Assignment no. 3

43212

Title - Android database connectivity

Problem statement -

- Android - database connectivity - create an SQLite application for android application & perform database applications.

Theory -

What is SQLite ?

It is an SQL datebase. So in SQL database we sturt data base in tables. The tables are structure for storing data consisting rows & colums

Android SQLite -

- It is very lightweight database which comes with android as. Android SQLite combines a clean SQL interface with a very small memory footprint & decent speed. For android SQLite, is "baked into" android runtime, so every android application can create its SQLite database,
- SQLite is typical relational database.

Android SQLite Helper -

- Android has features available to handle changing database schemes, which mostly depends on using SQLite Open Helper class
- It is designed to get rid of two very common problems:

[43212]

]When application runs first time, we do not yet have a database. So we will have to create tables, indexes, starter data on so on

- SQLiteOpenHelper is designed to wrap up these logic to create & custom subclass upgrade as per our specifications.

```
public DataBaseHelper(context context){
    Super (context, DB-NAME, null, DB-Version);
}
```

1. onCreate : To create database

2. onUpgrade : It is called when scheme version we need not match the scheme version of database. It passes us a SQLite Database object

Opening & closing Android SQLite database connection :-

- Before performing any database operations like insert, delete, update in table, open database connectivity by calling get WritableDatabase() method

```
public DBManager open() throw SQLException{
    dbHelper = new DatabaseHelper (context);
    database = dbHelper.getWritableDatabase();
    return this;
}
```

[43212]

- The dbHelper is an instance of subclass of SQLiteOpenHelper
- To close database connection, following method is involved

```
public void close () {
    db.Helper.close();
 }
```

Android SQLite cursor:-

- A cursor represents the entire result ~~set~~ of query. Once query is fetched a call to cursor MoveTofirst() is made. calling moveTofirst() ~~two~~ does two things:
- It allows us to test whether query returned an empty set
- It moves cursor to first result.

```
public Cursor_fetch () {
String[] colums = new string[] {
 Database Helper ID, DatabaseHelper, SUBJECT,
  DatabaseHelper.DESC};
 Cursor cursor = database. query (DatabaseHelper
                         TABLE_NAME,
    columns, null, null, null, null, null);
 if (cursor != NULL) {
     cursor.movetofirst();
 }
 return cursor;
}
```

43212

Conclusion- We learnt that,

- How to use SQLite DB & perform CRUD operations on it Android studio
- How to use cursors.