Lab 3

Table of Contents

Overview **3**
Data and table creation **3**
Query practice **3**
Scripts ‘design’? **5**
Appendices **6**
Customer Table 6
CustomerOrder Table 6
Stock Table 6

LAB 3

Overview

In the previous lab you became accustomed to how schema are realised in SQL via the creation of a table. You created a table and inserted data into this, using sqlite3. You then learned how to create an <.sql> script, which should always have the <.sql> extension, and you were encouraged to use a simple text editor, depending on which machine you are using (Notepad++, TextMate for Windows and Mac OS, respectively). **N.B:** The programs, sqlite3 and the text editor form an important part of your database 'development environment'. Therefore, if you are reading this overview and you are not confident with this environment, or if you have not looked at it yet, you should definitely make sure you have completed the previous lab, and understood it.

Data and table creation

The appendices to this lab contains data:

- Customer;
- CustomerOrder and;
- Stock

Before you do the following, remember…always write your code in a incremental manner. By this I mean write a bit of code, then test it. For example, if I wanted to drop a table and then create it in the script, I might firstly write the code that creates it, call the script to check that works. If it does work, then put some code in the script that drops the table (if the table exists) before creating it. Then check if that works. You will find your own way of doing things, but don't make life difficult for yourself by trying to write all you code before running it… it's likely that there will be errors in several places and it will take you a long time to debug your code….

…For each table you will write a single script that:

- drops the table if it already exists
- creates the table
- inserts some appropriate values into the table based on the data.

The commands to achieve this for the customer table were provided in the previous lab, so you can use that SQL as a starting point. However, you should move both the creation and insertion steps into single files for each table. Name the scripts:

- *<create_customer.sql>*
- *<create_customer_order.sql>*
- *<create_stock.sql>*

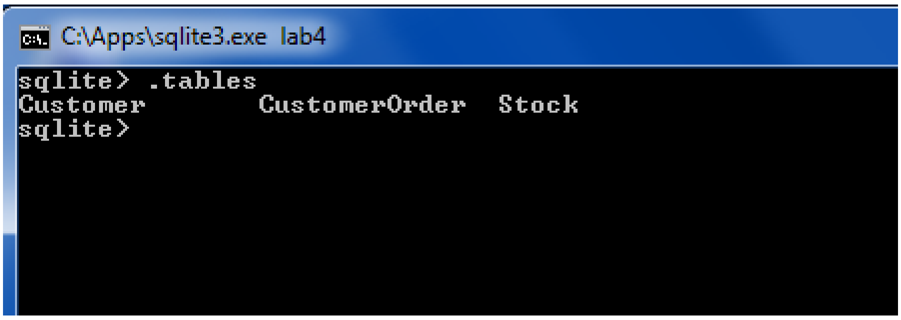
…respectively. Notice that some of the data in the Stock table is missing, so you might have to have two separate insert sections in that script, one for the all the rows with all the available data, and one for the two rows with missing data.

Query practice

If you have written your scripts and executed them, then you will have the table available to start querying against. Remember, check which tables are in your database with the following command:

<.tables>

And you should see something like this:



```
sqlite> .tables
Customer        CustomerOrder   Stock
sqlite>
```

Write SQL SELECT commands to achieve the output depicted:

List the customer no., name & email for each customer.

```
NW001|DISKS 'N STUFF|t.banks@abc.co.uk
NW002|CARRAGHER & SON|joe@carragher.freeserve.co.uk
NE001|TONY KENNEDY|music_shop@eastport.com
SE001|THE MUSIC SHOP|music_shop@eastport.com
SW001|THREADS ETC|0111-111-2222
```

List the customer of each customer order.

```
NW002
SE001
NE001
NW002
NW001
```

List the stock no., description & price for each stock item.

```
TS001|AUGUST COVER T-SHIRT|12.99
TS002|BAND SHOT T-SHIRT|12.99
TS003|ADAM (AUGUST) T-SHIRT|12.99
TS004|SATELLITES COVER T-SHIRT|12.99
SS001|AUGUST COVER SWEATSHIRT|25.99
SS002|SATELLITES COVER SWEATSHIRT|25.99
CD001|AUGUST & EVERYTHING AFTER|11.99
CD002|RECOVERING THE SATELLITES|11.99
CD003|ACROSS A WIRE|14.99
VD001|CROWS VIDEO COLLECTION|9.99
LJ001|CROWS LEATHER JACKET|125
LP001|AUGUST LP|18.99
LP002|SATELLITES LP|18.99
LP003|ACROSS A WIRE LP|21.99
```

```
MD001|AUGUST MD|16.99
MD002|SATELLITES MD|16.99
MD003|ACROSS A WIRE MD|18.99
CS001|AUGUST CASSETTE|8.99
CS002|SATELLITES CASSETTE|8.99
```

Have a google for examples of the usage of the following SQL key words:

WHERE

IN

For example, look at the form of the syntax found here:

https://www.sqlitetutorial.net/sqlite-in/

….using such key words can therefore allow you to select subsets of data from one table where a chosen field value is in a column of another table. For example:

List the customer nos and email of each customer who has placed an order.

```
NW001|t.banks@abc.co.uk
NW002|joe@carragher.freeserve.co.uk
NE001|music_shop@eastport.com
SE001|music_shop@eastport.com
```

Scripts 'design'?

In one of the exercises above, and it is worth reflecting on this, I asked you to:

- *<create_customer.sql>*
- *<create_customer_order.sql>*
- *<create_stock.sql>*

To be honest, the naming of the scripts isn't very descriptive… if you have written the scripts correctly they will contain the SQL that drops the table, if it exists, followed by SQL that creates the table, then, finally, SQL queries that insert all the data.

It's worth reflecting on this…if we call a table, e.g. "create_customer.sql" why would it contain anything other than code that creates the table? Maybe the inclusion of the DROP command is fine, but the insertion of data…could that be kept in a separate sql file? ….

- *<insert_customer_data.sql>*
- *<insert_customer_order_data.sql>*
- *<insert_stock_data.sql>*

…and could you imagine having a different kind of setup altogether… maybe you could put the data into .csv files, with a 'schama.csv' and 'data.csv' for stock, customer_order and customer data. Then you could have some simpler scripts that inserted the data with the <*.import*> sqlite3 command:

- <import_customer_data.sql>
- <import_customer_order_data.sql>
- <import_stock_data.sql>

In this lab, I encouraged you down one particular way of implementing the scripts, because this relates to the previous lab …but you should always think about different ways of implementing the same thing… it's good to practice coding, of course, but thinking/questioning what you are doing is more important than anything. As Einstein said:

"…the important thing is to not stop questioning. Curiosity has its own reason for existence"…

If you want to try the <.import> approach demonstrated in week-02 Lecture: demo, then feel free to do that and contact me during one of your sessions for feedback. However, if you do that and I complement your work, then if it helps know that Einstein also said the:

"…only way to escape the corruptible effect of praise is to go on working."

Appendices

Customer Table

customer_no	name	address	telno	email
NW001	DISKS 'N STUFF	5 MAIN ROAD, NEWTOWN, ST56 34ER	0142-432-5768	t.banks@abc.co.uk
NW002	CARRAGHER & SON	235 WAGNER WAY, ROMANVILLE, ST54 6WR	0134-223-5637	joe@carragher.freeserve.co.uk
NE001	TONY KENNEDY	55 BARNTON ROAD, MEADOW PARK, DUNWICH, D1 3TR	0162-444-1434	
SE001	THE MUSIC SHOP	CULLEN WYND, EASTPORT, FR54 3WX	0122-385-9028	music_shop@eastport.com
SW001	THREADS ETC.	11 CARNABY AVENUE, MAINTOWN, M1 2MM	0111-111-2222	

CustomerOrder Table

ORDER NO	ORDERDATE	CUSTOMERNO	SOURCE
00158	10-DEC-98	NW002	EMAIL
00159	11-DEC-98	SE001	PHONE
00160	11-DEC-98	NE001	MAIL
00161	14-DEC-98	NW002	EMAIL
00162	16-DEC-98	NW001	PHONE

Stock Table

STOCK NO	DESCRIPTION	QUANTITY IN HAND	REORDERLEVEL	REORDER QUANTITY	PRICE
TS001	AUGUST COVER T-SHIRT	55	30	30	12.99
TS002	BAND SHOT T-SHIRT	50	30	50	12.99
TS003	ADAM (AUGUST) T-SHIRT	28	30	30	12.99
TS004	SATELLITES COVER T-SHIRT	55	30	25	12.99
SS001	AUGUST COVER SWEATSHIRT	55	20	50	25.99
SS002	SATELLITES COVER SWEATSHIRT	55	20	50	25.99
CD001	AUGUST & EVERYTHING AFTER	125	100	100	11.99
CD002	RECOVERING THE SATELLITES	155	50	50	11.99
CD003	ACROSS A WIRE	300	150	200	14.99

VD001	CROWS VIDEO COLLECTION	229	100	50	9.99
LJ001	CROWS LEATHER JACKET	52	10	20	125
CS001	AUGUST CASSETTE	15			8.99
CS002	SATELLITES CASSETTE	43			8.99
LP001	AUGUST LP	10	5	10	18.99
LP002	SATELLITES LP	12	10	10	18.99
LP003	ACROSS A WIRE LP	52	10	20	21.99
MD001	AUGUST MD	5	5	10	16.99
MD002	SATELLITES MD	10	7	10	16.99
MD003	ACROSS A WIRE MD	60	10	20	18.99

~~~~~
~~~~~