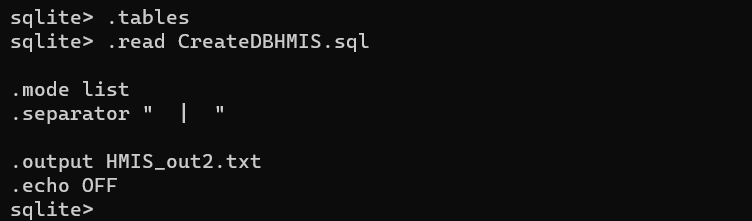
# Tutorial 4

# HMIS:

First created HMIS db from file



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(\*);

Text

Description automatically generated

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(\*);

Text

Description automatically generated

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;

Text

Description automatically generated

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;

Text

Description automatically generated

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;

Text

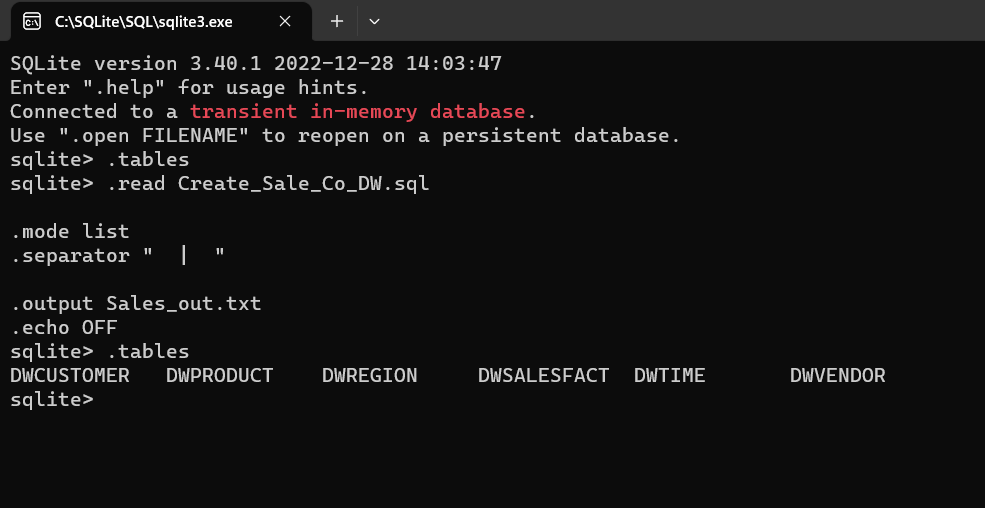
Description automatically generated

Text

Description automatically generated

# Sales\_Co\_DW:

Created db from reading file:



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;

Text

Description automatically generated

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;

Text

Description automatically generated

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;

Text

Description automatically generated

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;

Text

Description automatically generated

Select projection to create .csv file :

.headers on

.mode csv

.output file\_two.csv

SELECT \*

FROM DWPRODUCT

ORDER BY DWPRODUCT.P\_CODE, DWPRODUCT.P\_DESCRIPT;

Graphical user interface, text, application

Description automatically generated

# QC\_Checks\_db:

.open QC\_Checks.db

.headers on

.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(\*);

Text

Description automatically generated

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;

Text

Description automatically generated

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 ;

Text

Description automatically generated

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;

Text

Description automatically generated

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;

Graphical user interface, text, application

Description automatically generated