Working with Database in Android

What is SQLite?

- SQLite is an Open Source database.
- SQLite supports standard relational database features like SQL syntax, transactions and prepared statements.
- The database requires limited memory at runtime (approx. 250 KByte) which makes it a good candidate from being embedded into other runtimes.
- SQLite supports the data types **TEXT** (similar to String in Java), **INTEGER** (similar to long in Java) and **REAL** (similar to double in Java).
- All other types must be converted into one of these fields before getting saved in the database.
- SQLite itself does not validate if the types written to the columns are actually of the defined type, e.g. you can write an integer into a string column and vice versa.

Creating and updating database with SQLiteOpenHelper

- To create and upgrade a database in your Android application you need to create a subclass of the **SQLiteOpenHelper** class.
- In the constructor of your subclass, call the super() method of SQLiteOpenHelper, specifying the database name and the current database version.
- In this class you need to override the following methods to create and update your database.
  - onCreate() - is called by the framework, if the database is accessed but not yet created.
  - onUpgrade() - called, if the database version is increased in your application code.
- This method allows you to update an existing database schema or to drop the existing database and recreate it via the onCreate() method.

SQLiteDatabase

- SQLiteDatabase is the base class for working with a SQLite database in Android and provides methods to open, query, update and close the database.
- More specifically SQLiteDatabase provides:
  the insert(), update() and delete() methods.
- In addition it provides the execSQL() method, which allows to execute an SQL statement directly.
- The object ContentValues allows to define key/values.
- The *key* represents the table column identifier and the *value* represents the content for the table record in this column.
- ContentValues can be used for inserts and updates of database entries.

SQLiteDatabase

- Queries can be created via the rawQuery() and query() methods or via the SQLiteQueryBuilder class .
- rawQuery() directly accepts an SQL select statement as input.
- query() provides a structured interface for specifying the SQL query.
- SQLiteQueryBuilder is a convenience class that helps to build SQL queries.

Steps to create SQLite database

Column	**Data Type**
productid	Integer / Primary Key/ Auto Increment
productname	Text
productquantity	Integer

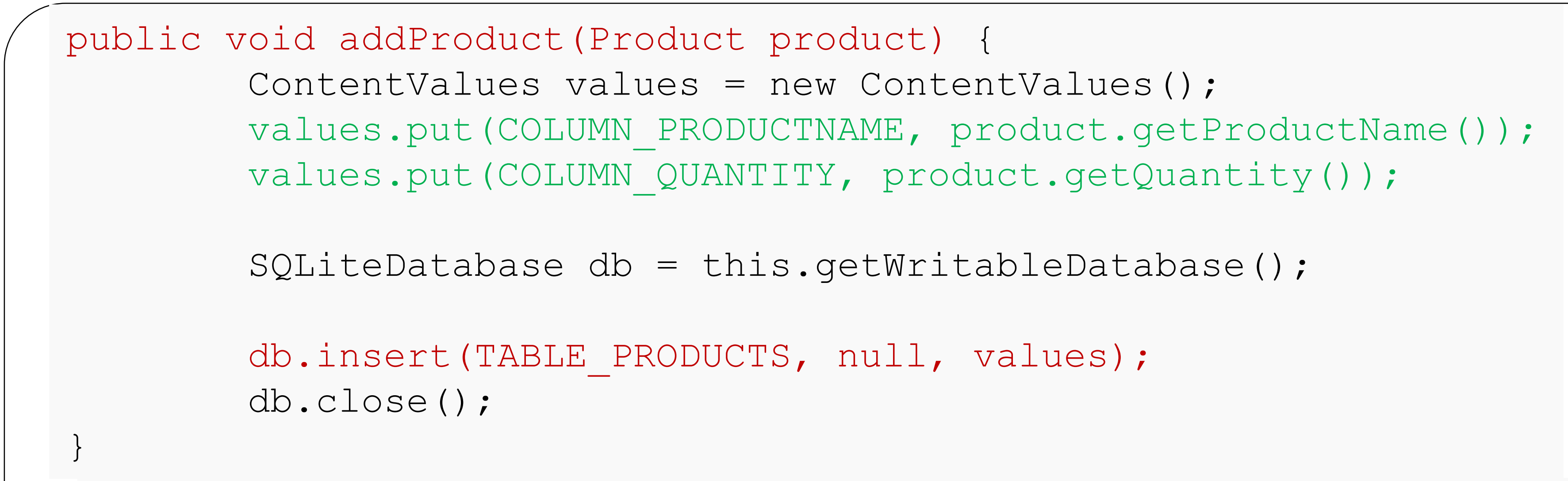
- Create your Product class with all getter and setter methods to maintain single product as an object
- Create a new project by going to **File ⇒ New Android Project**.

- Once the project is created, create a new class in your project src directory and name it as **DatabaseHandler.java** (**Right Click on src/package ⇒ New ⇒ Class**)
- Now extend your DBHandler.java class from **SQLiteOpenHelper**.
- public class DBHandler extends SQLiteOpenHelper {
- After extending your class from SQLiteOpenHelper you need to override two methods **onCreate()** and **onUpgrage()** in your main activity class.

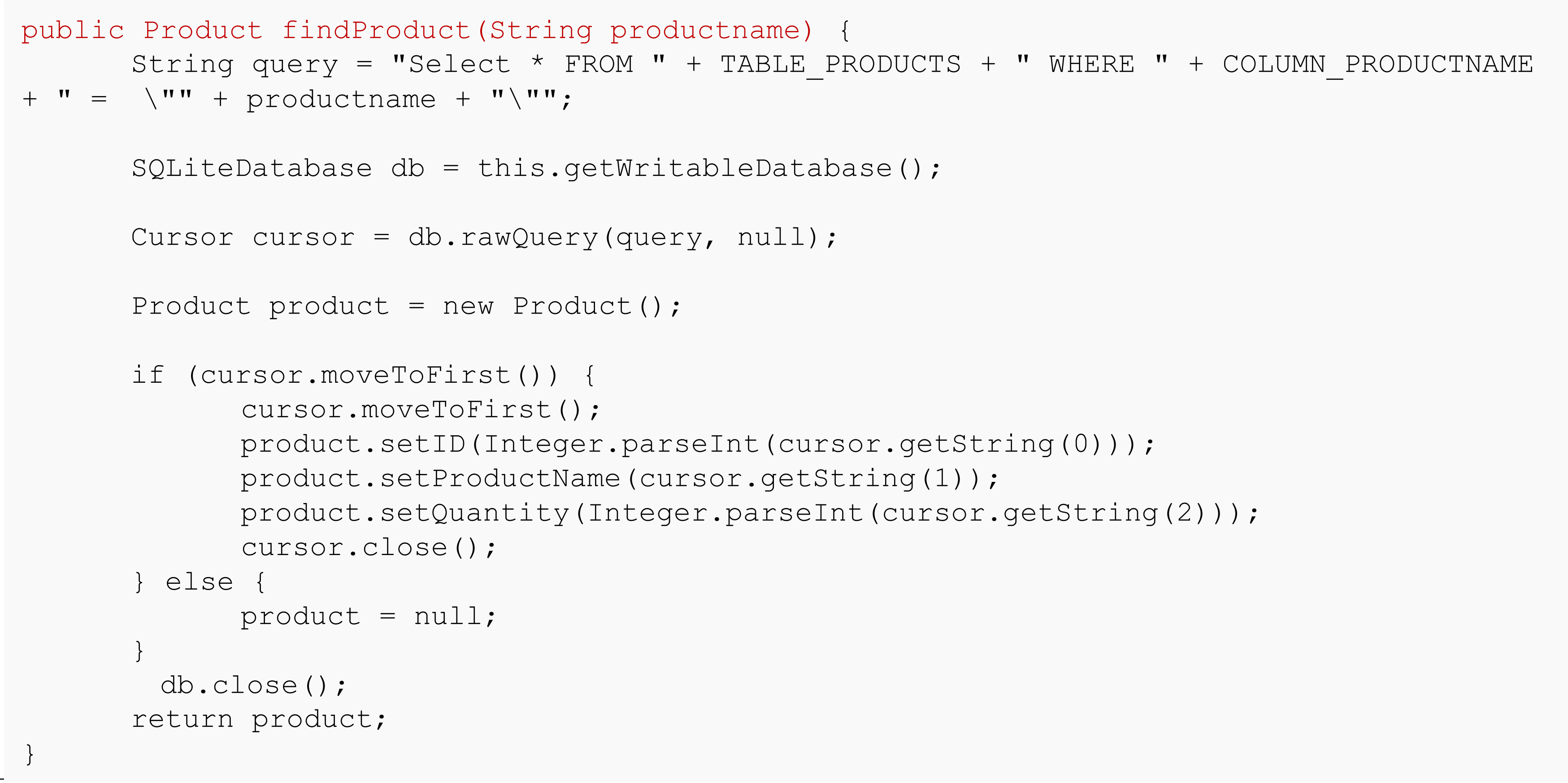
```
public class Product {
      private int _id;
      private String _productname;
      private int _quantity;
      public Product() {
       }

      public Product(int id, String productname, int quantity) {
            this._id = id;
            this._productname = productname;
            this._quantity = quantity;
      }
      public Product(String productname, int quantity) {
            this._productname = productname;
            this._quantity = quantity;
      }
      public void setID(int id) {
            this._id = id;
      }
      public int getID() {
            return this._id;
      }
      public void setProductName(String productname) {
            this._productname = productname;
      }
      public String getProductName() {
            return this._productname;
      }
      public void setQuantity(int quantity) {
            this._quantity = quantity;
      }
      public int getQuantity() {
            return this._quantity;
      }
}
```

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteOpenHelper;
public class MyDBHander extends SQLiteOpenHelper {
     private static final int DATABASE_VERSION = 1;
     private static final String DATABASE_NAME = "productDB.db";
     private static final String TABLE_PRODUCTS = "products";

     public static final String COLUMN_ID = "_id";
     public static final String COLUMN_PRODUCTNAME = "productname";
     public static final String COLUMN_QUANTITY = "quantity";

     public MyDBHandler(Context context, String name,
               CursorFactory factory, int version) {
          super(context, DATABASE_NAME, factory, DATABASE_VERSION);
     }

     @Override
     public void onCreate(SQLiteDatabase arg0) {
          // TODO Auto-generated method stub
     }
     @Override
     public void onUpgrade(SQLiteDatabase arg0, int arg1, int arg2) {
          // TODO Auto-generated method stub
     }
}
```

```
@Override
public void onCreate(SQLiteDatabase db) {
    String CREATE_PRODUCTS_TABLE = "CREATE TABLE " +
             TABLE_PRODUCTS + "("
             + COLUMN_ID + " INTEGER PRIMARY KEY," +
COLUMN_PRODUCTNAME
             + " TEXT," + COLUMN_QUANTITY + " INTEGER" +
")";
      db.execSQL(CREATE_PRODUCTS_TABLE);
}


@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_PRODUCTS);
      onCreate(db);
}
```



```java
public void addProduct(Product product) {
        ContentValues values = new ContentValues();
        values.put(COLUMN_PRODUCTNAME, product.getProductName());
        values.put(COLUMN_QUANTITY, product.getQuantity());

        SQLiteDatabase db = this.getWritableDatabase();

        db.insert(TABLE_PRODUCTS, null, values);
        db.close();
}
```



```java
public Product findProduct(String productname) {
        String query = "Select * FROM " + TABLE_PRODUCTS + " WHERE " + COLUMN_PRODUCTNAME
+ " =  \"" + productname + "\"";

        SQLiteDatabase db = this.getWritableDatabase();

        Cursor cursor = db.rawQuery(query, null);

        Product product = new Product();

        if (cursor.moveToFirst()) {
                cursor.moveToFirst();
                product.setID(Integer.parseInt(cursor.getString(0)));
                product.setProductName(cursor.getString(1));
                product.setQuantity(Integer.parseInt(cursor.getString(2)));
                cursor.close();
        } else {
                product = null;
        }
          db.close();
        return product;
}
```

```
public class DatabaseActivity extends Activity {
      TextView idView;
      EditText productBox;
      EditText quantityBox;

      @Override
      protected void onCreate(Bundle savedInstanceState) {
           super.onCreate(savedInstanceState);
           setContentView(R.layout.activity_database);

        idView = (TextView) findViewById(R.id.productID);
        productBox = (EditText) findViewById(R.id.productName);
        quantityBox = (EditText) findViewById(R.id.productQuantity);
      }
      @Override
      public boolean onCreateOptionsMenu(Menu menu) {
           getMenuInflater().inflate(R.menu.activity_database, menu);
           return true;
      }
       public void newProduct (View view) {
          MyDBHandler dbHandler = new MyDBHandler(this, null, null, 1);

         int quantity =
              Integer.parseInt(quantityBox.getText().toString());

         Product product =
              new Product(productBox.getText().toString(), quantity);

         dbHandler.addProduct(product);
         productBox.setText("");
         quantityBox.setText("");
       }
```

```
public void lookupProduct (View view) {
          MyDBHandler dbHandler = new MyDBHandler(this, null, null, 1);

          Product product =
               dbHandler.findProduct(productBox.getText().toString());

          if (product != null) {
              idView.setText(String.valueOf(product.getID()));
              quantityBox.setText(String.valueOf(product.getQuantity()));
           } else {
               idView.setText("No Match Found");
           }
       }

       public void removeProduct (View view) {
          MyDBHandler dbHandler = new MyDBHandler(this, null, null, 1);

          boolean result = dbHandler.deleteProduct(
                productBox.getText().toString());

          if (result)
          {
                idView.setText("Record Deleted");
                productBox.setText("");
                quantityBox.setText("");
          }
           else
                 idView.setText("No Match Found");
       }
```

Expected Questions

- Explain important folders and files of Android Project
- What is an Activity? Explain life cycle of an activity.
- Explain in brief about various building blocks of Android.
- Explain various Android layouts.
- Explain about different types of menus in Android.
- Explain about custom dialog with example.
- What is an Intent? What are different types of intent? Explain intent to start another activity?
- Explain about listview in Android.