Tracy Ficor

DBST651 SPRING 2019

DROP TABLE ORDERS;

DROP TABLE EMPLOYEE;

DROP TABLE PRODUCT;

DROP TABLE ORDER\_LINE;

DROP TABLE CUSTOMER;

CREATE TABLE CUSTOMER

(CUSTNO VARCHAR2 (8 BYTE) NOT NULL,

CUSTFIRSTNAME VARCHAR2 (20 BYTE) NOT NULL,

CUSTLASTNAME VARCHAR2 (30 BYTE) NOT NULL,

CUSTSTREET VARCHAR2 (50 BYTE),

CUSTCITY VARCHAR2 (30 BYTE),

CUSTSTATE VARCHAR2 (2 BYTE),

CUSTZIP VARCHAR2 (10 BYTE),

CUSTBAL NUMBER(12,2),

CONSTRAINT PK\_CUSTOMER PRIMARY KEY (CUSTNO));

CREATE TABLE PRODUCT

(PRODNO VARCHAR2 (8 BYTE) NOT NULL,

PRODNAME VARCHAR2 (50 BYTE) NOT NULL,

PRODMFG VARCHAR2 (20 BYTE) NOT NULL,

PRODQOH NUMBER(\*,0),

PRODPRICE NUMBER (12,2),

PRODNEXTSHIPDATE DATE,

CONSTRAINT PK\_PRODNO PRIMARY KEY (PRODNO));

CREATE TABLE EMPLOYEE

(EMPNO VARCHAR2(8 BYTE) NOT NULL,

EMPFIRSTNAME VARCHAR2(20 BYTE) NOT NULL,

EMPLASTNAME VARCHAR2(30 BYTE) NOT NULL,

EMPPHONE VARCHAR2(15 BYTE) ,

EMPEMAIL VARCHAR2(50 BYTE) NOT NULL,

SUPEMPNO VARCHAR2(8 BYTE),

EMPCOMMRATE NUMBER(3,3),

CONSTRAINT PK\_EMPNO PRIMARY KEY (EMPNO),

CONSTRAINT UC\_EMPEMAIL UNIQUE (EMPEMAIL),

CONSTRAINT FK\_SUPEMPNO FOREIGN KEY (EMPNO) REFERENCES EMPLOYEE(EMPNO));

CREATE TABLE ORDERS

(ORDNO VARCHAR2 (8 BYTE) NOT NULL,

ORDDATE DATE NOT NULL,

CUSTNO VARCHAR2 (8 BYTE) NOT NULL,

EMPNO VARCHAR2 (8 BYTE),

ORDNAME VARCHAR2 (50 BYTE),

ORDSTREET VARCHAR2 (50 BYTE),

ORDCITY VARCHAR2 (30 BYTE),

ORDSTATE VARCHAR2 (2 BYTE),

ORDZIP VARCHAR2 (10 BYTE),

CONSTRAINT PK\_ORDER\_NUM PRIMARY KEY(ORDNO),

CONSTRAINT FK\_CUSTNO FOREIGN KEY (CUSTNO) REFERENCES CUSTOMER(CUSTNO),

CONSTRAINT FK\_EMPNO FOREIGN KEY (EMPNO) REFERENCES EMPLOYEE(EMPNO));

CREATE TABLE ORDER\_LINE

(ORDNO VARCHAR2 (8 BYTE) NOT NULL,

PRODNO VARCHAR2 (8 BYTE) NOT NULL,

QTY NUMBER (\*,0),

CONSTRAINT PK\_oRDPRO PRIMARY KEY (ORDNO, PRODNO));

-- ADD DATA --

INSERT INTO CUSTOMER (CUSTLASTNAME, CUSTFIRSTNAME, CUSTNO, CUSTZIP)

VALUES ('SMITH', 'JOHN', '1', '23456');

SELECT \* FROM CUSTOMER;

COMMIT;

UPDATE CUSTOMER

SET CUSTZIP = '20001'

WHERE CUSTZIP = '23456' AND CUSTNO = '1';

SELECT \* FROM CUSTOMER;

ROLLBACK;

DELETE FROM CUSTOMER

WHERE CUSTNO = '1';

COMMIT'

--------PART II ----------------------

--Found difficulty with averaging overall for these queries.

SELECT CUSTOMER.CUSTLASTNAME,

CUSTOMER.CUSTSTATE,

CUSTOMER.CUSTCITY,

COUNT(ORDERS.ORDNO) AS Count\_ORDNO,

ORDER\_LINE.ORDNO,

ORDER\_LINE.PRODNO,

PRODUCT.PRODPRICE \* ORDER\_LINE.QTY AS "Order Total"

FROM PRODUCT,

CUSTOMER

INNER JOIN ORDERS

ON CUSTOMER.CUSTNO = ORDERS.CUSTNO,

ORDER\_LINE

WHERE CUSTOMER.CUSTSTATE LIKE 'WA%'

AND CUSTOMER.CUSTCITY LIKE 'Seattle%'

GROUP BY CUSTOMER.CUSTLASTNAME,

CUSTOMER.CUSTSTATE,

CUSTOMER.CUSTCITY,

ORDER\_LINE.ORDNO,

ORDER\_LINE.PRODNO,

PRODUCT.PRODPRICE \* ORDER\_LINE.QTY,

ORDER\_LINE.QTY,

PRODUCT.PRODPRICE;

--2B.

SELECT CUSTOMER.CUSTLASTNAME,

CUSTOMER.CUSTNO,

COUNT(DISTINCT PRODUCT.PRODNO) AS Count\_PRODNO

FROM ORDERS

INNER JOIN CUSTOMER

ON CUSTOMER.CUSTNO = ORDERS.CUSTNO,

PRODUCT

WHERE CUSTOMER.CUSTCITY LIKE 'Seattle%'

AND CUSTOMER.CUSTSTATE LIKE 'WA%'

GROUP BY CUSTOMER.CUSTLASTNAME,

CUSTOMER.CUSTNO

ORDER BY CUSTOMER.CUSTLASTNAME

-- 2C.

-- Still having troube obtaining averages

SELECT DISTINCT EMPLOYEE.EMPNO,

EMPLOYEE.EMPLASTNAME,

ORDERS.ORDNO,

SUM(ORDER\_LINE.QTY) AS Sum\_QTY,

ORDER\_LINE.PRODNO,

SUM(PRODUCT.PRODPRICE) AS Sum\_PRODPRICE

FROM EMPLOYEE

INNER JOIN ORDERS

ON EMPLOYEE.EMPNO = ORDERS.EMPNO,

ORDER\_LINE,

PRODUCT

WHERE EMPLOYEE.EMPCOMMRATE < '0.05'

GROUP BY EMPLOYEE.EMPNO,

EMPLOYEE.EMPLASTNAME,

ORDERS.ORDNO,

ORDER\_LINE.PRODNO

ORDER BY EMPLOYEE.EMPLASTNAME

-- 2D.

SELECT DISTINCT PRODUCT.PRODNAME,

COUNT(CUSTOMER.CUSTNO) AS Count\_CUSTNO,

PRODUCT.PRODNO

FROM ORDERS

INNER JOIN CUSTOMER

ON CUSTOMER.CUSTNO = ORDERS.CUSTNO,

PRODUCT

WHERE ORDERS.ORDDATE BETWEEN '01-JAN-2007' AND '31-JAN-2007'

GROUP BY PRODUCT.PRODNAME,

PRODUCT.PRODNO

ORDER BY PRODUCT.PRODNAME

------------------------------------------------------------------------------------------------------------------------------------------

Table ORDERS dropped.

Table EMPLOYEE dropped.

Table PRODUCT dropped.

Table ORDER\_LINE dropped.

Table CUSTOMER dropped.

Table CUSTOMER created.

Table PRODUCT created.

Table EMPLOYEE created.

Table ORDERS created.

Table ORDER\_LINE created.

1 row deleted.

Commit complete.

IV.

1. 1 row inserted.



1. Commit complete.
2. 1 row inserted.



d. Rollback complete.



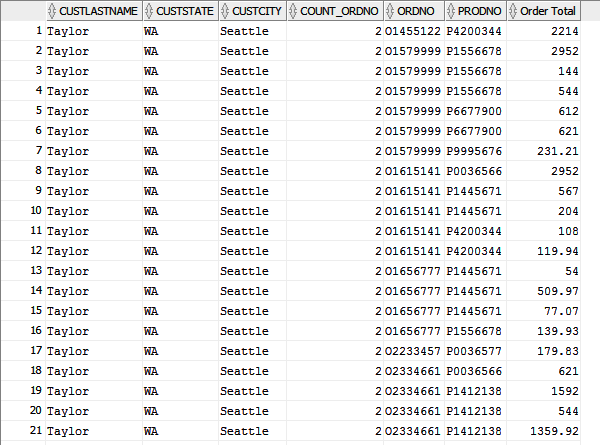
e. 1 row deleted.



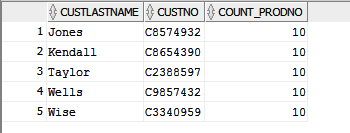
f. Commit complete.

QUESTION 2

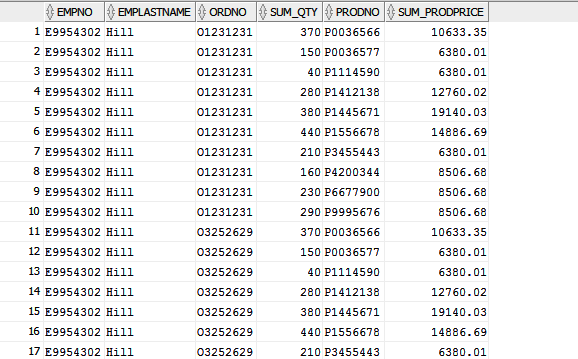
1. Could not manage to identify the way to average for this particular question.



b.



c. Still was unsure how to construct the average



d.

