UNIVERSITY OF TROMSØ

INF-2700: Database Systems

Assignment 1
SQL and Database Programming

Camilla Stormoen (cst042)

August 21, 2015

Part 1 SQL

Task 1 Database schema
The inf2700_orders describes the different tables and their attributes. It describes the structure of the
database.
Tables:
Customers
Employees
Offices
OrderDetails
Orders
Payments
Products
ProductLines

Task 2 Run the given SQL queries
a) Prints out name, last name and customername from the customer table.

b) Prints out all the orders from the orders table where shipped date is NULL, which means that the
orders haven't been shipped (status is cancelled, on hold or in process).

c) The customerName is set as "*Customer*" in the header and the quantity ordered is summed together
with SUM and is named "*Total*". The customernumber and ordernumber must be alike in both orders
and customer and orderdetails. Prints out customer and total amount of orders.

d) Prints out the productname and their totalquantity over 1000.

Task 3 Write your own SQL queries
a) See *task3a.sql*

b) See *task3b.sql*

c) See *task3c.slq*

d) See *task3d.slq*

e) See *task3e.sql*

Part 2 Database Programming

1. Introduction
This report describes how to implement a sql-json shell in C and run SQL quieries interactivly.

2. Technical Background
2.1 Sql-json shell
JSON (JavaScript Object Notation) is a lightweight syntax for exchaning data that is designed to be understood easily by humans, and parsed easily by machines.
JSON is defined by two basic structures:
- Name/value pairs
- Ordered list of values

3. Design
The implementation of the sqlite is based on the SQLite library. It is made quite simple, and doesn't include much code or knowledge about implementing a sql-json.
By checking for SQLite-codes as SQLITE_OK, SQLITE_ERROR etc., the SQLite libarary automaticly specify the error message to the user. When an error occure, the database is being closed.
There are two commands that is implemented, "*.help*" and "*.quit*". Using a string compare, can determine if the input is a command or query. If it is a query, a sqlite-prepare function reads the query up to it is zero terminated. Then a step-function loops through the database to find columns such as column-name, column-text and so on. Then it's being printed out as json in the shell as Figure 1 shows.

[{"orderNumber": 10167, "orderDate": "2003-10-23"},
 {"orderNumber": 10167, "orderDate": "2003-10-23"}]

Figure 1

4. Implementation
The implementation is written in C language and output is presented in JSON.
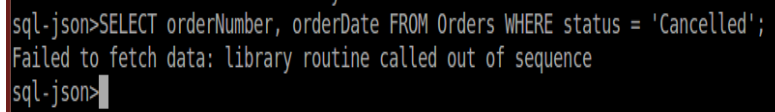
5. Discussion
Both fgets and scanf can be used to read the query. However, the fgets is an better option because this function is terminaded with a null byte, *"/0"*. The fgets return the buffer where everything read is beging putted. The scanf reads only to whitespace, which makes the use of this function bad since a query is most likely more than one word.

A problem with this code is that when you write a query, you can not use whitespace between "*status='Cancelled';*" shown in Figure 2 and 3, which is strange.

```
ite5
Opened database succesfully
Welcome to sql-json
Enter ".help" for instructions
Enter a SQL statment terminated with ";" for a query
Enter ".quit" to exit
sql-json>SELECT orderNumber, orderDate FROM Orders WHERE status='Cancelled';
[{"orderNumber": 10167, "orderDate": "2003-10-23"},
{"orderNumber": 10179, "orderDate": "2003-11-11"},
{"orderNumber": 10248, "orderDate": "2004-05-07"},
{"orderNumber": 10253, "orderDate": "2004-06-01"},
{"orderNumber": 10260, "orderDate": "2004-06-16"},
{"orderNumber": 10262, "orderDate": "2004-06-24"},
]
```

Figure 2

Figure 3

6. Conclution
This report has described how to implement a sql-json shell in C.

7. References
Spring, "*Understanding JSON*" - https://spring.io/understanding/JSON [Online accessed 20-September-2015]

"*SQLite C tutorial*" - http://zetcode.com/db/sqlitec/ [Online accesses 20-September-2015]

SQLite, "SQLite C Interface, List of functions" - *https://www.sqlite.org/c3ref/funclist.html* *[Online accessed 20-September-2015]*