43233

Date: / / Page no:

Assignment No 3.

Title : Android Database Connectivity.

Problem Statement: Android - database connectivity
Create SQLite application for Android application & perform CRUD operation (create, Read, update, Delete) database applications

Theory :-

1) what is SQLite ?

SQLite is SQL database. So in SQL database we store the data in tables. The tables are structure for string data consisting of rows & columns.

Android SQLite :

It is very lightweight database which comes with Android as Android SQLite combines clean SQL interface with very small memory footprint & decent speed. For android SQLite, is "baked into" Android runtime. So every Android applications can create its own SQLite databases.

- SQLite is typical relational database.

Android SQLite SQLite

Android has features available to handle changing database schemas, which mostly depends on using SQLite Helper class

SQLiteOpen Helper is designed to get rid of two very common problems:

43233

Date / / Page no:

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

```
public DataBaseHelper ( Context context) {
        Super ( context, DB_NAME, null, DB_VERSION);
}
```

Opening & closing Android SQLite database connection -

- Before performing any database operations like insert, delete, update records in the table, open database connection by calling getWritableDatabase() method
- The dbHelper is an instance of the subclass of SQLiteOpenHelper
- To close database connection, following method

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

The dbHelper is an instance of subclass of SQLiteOpenHelper To class the database connection, following method is invoked

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

43233

Date: / / Page no:

Android SQLite Cursor-

A cursor represents entire result set of query. Once query is fetched a call to cursor moveToFirst() is made calling moveToFirst() does two things:

→ It allows us to test whether query returned as empty sets.

→ It move cursor to first result.

```
public Custom Fetch() {
    String[] columns = new String[]{
    DatabaseHelper.ID, DatabaseHelper.SUBJECT,
    DatabaseHelper.DESC};
    Cursor cursor = database.query(DatabaseHelper.
    TABLE_NAME, columns, null, null, null, null);
    if (cursor != NULL) {
        cursor.moveToFirst();
    }
    return cursor;
}
```

Conclusion: To this assignment, I have learnt how to use SQLite DB & perform CRUD operations on it ~~Add~~
Android Studio
How to use Cursor.