction((tx) => {

          tx.executeSql(

              "DELETE FROM Users",

              [],

              console.log("Record deleted"),

              error => { console.log(error) }

          )

      })

  } catch (error) {

      console.log(error);

  }

}

const App = () => {

  return (

<View

style = {{marginTop: 50}}

>

<Button

title="Create Table"

onPress = {() => createTable()}

/>

<Button

title="Set Data"

onPress = {() => setData()}

/>

<Button

title="Get Data"

onPress = {() => getData()}

/>

<Button

title="Update Data"

onPress = {() => updateData()}

/>

<Button

title="Remove Data"

onPress = {() => removeData()}

/>

    </View>

  )

  }

export default App;