

Tutorial 4

HMIS:

First created HMIS db from file

```
sqlite> .tables
sqlite> .read CreateDBHMIS.sql

.mode list
.separator " | "

.output HMIS_out2.txt
.echo OFF
sqlite>
```

Subquery 1 :

```
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> 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

```

Subquery 2:

```

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(*);

```

```

sqlite> 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(*);
Austin---Computerized Practitioner Order Entry (CPOE) --- | Not Automated | 4
Austin---Electronic Data Interchange (EDI) - Clearing House Vendor --- | Not Automated | 4
Austin---Enterprise EMR --- | Not Automated | 5
Austin---Enterprise Resource Planning --- | Not Automated | 3
Austin---Executive Information System --- | Not Automated | 3
Dallas---Computerized Practitioner Order Entry (CPOE) --- | Not Automated | 10
Dallas---Electronic Data Interchange (EDI) - Clearing House Vendor --- | Not Automated | 9
Dallas---Enterprise EMR --- | Not Automated | 7
Dallas---Enterprise Resource Planning --- | Not Automated | 10
Dallas---Executive Information System --- | Not Automated | 7
El Paso---Computerized Practitioner Order Entry (CPOE) --- | Not Automated | 6
El Paso---Electronic Data Interchange (EDI) - Clearing House Vendor --- | Not Automated | 1
El Paso---Enterprise EMR --- | Not Automated | 3
El Paso---Enterprise Resource Planning --- | Not Automated | 4
El Paso---Executive Information System --- | Not Automated | 1
Fort Worth---Computerized Practitioner Order Entry (CPOE) --- | Not Automated | 6
Fort Worth---Electronic Data Interchange (EDI) - Clearing House Vendor --- | Not Automated | 8
Fort Worth---Enterprise EMR --- | Not Automated | 6
Fort Worth---Enterprise Resource Planning --- | Not Automated | 4
Fort Worth---Executive Information System --- | Not Automated | 2
Houston---Computerized Practitioner Order Entry (CPOE) --- | Not Automated | 18

```

Subquery 3 :

```

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 50;

```

```

sqlite> 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 50;
Abilene---Computerized Practitioner Order Entry (CPOE) --- Not Automated | 2
Abilene---Electronic Data Interchange (EDI) - Clearing House Vendor --- Not Automated | 1
Abilene---Enterprise EMR --- Not Automated | 2
Abilene---Enterprise Resource Planning --- Not Automated | 2
Abilene---Executive Information System --- Not Automated | 1
Alice---Computerized Practitioner Order Entry (CPOE) --- Not Automated | 1
Alice---Electronic Data Interchange (EDI) - Clearing House Vendor --- Not Automated | 1
Allen---Enterprise Resource Planning --- Not Automated | 1
Allen---Executive Information System --- Not Automated | 1
Alpine---Computerized Practitioner Order Entry (CPOE) --- Not Automated | 1
Alpine---Enterprise EMR --- Not Automated | 1
Alpine---Enterprise Resource Planning --- Not Automated | 1
Amarillo---Computerized Practitioner Order Entry (CPOE) --- Not Automated | 1
Amarillo---Electronic Data Interchange (EDI) - Clearing House Vendor --- Not Automated | 2
Amarillo---Enterprise Resource Planning --- Not Automated | 2
Amarillo---Executive Information System --- Not Automated | 1
Aransas Pass---Electronic Data Interchange (EDI) - Clearing House Vendor --- Not Automated | 1
Aransas Pass---Enterprise Resource Planning --- Not Automated | 1
Aransas Pass---Executive Information System --- Not Automated | 1
Arlington---Computerized Practitioner Order Entry (CPOE) --- Not Automated | 2
Arlington---Electronic Data Interchange (EDI) - Clearing House Vendor --- Not Automated | 1
Arlington---Enterprise EMR --- Not Automated | 1
Arlington---Enterprise Resource Planning --- Not Automated | 1

```

Select projection :

```

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 10;

```

```

sqlite> CREATE TABLE TX_DALLAS_HOSPITALS AS
...> SELECT *
...> FROM LEADS
...> WHERE State = "TX" AND City = "Dallas";
sqlite> SELECT Entity_No, Name, Zip, Application || "----" || Status As AppStat
...> FROM TX_DALLAS_HOSPITALS
...> WHERE Status = "Live and Operational"
...> ORDER BY ZIP
...> LIMIT 10;
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 Operational
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
sqlite>

```

Select projection to create csv file:

.headers on

.mode csv

.output file_one.csv

SELECT Entity_No, Name, Zip, Application || "----" || Status As AppStat

FROM TX_DALLAS_HOSPITALS

WHERE Status = "Live and Operational"

ORDER BY ZIP;

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

file_one				
Entity_No	Name	Zip	AppStat	
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 Operational"	
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 Operational"	
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"	
10488	"RHD Memorial Medical Center"	75234	"Executive Information System---Live and Operational"	
8018	"UT Southwestern University Hospitals - St. Paul Building"	75235	"Executive Information System---Live and Operational"	
22717	"UT Southwestern University Hospitals - Zale Lipshy Building"	75235	"Executive Information System---Live and Operational"	
25922	"Parkland Memorial Hospital"	75235	"Executive Information System---Live and Operational"	
31905	"Children Medical Center of Dallas"	75235	"Electronic Data Interchange (EDI) - Clearing House Vendor---Live and Operational"	
31905	"Children Medical Center of Dallas"	75235	"Enterprise Resource Planning---Live and Operational"	
31905	"Children Medical Center of Dallas"	75235	"Executive Information System---Live and Operational"	
100043475	"LifeCare Hospitals of Dallas"	75235	"Executive Information System---Live and Operational"	
100058455	"Texas Specialty Hospital - Dallas"	75235	"Computerized Practitioner Order Entry (CPOE)---Live and Operational"	
100058455	"Texas Specialty Hospital - Dallas"	75235	"Enterprise Resource Planning---Live and Operational"	
1221	"Methodist Charlton Medical Center"	75237	"Enterprise EMR---Live and Operational"	
1221	"Methodist Charlton Medical Center"	75237	"Executive Information System---Live and Operational"	
5773	"Kindred Hospital - Dallas"	75243	"Electronic Data Interchange (EDI) - Clearing House Vendor---Live and Operational"	
5773	"Kindred Hospital - Dallas"	75243	"Enterprise Resource Planning---Live and Operational"	
110	"Baylor University Medical Center"	75246	"Electronic Data Interchange (EDI) - Clearing House Vendor---Live and Operational"	
110	"Baylor University Medical Center"	75246	"Enterprise EMR---Live and Operational"	
110	"Baylor University Medical Center"	75246	"Enterprise Resource Planning---Live and Operational"	

Sales_Co_DW:

Created db from reading file:

```
C:\SQLite\SQL\sqlite3.exe x + v
SQLite version 3.40.1 2022-12-28 14:03:47
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .tables
sqlite> .read Create_Sale_Co_DW.sql

.mode list
.separator " | "

.output Sales_out.txt
.echo OFF
sqlite> .tables
DWCUSTOMER  DWPRODUCT  DWREGION    DWSALESFACT  DWTIME      DWVENDOR
sqlite>
```

Subquery 1:

```
SELECT V_STATE || '--' || TRIM(V_NAME) || '--' || TRIM(P_DESCRIPT) AS 'State-Vendor-Product',
V_STATE, SUM(DWSALESFACT.SALE_UNITS), COUNT(DWSALESFACT.SALE_UNITS)
FROM DWVENDOR, DWPRODUCT, DWSALESFACT
WHERE DWVENDOR.V_CODE = DWPRODUCT.V_CODE AND DWPRODUCT.P_CODE =
DWSALESFACT.P_CODE AND DWVENDOR.V_CODE IN (SELECT V_CODE
FROM DWVENDOR WHERE V_STATE IN
(SELECT V_STATE FROM DWREGION, DWCUSTOMER, DWVENDOR, DWPRODUCT,
DWSALESFACT
WHERE DWREGION.REG_NAME = 'SE' AND
DWREGION.REG_ID = DWCUSTOMER.REG_ID AND
DWCUSTOMER.CUS_CODE = DWSALESFACT.CUS_CODE AND
DWSALESFACT.P_CODE = DWPRODUCT.P_CODE))
GROUP BY V_STATE, 'State-Vendor-Product'
ORDER BY V_STATE;
```

```

sqlite> SELECT V_STATE || '--' || TRIM(V_NAME) || '--' || TRIM(P_DESCRIPT) AS 'State-Vendor-Product',
...> V_STATE, SUM(DWSALESFACT.SALE_UNITS), COUNT(DWSALESFACT.SALE_UNITS)
...> FROM DWVENDOR, DWPRODUCT, DWSALESFACT
...> WHERE DWVENDOR.V_CODE = DWPRODUCT.V_CODE AND DWPRODUCT.P_CODE =
...> DWSALESFACT.P_CODE AND DWVENDOR.V_CODE IN (SELECT V_CODE
...> FROM DWVENDOR WHERE V_STATE IN
...> (SELECT V_STATE FROM DWREGION, DWCUSTOMER, DWVENDOR, DWPRODUCT,
...> DWSALESFACT
...> WHERE DWREGION.REG_NAME = 'SE' AND
...> DWREGION.REG_ID = DWCUSTOMER.REG_ID AND
...> DWCUSTOMER.CUS_CODE = DWSALESFACT.CUS_CODE AND
...> DWSALESFACT.P_CODE = DWPRODUCT.P_CODE))
...> GROUP BY V_STATE, 'State-Vendor-Product'
...> ORDER BY V_STATE;
FL--Rubicon Systems--B\&D cordless drill, 1/2-in. | FL | 8 | 4
GA--Randsets Ltd.--Hrd. cloth, 1/4-in., 2x50 | GA | 2 | 2
KY--Gomez Bros.--7.25-in. pwr. saw blade | KY | 28 | 12
TN--Bryson, Inc.--Claw hammer | TN | 54 | 18
sqlite>

```

Subquery 2 :

```

SELECT V_STATE || '--' || TRIM(V_NAME) || '--' ||
TRIM(P_DESCRIPT) AS 'State-Vendor-Product', V_STATE, SUM(DWSALESFACT.SALE_UNITS),
COUNT(DWSALESFACT.SALE_UNITS)
FROM DWVENDOR, DWPRODUCT, DWSALESFACT
WHERE DWVENDOR.V_CODE = DWPRODUCT.V_CODE AND
DWPRODUCT.P_CODE = DWSALESFACT.P_CODE AND
DWVENDOR.V_CODE = (SELECT V_CODE FROM DWVENDOR
WHERE V_STATE = (SELECT V_STATE FROM DWREGION, DWCUSTOMER,
DWVENDOR, DWPRODUCT, DWSALESFACT
WHERE DWREGION.REG_NAME = 'SE' AND
DWREGION.REG_ID = DWCUSTOMER.REG_ID AND
DWCUSTOMER.CUS_CODE = DWSALESFACT.CUS_CODE AND
DWSALESFACT.P_CODE = DWPRODUCT.P_CODE))
GROUP BY V_STATE, 'State-Vendor-Product'
ORDER BY V_STATE;

```

```

sqlite> SELECT V_STATE || '--' || TRIM(V_NAME) || '--' ||
...> TRIM(P_DESCRIPT) AS 'State-Vendor-Product', V_STATE, SUM(DWSALESFACT.SALE_UNITS),
...> COUNT(DWSALESFACT.SALE_UNITS)
...> FROM DWVENDOR, DWPRODUCT, DWSALESFACT
...> WHERE DWVENDOR.V_CODE = DWPRODUCT.V_CODE AND
...> DWPRODUCT.P_CODE = DWSALESFACT.P_CODE AND
...> DWVENDOR.V_CODE = (SELECT V_CODE FROM DWVENDOR
...> WHERE V_STATE = (SELECT V_STATE FROM DWREGION, DWCUSTOMER,
...> DWVENDOR, DWPRODUCT, DWSALESFACT
...> WHERE DWREGION.REG_NAME = 'SE' AND
...> DWREGION.REG_ID = DWCUSTOMER.REG_ID AND
...> DWCUSTOMER.CUS_CODE = DWSALESFACT.CUS_CODE AND
...> DWSALESFACT.P_CODE = DWPRODUCT.P_CODE))
...> GROUP BY V_STATE, 'State-Vendor-Product'
...> ORDER BY V_STATE;
TN--Bryson, Inc.--Claw hammer | TN | 50 | 14
sqlite>

```

Subquery 3:

```

SELECT DWVENDOR.V_STATE || '--' || TRIM(DWVENDOR.V_NAME) || '--' ||
TRIM(P_DESCRIPT) AS StateVendorProduct, DWVENDOR.V_STATE,
SUM(DWSALESFACT.SALE_UNITS), COUNT(DWSALESFACT.SALE_UNITS)
FROM DWVENDOR, DWPRODUCT, DWSALESFACT
WHERE DWVENDOR.V_CODE = DWPRODUCT.V_CODE AND
DWPRODUCT.P_CODE = DWSALESFACT.P_CODE AND
DWPRODUCT.V_CODE IN
( SELECT DWPRODUCT.V_CODE FROM DWREGION, DWVENDOR,
DWCUSTOMER, DWVENDOR, DWPRODUCT, DWSALESFACT
WHERE DWREGION.REG_NAME = 'SE' AND
DWREGION.REG_ID = DWCUSTOMER.REG_ID AND
DWCUSTOMER.CUS_CODE = DWSALESFACT.CUS_CODE AND
DWSALESFACT.P_CODE = DWPRODUCT.P_CODE)
GROUP BY V_STATE, StateVendorProduct
ORDER BY V_STATE;

```



```

sqlite> SELECT DWVENDOR.V_STATE || '--' || TRIM(DWVENDOR.V_NAME) || '--' ||
...> TRIM(P_DESCRIPT) AS StateVendorProduct, DWVENDOR.V_STATE,
...> SUM(DWSALESFACT.SALE_UNITS), COUNT(DWSALESFACT.SALE_UNITS)
...> FROM DWVENDOR, DWPRODUCT, DWSALESFACT
...> WHERE DWVENDOR.V_CODE = DWPRODUCT.V_CODE AND
...> DWPRODUCT.P_CODE = DWSALESFACT.P_CODE AND
...> DWPRODUCT.V_CODE IN
...> ( SELECT DWPRODUCT.V_CODE FROM DWREGION, DWVENDOR,
...> DWCUSTOMER, DWVENDOR, DWPRODUCT, DWSALESFACT
...> WHERE DWREGION.REG_NAME = 'SE' AND
...> DWREGION.REG_ID = DWCUSTOMER.REG_ID AND
...> DWCUSTOMER.CUS_CODE = DWSALESFACT.CUS_CODE AND
...> DWSALESFACT.P_CODE = DWPRODUCT.P_CODE)
...> GROUP BY V_STATE, StateVendorProduct
...> ORDER BY V_STATE;
FL--Rubicon Systems--B\&D cordless drill, 1/2-in. | FL | 2 | 2
FL--Rubicon Systems--Steel matting, 4'x8'x1/6", .5" mesh | FL | 6 | 2
GA--Randsets Ltd.--Hrd. cloth, 1/4-in., 2x50 | GA | 2 | 2
KY--Gomez Bros.--7.25-in. pwr. saw blade | KY | 16 | 6
KY--Gomez Bros.--Rat-tail file, 1/8-in. fine | KY | 12 | 6
TN--Bryson, Inc.--1.25-in. metal screw, 25 | TN | 6 | 2
TN--Bryson, Inc.--Claw hammer | TN | 10 | 8
TN--Bryson, Inc.--PVC pipe, 3.5-in., 8-ft | TN | 34 | 4
TN--ORDVA, Inc.--B\&D jigsaw, 12-in. blade | TN | 2 | 2
TN--ORDVA, Inc.--Hicut chain saw, 16 in. | TN | 2 | 2
sqlite>

```

Select projection :

```

SELECT DISTINCT DWPRODUCT.P_CODE AS P_CODE, DWPRODUCT.P_DESCRIPT AS P_DESC,
DWPRODUCT.P_CATEGORY AS P_CAT, DWPRODUCT.V_CODE AS V_CODE,
DWVENDOR.V_Name AS P_Vendor, DWSALESFACT.SALE_UNITS AS UNITS,
DWSALESFACT.SALE_PRICE AS PRICE
FROM DWVENDOR, DWPRODUCT, DWSALESFACT
WHERE DWVENDOR.V_CODE = DWPRODUCT.V_CODE AND
DWPRODUCT.P_CODE = DWSALESFACT.P_CODE;

```

```

sqlite> SELECT DISTINCT DWPRODUCT.P_CODE AS P_CODE, DWPRODUCT.P_DESCRIPT AS P_DESC, DWPRODUCT.P_CATEGORY AS P_CAT, DWPRO
DUCT.V_CODE AS V_CODE,
...> DWVENDOR.V_Name AS P_Vendor, DWSALESFACT.SALE_UNITS AS UNITS, DWSALESFACT.SALE_PRICE AS PRICE
...> FROM DWVENDOR, DWPRODUCT, DWSALESFACT
...> WHERE DWVENDOR.V_CODE = DWPRODUCT.V_CODE AND
...> DWPRODUCT.P_CODE = DWSALESFACT.P_CODE;
13-Q2/P2 | 7.25-in. pwr. saw blade | CAT1 | 21344 | Gomez Bros. | 1 | 14.99
23109-HB | Claw hammer | CAT4 | 21225 | Bryson, Inc. | 1 | 9.95
54778-2T | Rat-tail file, 1/8-in. fine | CAT1 | 21344 | Gomez Bros. | 2 | 4.99
2238/QPD | B\&D cordless drill, 1/2-in. | CAT3 | 25595 | Rubicon Systems | 1 | 38.95
1546-QQ2 | Hrd. cloth, 1/4-in., 2x50 | CAT2 | 23119 | Randsets Ltd. | 1 | 39.95
13-Q2/P2 | 7.25-in. pwr. saw blade | CAT1 | 21344 | Gomez Bros. | 5 | 14.99
54778-2T | Rat-tail file, 1/8-in. fine | CAT1 | 21344 | Gomez Bros. | 3 | 4.99
23109-HB | Claw hammer | CAT4 | 21225 | Bryson, Inc. | 2 | 9.95
PVC23DRT | PVC pipe, 3.5-in., 8-ft | CAT3 | 21225 | Bryson, Inc. | 12 | 5.87
SM-18277 | 1.25-in. metal screw, 25 | CAT4 | 21225 | Bryson, Inc. | 3 | 6.99
2232/PTY | B\&D jigsaw, 12-in. blade | CAT2 | 24288 | ORDVA, Inc. | 1 | 109.92
89-WRE-Q | Hicut chain saw, 16 in. | CAT2 | 24288 | ORDVA, Inc. | 1 | 256.99
13-Q2/P2 | 7.25-in. pwr. saw blade | CAT1 | 21344 | Gomez Bros. | 2 | 14.99
54778-2T | Rat-tail file, 1/8-in. fine | CAT1 | 21344 | Gomez Bros. | 1 | 4.99
PVC23DRT | PVC pipe, 3.5-in., 8-ft | CAT3 | 21225 | Bryson, Inc. | 5 | 5.87
WR3/TT3 | Steel matting, 4'x8'x1/6", .5" mesh | CAT3 | 25595 | Rubicon Systems | 3 | 119.95

```

Select projection to create .csv file :

.headers on

.mode csv

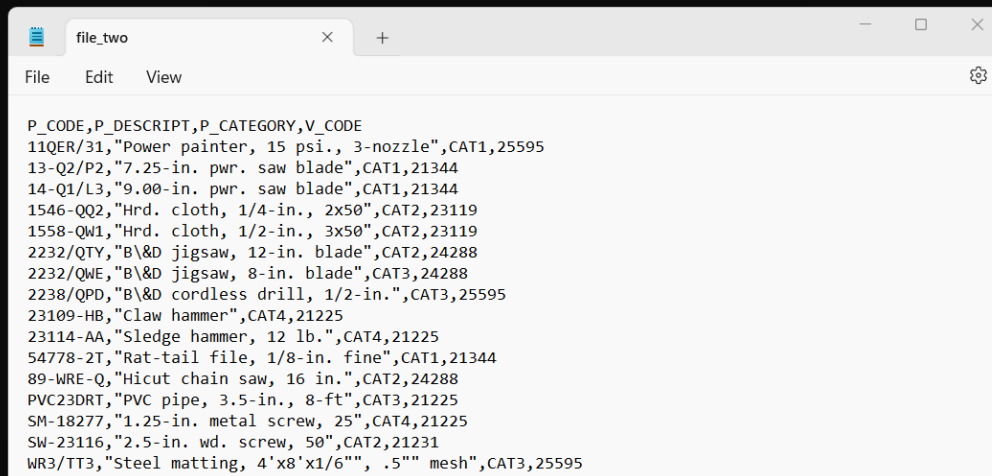
.output file_two.csv

SELECT *

FROM DWPRODUCT

ORDER BY DWPRODUCT.P_CODE, DWPRODUCT.P_DESCRIPTION;

```
sqlite> .headers on
sqlite> .mode csv
sqlite> .output file_two.csv
sqlite> SELECT *
...> FROM DWPRODUCT
...> ORDER BY DWPRODUCT.P_CODE, DWPRODUCT.P_DESCRIPTION;
sqlite>
```



P_CODE	P_DESCRIPTION	P_CATEGORY	V_CODE
11QER/31	"Power painter, 15 psi., 3-nozzle"	CAT1	25595
13-Q2/P2	"7.25-in. pwr. saw blade"	CAT1	21344
14-Q1/L3	"9.00-in. pwr. saw blade"	CAT1	21344
1546-QQ2	"Hrd. cloth, 1/4-in., 2x50"	CAT2	23119
1558-QW1	"Hrd. cloth, 1/2-in., 3x50"	CAT2	23119
2232/QT	"B&D jigsaw, 12-in. blade"	CAT2	24288
2232/QWE	"B&D jigsaw, 8-in. blade"	CAT3	24288
2238/QPD	"B&D cordless drill, 1/2-in."	CAT3	25595
23109-HB	"Claw hammer"	CAT4	21225
23114-AA	"Sledge hammer, 12 lb."	CAT4	21225
54778-2T	"Rat-tail file, 1/8-in. fine"	CAT1	21344
89-WRE-Q	"Hicut chain saw, 16 in."	CAT2	24288
PVC23DRT	"PVC pipe, 3.5-in., 8-ft"	CAT3	21225
SM-18277	"1.25-in. metal screw, 25"	CAT4	21225
SW-23116	"2.5-in. wd. screw, 50"	CAT2	21231
WR3/TT3	"Steel matting, 4'x8'x1/6\"", .5" mesh"	CAT3	25595

QC_Checks_db:

.open QC_Checks.db

.headers on

.separator " | "

```
sqlite> .open QC_Checks.db
sqlite> .headers on
sqlite> .separator " | "
```

Subquery 1:

```
SELECT Check_ID || " --- " || Staff || "—" AS ERR, COUNT(*) FROM ERRORS
WHERE Staff IN (SELECT Staff FROM ERRORS
WHERE Check_ID IN (SELECT Check_ID FROM ERRORS
GROUP BY Check_ID
ORDER BY COUNT(*) DESC
Limit 5)
GROUP BY Staff
ORDER BY COUNT(*) DESC
LIMIT 5)
GROUP BY ERR
ORDER BY ERR, COUNT(*);
```

```
^--- error here
sqlite> SELECT Check_ID || " --- " || Staff || "—" AS ERR, COUNT(*) FROM ERRORS
...> WHERE Staff IN (SELECT Staff FROM ERRORS
...> WHERE Check_ID IN (SELECT Check_ID FROM ERRORS
...> GROUP BY Check_ID
...> ORDER BY COUNT(*) DESC
...> Limit 5)
...> GROUP BY Staff
...> ORDER BY COUNT(*) DESC
...> LIMIT 5)
...> GROUP BY ERR
...> ORDER BY ERR, COUNT(*);
```

ERR	COUNT(*)
101 --- -	5
102 --- -	1
105 --- -	2
106 --- -	1
2001 --- -	1
2001 --- RN-42-	1
2002 --- -	1
2006 --- -	1
2008 --- -	6
3001 --- -	17
3002 --- RN-120-	1
3002 --- RN-42-	1
3003 --- RN-120-	8
3003 --- RN-136-	12
3003 --- RN-42-	9
3003 --- RN-6-	9
3005 --- RN-6-	1

Subquery2:

```
SELECT Check_ID || " — " || SUBSTR(QC_Note,1,70) || " --> " AS ERR, COUNT(*) FROM ERRORS
WHERE Check_ID IN (SELECT Check_ID FROM ERRORS
WHERE Staff == (SELECT Staff FROM ERRORS
```

```

GROUP BY Staff
ORDER BY COUNT(*) DESC
Limit 1)
GROUP BY Check_ID
ORDER BY COUNT(*) DESC)
GROUP BY ERR
HAVING COUNT(*) > 2
ORDER BY COUNT(*) DESC , ERR ASC;

```

```

sqlite> SELECT Check_ID || " - " || SUBSTR(QC_Note,1,70) || " --> " AS ERR, COUNT(*) FROM ERRORS
...> WHERE Check_ID IN (SELECT Check_ID FROM ERRORS
...> WHERE Staff == (SELECT Staff FROM ERRORS
...> GROUP BY Staff
...> ORDER BY COUNT(*) DESC
...> Limit 1)
...> GROUP BY Check_ID
...> ORDER BY COUNT(*) DESC)
...> GROUP BY ERR
...> HAVING COUNT(*) > 2
...> ORDER BY COUNT(*) DESC , ERR ASC;
ERR|COUNT(*)
106 - If the PACU In time () is not complete then the location the patient i --> |40
2008 - The total surgery minutes (1) must be greater than or equal to one min --> |16
101 - The Surgery Complete Indicator is set to 0 and must be set to 1 to pas --> |5
102 - If the Prep Hold In time is not null then the Nurse Assigned - Nurse --> |5
2008 - The total surgery minutes (0) must be greater than or equal to one min --> |5
3001 - The PACU Surgeon(Partington - Erin Jayne MD) must be the same as the --> |4
sqlite>

```

Subquery 3 :

```

SELECT Check_ID || " --> " AS ERR, COUNT(*) FROM ERRORS
WHERE Check_ID NOT IN (SELECT Check_ID FROM ERRORS
WHERE Staff == (SELECT Staff FROM ERRORS
GROUP BY Staff
ORDER BY COUNT(*) DESC
Limit 1)
GROUP BY Check_ID
ORDER BY COUNT(*) DESC)
GROUP BY ERR
ORDER BY COUNT(*) DESC ;

```

```
C:\SQLite\SQL\sqlite3.exe
sqlite> SELECT Check_ID || " -> " AS ERR, COUNT(*) FROM ERRORS
...> WHERE Check_ID NOT IN (SELECT Check_ID FROM ERRORS
...> WHERE Staff == (SELECT Staff FROM ERRORS
...> GROUP BY Staff
...> ORDER BY COUNT(*) DESC
...> Limit 1)
...> GROUP BY Check_ID
...> ORDER BY COUNT(*) DESC)
...> GROUP BY ERR
...> ORDER BY COUNT(*) DESC ;
ERR|COUNT(*)
3003 -> |60
3005 -> |19
1003 -> |12
3002 -> |11
1001 -> |11
3006 -> |8
2004 -> |5
1002 -> |4
3004 -> |1
104 -> |1
103 -> |1
1004 -> |1
sqlite> |
```

Select projection :

```
CREATE TABLE QC_Check_Errs AS
SELECT Check_ID, Staff, COUNT(*) FROM ERRORS
GROUP BY Check_ID, Staff
HAVING COUNT(*) > 1
ORDER BY COUNT(*) DESC ;
SELECT * FROM QC_Check_Errs;
```

```

1004 -> |1
sqlite> CREATE TABLE QC_Check_Errs AS
...> SELECT Check_ID, Staff, COUNT(*) FROM ERRORS
...> GROUP BY Check_ID, Staff
...> HAVING COUNT(*) > 1
...> ORDER BY COUNT(*) DESC ;
sqlite> SELECT * FROM QC_Check_Errs;
Check_ID|Staff|COUNT(*)
3001||17
3003|RN-136|12
3003|RN-42|9
3003|RN-6|9
5002||9
3003|RN-120|8
106|RN-119|7
106|RN-68|6
2008||6
101||5
3006|RN-87|5
106|RN-93|4
1003|RN-118|4
3003|RN-108|4
5001||4
2001|RN-132|3
2008|RN-134|3
3005|RN-7|3

```

Select projection to create csv file :

```

.headers on
.mode csv
.output file_three.csv
SELECT Check_ID, COUNT(*)
FROM ERRORS
GROUP BY Check_ID
ORDER BY COUNT(*) DESC;

```

C:\SQLite\SQL\sqlite3.exe

```
sqlite> .headers on
sqlite> .mode csv
sqlite> .output file_three.csv
sqlite> SELECT Check_ID, COUNT(*)
...> FROM ERRORS
...> GROUP BY Check_ID
...> ORDER BY COUNT(*) DESC;
sqlite>
```

file_three

File Edit View

Check_ID	COUNT(*)
3003	60
2001	40
106	40
3001	29
2008	21
105	20
3005	19
5002	16
5001	16
1003	12
3002	11
1001	11
3006	8
2002	8
2006	6
2004	5
102	5