Using rawQuery() can perform Read operation like Select.

Sample rawQuery examples for Select

Examples

Db.rawQuery("SELECT id, name FROM people WHERE ?= ?", new String[] {"column_name", "David"});

Db.rawQuery("SELECT id, name FROM people WHERE name = ? AND id = ?", new String[] {"David", "2"});

String q = "SELECT * FROM customer";

Cursor mCursor = mDb.rawQuery(q, null);

String q = "SELECT * FROM customer WHERE _id = " + customerDbId ;

Cursor mCursor = mDb.rawQuery(q, null);

execSQL() - Can perform CIUD

dBase.execSQL("create table if not exists employee(id number,name varchar(50), desig varchar(50),dept varchar(50))");

```
  private static final String DATABASE_NAME = "dbForTest.db";
  private static final int DATABASE_VERSION = 1;
  private static final String TABLE_NAME = "diary";
  private static final String TITLE = "title";
  private static final String BODY = "body";

String sql = "CREATE TABLE " + TABLE_NAME + " (" + TITLE
        + " text not null, " + BODY + " text not null " + ");";
db.execSQL(sql);
db.execSQL("DROP TABLE IF EXISTS diary");
```

String query;
query = “CREATE TABLE books (bookId INTEGER PRIMARY KEY, bookName TEXT)”;
database.execSQL(query);

String query;
query = “DROP TABLE IF EXISTS books”;
database.execSQL(query);

static final String DATABASE_CREATE =
"create table contacts (_id integer primary key auto increment, "+ "name text not null, email text not null);";
db.execSQL(DATABASE_CREATE);

db.execSQL("ALTER TABLE foo ADD COLUMN new_column INTEGER DEFAULT 0");

String dbQuery = "CREATE TABLE Items (id INTEGER PRIMARY KEY
AUTOINCREMENT,name TEXT, description TEXT)";
sqLiteDatabase.execSQL(dbQuery);

db.execSQL("INSERT INTO Semesters
VALUES(sessionName,startYear,startMonth,startDay,endYear,endMonth,endDay);");

String Insert_Data="INSERT INTO CashData
VALUES(2,'Electricity',500,1,'04/06/2017')";
db.execSQL(Insert_Data);

ourDatabase.execSQL("INSERT INTO peopleTable (persion_name,persion_hotness)
VALUES('Lalit','Kushwah')");

mydatabase.execSQL("CREATE TABLE IF NOT EXISTS login(Username
VARCHAR,Password VARCHAR);");
mydatabase.execSQL("INSERT INTO login VALUES('admin','admin');");

String update = "UPDATE "+table_name+" SET Steps = '"+ new_text +"' WHERE ID = "
+ position;
db.execSQL(update);

String update = "UPDATE STUDENT SET IS_name = '"+ name +"' WHERE ID = " + id;
db.execSQL(update);

```
final String Create_CashBook =
        "CREATE TABLE CashData ("
        + "id INTEGER PRIMARY KEY AUTOINCREMENT,"
        + "Description TEXT,"
        + "Amount REAL,"
        + "Trans INTEGER,"
        + "EntryDate TEXT);";

    db.execSQL(Create_CashBook);
final String Insert_Data="INSERT INTO CashData
VALUES(2,'Electricity',500,1,'04/06/2017')";
    db.execSQL(Insert_Data);
```

DB.insert() examples

Convenience method for inserting a row into the database.

Syntax - insert(String table, String nullColumnHack, ContentValues values)
Example - 1

```
dBase.execSQL("create table if not exists employee(id number,name varchar(50),desig
varchar(50),dept varchar(50))");

ContentValues values=new ContentValues();
values.put("id",Integer.parseInt(et1.getText().toString()));
values.put("name",et2.getText().toString());
values.put("desig",et3.getText().toString());
values.put("dept", et4.getText().toString());
dBase.insert("employee",null,values);
```

Example - 2

```
final String Create_CashBook =
        "CREATE TABLE CashData ("
        + "id INTEGER PRIMARY KEY AUTOINCREMENT,"
        + "Description TEXT,"
        + "Amount REAL,"
        + "Trans INTEGER,"
        + "EntryDate TEXT);";

    db.execSQL(Create_CashBook);
```

```
ContentValues insertValues = new ContentValues();
insertValues.put("Description", "Electricity");
insertValues.put("Amount", 500);
insertValues.put("Trans", 1);
insertValues.put("EntryDate", "04/06/2011");
db.insert("CashData", null, insertValues);
```

DB.query() – Examples

Syntax :

```
Cursor cursor = sqLiteDatabase.query(
   tableName, tableColumns, whereClause, whereArgs, groupBy, having, orderBy);
```

tableColumns - columns parameter is constructed as follows.

```
String[] columns = new String[]{KEY_ID, KEY_CONTENT};
```

Example 1

```
String table = "table2";
String[] columns = {"column1", "column3"};
String selection = "column3 =?";
String[] selectionArgs = {"apple"};
String groupBy = null;
String having = null;
String orderBy = "column3 DESC";
String limit = "10";

Cursor cursor = db.query(table, columns, selection, selectionArgs, groupBy, having,
orderBy, limit);
```

Example 2

```
Cursor c=dBase.query("employee",null,"id=?",
        new String[]{et1.getText().toString()},null,null,null);
```

Example 3

```
Cursor cursor = db.query(TABLE_CONTACTS, new String[] { KEY_ID,
          KEY_NAME, KEY_PH_NO }, KEY_ID + "=?",
```

```
        new String[] { String.valueOf(id) }, null, null, null, null);
```

DB.update()

Convenience method for updating rows in the database and return an int value.

update(String table, ContentValues values, String whereClause, String[] whereArgs)

Example - 1

```
int count=dBase.update("employee",values,"id=?",
      new String[]{et1.getText().toString()});
```

Example - 2

```
    ContentValues contentValues = new ContentValues();
     contentValues.put("name", name);
     contentValues.put("phone", phone);
     contentValues.put("email", email);
     contentValues.put("street", street);
     contentValues.put("place", place);
     db.update("contacts", contentValues, "id = ? ", new String[] {
                Integer.toString(id) } );
```

Example – 3

```
myDB.update(TableName, "(Field1, Field2, Field3)" + " VALUES ('Bob', 19, 'Male')",
"where _id = 1", null);
```

Example – 4

```
ContentValues cv = new ContentValues();
cv.put("Field1","Bob");
cv.put("Field2","19");
cv.put("Field2","Male");
myDB.update(TableName, cv, "_id="+id, null);

myDB.update(MY_TABLE_NAME, cv, "_id = ?", new String[]{id});
```

DB.delete()

Convenience method for deleting rows in the database and return int.

delete(String table, String whereClause, String[] whereArgs)

Example – 1

```
String table = "beaconTable";
String whereClause = "_id=?";
String[] whereArgs = new String[] { String.valueOf(row) };
db.delete(table, whereClause, whereArgs);
```

Example – 2

```
db.delete("tablename","id=? and name=?",new String[]{"1","jack"});
```

Example – 3

```
db.delete(TABLE_CONTACTS, KEY_NAME + "=" + mname, null);
```

Example – 4

```
db.delete("contacts", "id = ? ",new String[] { Integer.toString(id) });
```

Example – 5 – Delete all the records

```
sqLiteDatabase.delete(MYDATABASE_TABLE, null, null);
```