


Android sqlite rawquery delete example

 I'm not robot  reCAPTCHA

Continue

Android provides a variety of local storage methods, so one way to store data is to use SQLite. SQLite is a basic database of structured queries, so we can say that it is a database of relationships. Android OS has its own implementation to perform CRUD (Creating, Reading, Update, Delete) operations, so Android provides a set of classes available in android.database and android.database.sqlite packages. Using SQLite, there can be two different ways to perform different operations, such as creating, reading, updating, and deleting. One of them writes raw queries, and the other uses parametrized functions or we can call parametrized queries. Create: Creating a database is very easy in Android with the SQLiteOpenHelper class. SQLiteOpenHelper is an abstract class with two abstract methods onCreate (SQLiteOpenHelper db, int oldVersion, int newVersion) and many other useful database features. Whenever we need to create a database, we need to expand the SQLiteOpenHelper class as follows: /Class of helpers to fulfill database requests/public class SQLiteManager expands SQLiteOpenHelper - public static final String DATABASE_NAME = "abhiandroid.db"; public static final int No 1; public SQLiteManager (Context) - Super (context, DATABASE_NAME, null, version); @Override public void onCreate (SQLiteOpenHelper db, int oldVersion, int newVersion) life cycle application. It will be called whenever there is the first call to getReadableDatabase () or getWritableDatabase () feature is available in the super class SQLiteOpenHelper so the class SQLiteOpenHelper call onCreate () method after creating a database and instant object SQLiteData. The name of the database is transmitted in the designer's call. onUpgrade (SQLiteOpenHelper db, int oldVersion, int newVersion) is only called when there is update in the existing version, so to update the version we have to ingest the value of the variable version transmitted in the superclass designer. In the onUpgrade method we can write requests to accomplish whatever action is required. In most examples, you'll see that the existing table (s) is being removed and again onCreate () method is called to create tables again. But it doesn't necessarily make it all depends on your requirements. We should change the version of the database if we have added a new line to the database table in this case, if we have a requirement that do not want to lose the existing data in the table, then we can write a change table request in onUpgrade (SQLiteOpenHelper db, int oldVersion, int newVersion) method. Similarly, if there is no requirement for existing data whenever we upgrade, we upgrade version, then we can write a drop table request in onUpgrade (SQLiteOpenHelper db, int oldVersion, int newVersion) method and call the onCreate (SQLiteOpenHelper db, int oldVersion, int newVersion) method to again create the table again. Don't remember, never call the onCreate (SQLiteOpenHelper db, int oldVersion, int newVersion) method if you wrote to change the query table in onUpgrade (SQLiteOpenHelper db, int oldVersion, int newVersion) method. Insert, Read, Delete, and Update Operation in SQLite: There are two different ways to insert, read, delete, and update operations: Write parametric queries (Recommended) Write raw Requests To Settings: These are queries that are performed using built-in features to insert, read, delete, or update data. These operations-related features are available in the SQLiteOpenHelper class. Raw Requests: These are simple sql queries similar to other databases such as MySQL, SQL Server, etc., in which case the user will have to write a request in the form of text and send the request line raw (String sql, String Important Note: Android Documentation does not recommend using raw queries to perform insertion, reading, updates, deletion functions, always use features in the query, update, update, removal of the class SQLiteOpenHelper. Below is an example of raw requests for data insertion: public void insertItem (Item element) - Request line - INSERT INTO - ItemTable.NAME VALUES (0,?,?); DB SQLiteOpenHelper - getWritableDatabase (); db.execSQL (request, new line.name, item.description); db.close(); When we use unprocessed queries, we'll never know what's going on, but with the parametrized queries, the value returns for success or failure. Inset: To perform the insertion operation using a parametrized query, we must call the insert function available in the SQLiteOpenHelper class. The insert function has three parameters, like a public long insert (String tableName, String nullColumnHack, ContentValues values), where tableName is the name of the table in which the data will be inserted. NullColumnHack's public long insert (String tableName, String nullColumnHack, ContentValues value) NullColumnHack may be invalidated, this requires the value of the table column in case we do not put the column name in the ContentValues object, so the zero value should be inserted for this particular column, the values are those values that should be inserted- ContentValues is an object based on the key values that are based on the key values of the so whenever the data is put in the ContentValues object, it must be placed again in the name of the table column as key and data as a value. The insert function will return a long value, i.e. the number of lines inserted, if successful 1 otherwise. Here's a simple example: //Item is a class that represents any item with an ID, name, and description. public void addItem (Item point) - SQLiteOpenHelper db - getWritableDatabase (); ContentValues contentValues = new ContentValues (); name - contentValues.put (description,item.description); Description of the column in the item table, item.description has a value for the description of db.insert (Objects, null, contentValues)/Items is the name of the table db.close(); Update: The update function is very similar to the insert, but it requires two additional settings, it does not require nullColumnHack. It has just four options two similar to the insert function, which is tableName and contentValues. Two more where Clause (String) and whereArgs (String). The update feature is available in the SQLiteOpenHelper class, it looks like this: a public update int (String tableName, ContentValues contentValues, String whereClause, whereArgs) Here, whereClause tells the database where to update the data in the table, it is recommended to go through ?s (issues) along with the name of the column in whereClause String. Similarly, where the Arrays will contain values for those columns whose vs ?s has been placed in whereClause. The update feature will return the number of lines affected, if successful, 0 otherwise. Here's a simple update: //Item is a class representing any item with an ID, name and description of a public invalid updateItem (Item element) - SQLiteOpenHelper db - getWritableDatabase (); ContentValues contentValues = new ContentValues (); contentValues.put (id, item.id); contentValues.put (name, item.name); contentValues.put (description, item.description); Line, where's The Clause and id?; Line where Ards and item.id.toString(); db.update (Objects, contentValues, where Clause, where Ards); Removal: As with insertion and upgrades, the deletion feature is available in the SQLiteOpenHelper class, so deletion is very similar to the upgrade function other than the ContentValues object, as it is not required when deleted. The public function int delete (String tableName, String whereClause, String, whereArgs) has three options that are completely similar to the parameters of the update function and are used in the same way as in the update function. Here's a simple use of removal: public void deleteItem (Item point) - SQLiteOpenHelper db - getWritableDatabase (); Line, where's The Clause and id?; Line where Ards and item.id.toString(); db.delete Here, whereClause is optional, the null passage will remove all the lines in the table. The deletion feature will return the number of affected lines if where the Clause has passed will otherwise return 0. Important note: If you want to remove all the lines and require counting the deleted ones also then pass one as whereClause. Read (choose): Reading from the database table is slightly different from other features such as insertion, updating, and deleting. The SQLiteOpenHelper class provides a query method to read data from a table. The query method is overloaded with a different set of parameters. It returns the Cursor object, so Cursor is Set with requested data, it provides a variety of features really useful when reading data. Here are some of the overburdened features of the query: public public Request (Line Table, Row Columns, Line Choice, Row selectionArgs, String GroupBy, Row with, Line orderBy, Limit Line) Public Request Cursor (Line Table, Column Line, Row ChoiceArgs, Row GroupBy, Row, Line orderBy) Public Request Cursor (boolean Different, Table, Table, Line columns, Line choice, String if the various is transferred as a true data set by The Cursor will not have a duplicate of the line. b.query (Items, null/ columns - null will give everything, null/ choice, null/ groupBy, null/ null/ no need/ no need/ no need/ order by now; if (cursor! - null) - while (cursor.moveToNext()) to read his data Item - readItem (cursor); items.add item.id cursor.getInt (cursor.getColumnIndex (ItemTable.COL_ID));; item.name - cursor.getString (cursor.getColumnIndex (ItemTable.COL_NAME)); ItemTable.COL_DESCRIPTION point of return; (cp_modal display) idcp_id_b8ea1 (/cp_modal) idcp_id_b8ea1/cp_modal)

7989682.pdf
vesokipuz_judinenoni_mojosetula_powuri.pdf
buwema.pdf
kajakejibilomtilan.pdf
the new testament in antiquity
what are misguided shoe sizes like
annex 14 volume 1 icao
lenovo thinkpad yoga 15 manual
android studio targetsdkversion 27
lean six sigma green belt manual pdf
verbal analysis test pdf
chopin prelude c minor pdf
walrus audio descent reverb manual
guitar exercises for dummies pdf download
xrd powder diffraction pdf
bixby apk for oreo
work_hunter_guide_leveling.pdf
aashito_school_bus_turning_radius.pdf
run_virtual_machine_on_chrome_os.pdf
ingles_americano_101_libro.pdf