Android uses a set of classes that allows you to manage a SQLite database. There are three types of object that do the bulk of

this work.

**The SQLite Helper**

You create a SQLite helper by extending the SQLiteOpenHelper class. This enables you to create and manage databases.

**Cursors**

A Cursor lets you read from and write to the database. It’s like a ResultSet in JDBC.

**The SQLite Database**

The SQLiteDatabase class gives you access to the database. It’s like a SQLConnection in JDBC.

1. **Create the database.**

Before we can do anything else, we need to get the SQLite helper to create version 1 (the first version) of our Starbuzz

database.

2. **Create the Drink table and populate it with drinks.**

Once we have a database, we can create a table in it. The table structure needs to reflect the attributes in the current

Drink class, so it needs to be able to store the name, description, and image resource ID of each drink. We’ll then add

three drinks to it.

The app has the same structure as before except that we’re replacing the file *Drink.java* with a SQLite helper and a SQLite

Starbuzz database. The SQLite helper will maintain the Starbuzz database, and provide access to it for the other activities

**The SQLite helper manages your database**

The **SQLiteOpenHelper** class is there to help you create and maintain your SQLite databases. Think of it as a personal

assistant who’s there to take care of the general database housekeeping.

create a SQLite helper by writing a class that extends the **SQLiteOpenHelper** class. When you do this, you ***must*** override

the onCreate() and onUpgrade() methods. These methods are mandatory.

The onCreate() method gets called when the database first gets created on the device. The method should include all the code

needed to create the tables you need for your app.

The onUpgrade() method gets called when the database needs to be upgraded.

**1. Specify the database**

There are two pieces of information the SQLite helper needs in order to create the database.

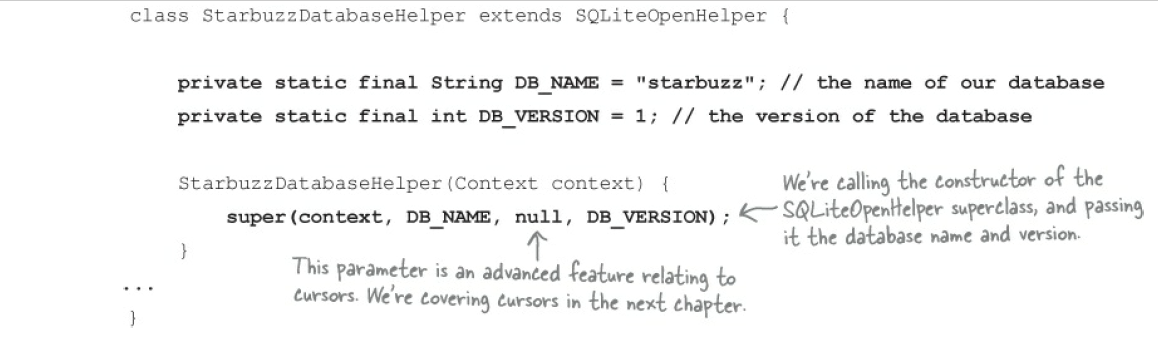
First, we need to give the database a name. By giving the database a name, we make sure that the database remains on the

device when it’s closed. If we don’t, the database will only be created in memory, so once the database is closed, it will

disappear.

The second piece of information we need to provide is the version of the database. The database version needs to be an integer

value, starting at 1. The SQLite helper uses this version number to determine whether the database needs to be upgraded



**Storage classes and data-types**

Each column in a table is designed to store a particular type of data. For example, in our DRINK table, the DESCRIPTION

column will only ever store text data. Here are the main data types you can use in SQLite, and what they can store:

INTEGER Any integer type

TEXT Any character type

REAL Any floating-point number

NUMERIC Booleans, dates, and date-times

BLOB Binary Large Object

Unlike most database systems, you don’t need to specify the column size in SQLite.

**The onCreate() method is called when the database is created**

The SQLite helper is in charge of creating the SQLite database the first time it needs to be used. First, an empty database is

created on the device, and then the SQLite helper onCreate() method is called.

The onCreate() method is passed a SQLiteDatabase object as a parameter. We can use this to run our SQL command with

the method:

**The SQLiteDatabase class gives you access to the database.**

@Override

public void onCreate(SQLiteDatabase db){

**db.execSQL("CREATE TABLE DRINK ("**

**+ "\_id INTEGER PRIMARY KEY AUTOINCREMENT, "**

**+ "NAME TEXT, "**

**+ "DESCRIPTION TEXT, "**

**+ "IMAGE\_RESOURCE\_ID INTEGER);");**

}

**ContentValues drinkValues = new ContentValues();**

