Using Functions, Subqueries, and ROLAP in SQL Queries

Physical Table Design of Sale_Co_DW.db, QC_Checks.db, or Company.db

This physical table structure for the three SQLite databases may be obtained via DBBrowser for SQLite or via .schema dot command in sqlite3.exe.

Sale_Co_DW.db is more of a data warehouse design having fact tables and dimension tables.

QC_Checks.db contains quality check errors from the case study reviewed during Week 05.

Company.db contains company data used to illustrate subqueries in Classwork 6.2.

Exercise:

1) Use any of the three databases above to complete each subsection in step 2. Do not use any of the subqueries illustrated in Classwork 6.1 or Classwork 6.2.

2) Create five unique, executable queries (2 points), using a minimum of four functions on average (2 points), having multiple grouping indexes (2 points), and :
   a. A Type I subquery (2 points) nested with two inner queries (2 points).
   b. A Type II subquery (2 points) nested with two inner queries (2 points).
   c. A Type III correlated subquery (2 points).
   d. The SELECT projection from a table created by a SELECT statement (2 points) with 5 columns.
   e. The SELECT projection from tables saved to a CSV file (2 points).

3) Submit to Canvas Assignment in one PDF document:
   a. Your SQL scripts for each query.
   b. Legible output projection from running each query.

Answers using HMIS.db:

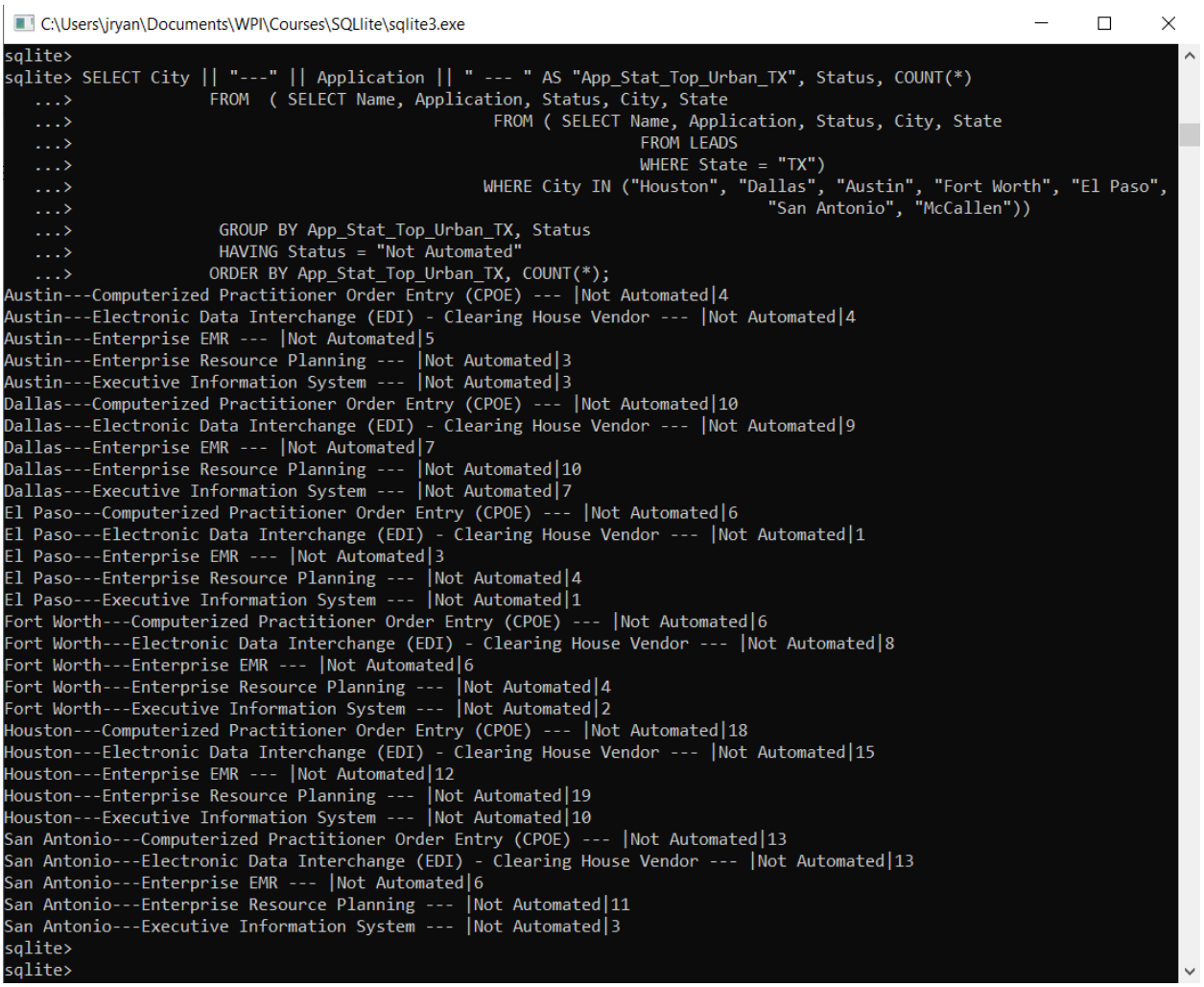
2a Type I subquery (2 points) nested with two inner queries (2 points).

```
SELECT State || " --- " || Application || "---" || Status AS "New_England_App_Status", COUNT(*)
        FROM  ( SELECT Name, Application, Status, City, State
                        FROM ( SELECT Name, Application, Status, City, State
                                        FROM LEADS
                                        WHERE State = "ME" OR State = "NH" OR State = "VT" OR
                                        State = "NY" OR State = "MA" OR State = "NJ" OR State = "CT" OR
                                        State = "RI")
                        WHERE Status = "Not Automated" )
        GROUP BY New_England_App_Status
        ORDER BY New_England_App_Status, COUNT(*);
```

```
sqlite>
sqlite> SELECT State || " --- " || Application || "---" || Status AS "New_England_App_Status", COUNT(*)
   ...>             FROM  ( SELECT Name, Application, Status, City, State
   ...>                                     FROM ( SELECT Name, Application, Status, City, State
   ...>                                                     FROM LEADS
   ...>                                                     WHERE State = "ME" OR State = "NH" OR State = "VT" OR
   ...>                                                     State = "NY" OR State = "MA" OR State = "NJ" OR State = "CT" OR
   ...>                                                     State = "RI")
   ...>                                     WHERE Status = "Not Automated" )
   ...>             GROUP BY New_England_App_Status
   ...>             ORDER BY New_England_App_Status, COUNT(*);
CT --- Computerized Practitioner Order Entry (CPOE)---Not Automated|12
CT --- Electronic Data Interchange (EDI) - Clearing House Vendor---Not Automated|14
CT --- Enterprise EMR---Not Automated|16
CT --- Enterprise Resource Planning---Not Automated|27
CT --- Executive Information System---Not Automated|10
MA --- Computerized Practitioner Order Entry (CPOE)---Not Automated|37
MA --- Electronic Data Interchange (EDI) - Clearing House Vendor---Not Automated|54
MA --- Enterprise EMR---Not Automated|22
MA --- Enterprise Resource Planning---Not Automated|70
MA --- Executive Information System---Not Automated|17
ME --- Computerized Practitioner Order Entry (CPOE)---Not Automated|9
ME --- Electronic Data Interchange (EDI) - Clearing House Vendor---Not Automated|17
ME --- Enterprise EMR---Not Automated|12
ME --- Enterprise Resource Planning---Not Automated|27
ME --- Executive Information System---Not Automated|18
NH --- Computerized Practitioner Order Entry (CPOE)---Not Automated|10
NH --- Electronic Data Interchange (EDI) - Clearing House Vendor---Not Automated|7
NH --- Enterprise EMR---Not Automated|6
NH --- Enterprise Resource Planning---Not Automated|12
NH --- Executive Information System---Not Automated|4
NJ --- Computerized Practitioner Order Entry (CPOE)---Not Automated|39
NJ --- Electronic Data Interchange (EDI) - Clearing House Vendor---Not Automated|42
NJ --- Enterprise EMR---Not Automated|25
NJ --- Enterprise Resource Planning---Not Automated|62
NJ --- Executive Information System---Not Automated|24
NY --- Computerized Practitioner Order Entry (CPOE)---Not Automated|107
NY --- Electronic Data Interchange (EDI) - Clearing House Vendor---Not Automated|125
NY --- Enterprise EMR---Not Automated|83
NY --- Enterprise Resource Planning---Not Automated|162
NY --- Executive Information System---Not Automated|97
RI --- Computerized Practitioner Order Entry (CPOE)---Not Automated|6
RI --- Electronic Data Interchange (EDI) - Clearing House Vendor---Not Automated|4
RI --- Enterprise EMR---Not Automated|1
RI --- Enterprise Resource Planning---Not Automated|10
RI --- Executive Information System---Not Automated|3
VT --- Computerized Practitioner Order Entry (CPOE)---Not Automated|6
VT --- Electronic Data Interchange (EDI) - Clearing House Vendor---Not Automated|5
VT --- Enterprise EMR---Not Automated|3
VT --- Enterprise Resource Planning---Not Automated|5
VT --- Executive Information System---Not Automated|4
sqlite>
```

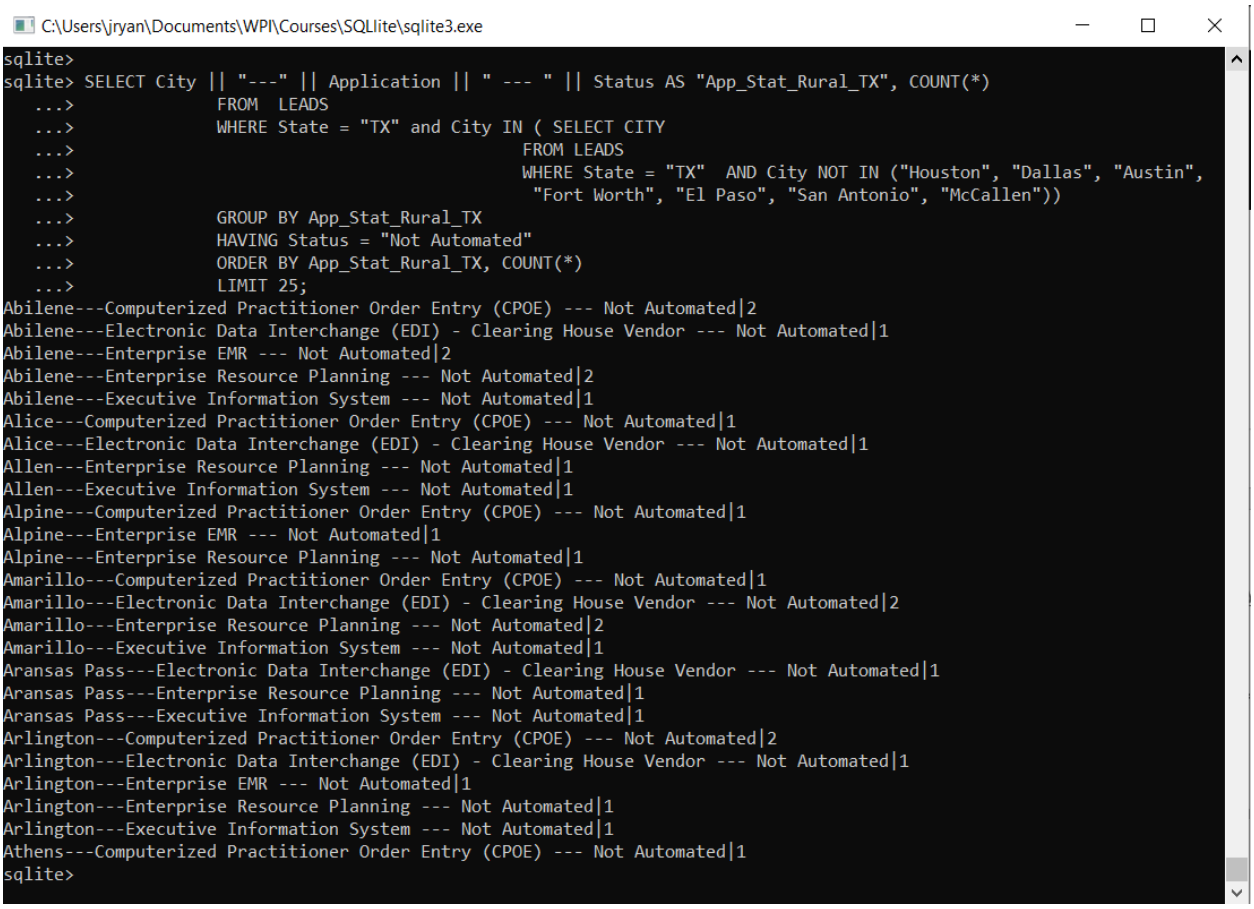
2b: A Type II subquery (2 points) nested with two inner queries (2 points).

```
SELECT City || "---" || Application || " --- " AS "App_Stat_Top_Urban_TX", Status, COUNT(*)
        FROM  ( SELECT Name, Application, Status, City, State
                        FROM ( SELECT Name, Application, Status, City, State
                                FROM LEADS
                                WHERE State = "TX")
                        WHERE City IN ("Houston", "Dallas", "Austin", "Fort Worth", "El Paso",
                                "San Antonio", "McCallen"))
         GROUP BY App_Stat_Top_Urban_TX, Status
         HAVING Status = "Not Automated"
        ORDER BY App_Stat_Top_Urban_TX, COUNT(*);
```



2c: A Type III correlated subquery (2 points).

```
SELECT City || "---" || Application || " --- " || Status AS "App_Stat_Rural_TX", COUNT(*)
        FROM  LEADS
        WHERE State = "TX" and City IN ( SELECT CITY
                        FROM LEADS
                        WHERE State = "TX"  AND City NOT IN ("Houston", "Dallas", "Austin",
                         "Fort Worth", "El Paso", "San Antonio", "McCallen"))
        GROUP BY App_Stat_Rural_TX
        HAVING Status = "Not Automated"
        ORDER BY App_Stat_Rural_TX, COUNT(*)
        LIMIT 25;
```



Remove the LIMIT 25 clause from the SQLite script to see the complete rural hospital listing.

The correlated subquery above is an example of not using EXISTS.

2d: A SELECT projection from a table created by a SELECT statement (2 points) with 5 columns.

```
CREATE TABLE TX_DALLAS_HOSPITALS AS
SELECT *
        FROM LEADS
        WHERE State = “TX” AND City = “Dallas”;

SELECT Entity_No, Name, Zip, Application || “---” || Status As AppStat
        FROM TX_DALLAS_HOSPITALS
        WHERE Status = “Live and Operational”
        ORDER BY ZIP
        LIMIT 15;
```

```
C:\Users\jryan\Documents\WPI\Courses\SQLite\sqlite3.exe
sqlite>
sqlite> CREATE TABLE TX_DALLAS_HOSPITALS AS
   ...> SELECT *
   ...>               FROM LEADS
   ...>               WHERE State = "TX" AND City = "Dallas";
sqlite>
sqlite> .tables
LEADS               TX_DALLAS_HOSPITALS
sqlite>
sqlite>
sqlite> SELECT Entity_No, Name, Zip, Application || "---" || Status As AppStat
   ...>               FROM TX_DALLAS_HOSPITALS
   ...>               WHERE Status = "Live and Operational"
   ...>               ORDER BY ZIP
   ...>               LIMIT 15;
1223|Methodist Dallas Medical Center|75203|Enterprise EMR---Live and Operational
1223|Methodist Dallas Medical Center|75203|Executive Information System---Live and Operational
10483|Doctors Hospital of Dallas|75218|Enterprise EMR---Live and Operational
10483|Doctors Hospital of Dallas|75218|Executive Information System---Live and Operational
10002599|Kindred Hospital - White Rock|75218|Electronic Data Interchange (EDI) - Clearing House Vendor---Live and Operat
ional
10002599|Kindred Hospital - White Rock|75218|Enterprise Resource Planning---Live and Operational
9292|Medical City Dallas Hospital|75230|Electronic Data Interchange (EDI) - Clearing House Vendor---Live and Operational

9292|Medical City Dallas Hospital|75230|Enterprise EMR---Live and Operational
9292|Medical City Dallas Hospital|75230|Enterprise Resource Planning---Live and Operational
9292|Medical City Dallas Hospital|75230|Executive Information System---Live and Operational
3565|Presbyterian Hospital of Dallas|75231|Electronic Data Interchange (EDI) - Clearing House Vendor---Live and Operatio
nal
100059507|Kindred Hospital - Dallas - Walnut Hill|75231|Electronic Data Interchange (EDI) - Clearing House Vendor---Live
 and Operational
100059507|Kindred Hospital - Dallas - Walnut Hill|75231|Enterprise Resource Planning---Live and Operational
10488|RHD Memorial Medical Center|75234|Electronic Data Interchange (EDI) - Clearing House Vendor---Live and Operational

10488|RHD Memorial Medical Center|75234|Enterprise EMR---Live and Operational
sqlite>
```

2e: The SELECT projection from tables saved to a CSV file (2 points).

```
.headers on
.mode csv
.output tut_four.csv
SELECT Entity_No, Name, Zip, Application || “---” || Status As AppStat
        FROM TX_DALLAS_HOSPITALS
        WHERE Status = “Live and Operational”
        ORDER BY ZIP;
```