CS 532 Database System Project 1

Team Members:

Name: Akshay Kumar Anvekar
B No: B00813848
aanveka2@binghamton.edu

Name: Sanjay Bettadapura Ganesha
B No: B00816568
sbettad1@binghamton.edu

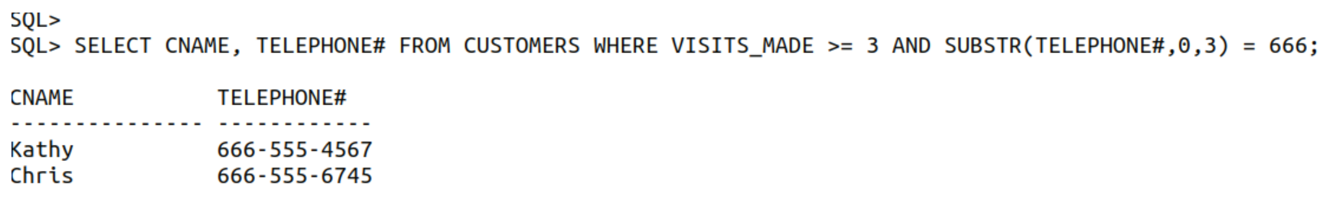
Name: Akshay Kumar Anvekar
B No: B00813848

Name: Sanjay Bettadapura Ganesha
B No: B00816568

1. Find the names and telephone#s of those customers who have visited the retail business at least 3 times and whose telephone# has an area code 666.

Query:

SELECT CNAME, TELEPHONE# FROM CUSTOMERS WHERE VISITS_MADE >= 3 AND SUBSTR(TELEPHONE#,0,3) = 666;

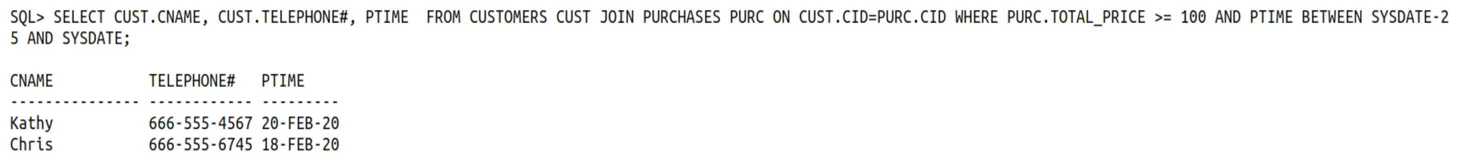


```
SQL>
SQL> SELECT CNAME, TELEPHONE# FROM CUSTOMERS WHERE VISITS_MADE >= 3 AND SUBSTR(TELEPHONE#,0,3) = 666;

CNAME           TELEPHONE#
--------------- ------------
Kathy           666-555-4567
Chris           666-555-6745
```

2. Find the names and telephone#s of those customers who made at least one purchase with a total price of at least 100 dollars in the last 25 days (from the day your query is issued).

Query:

SELECT CUST.CNAME, CUST.TELEPHONE#, PTIME FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID=PURC.CID WHERE PURC.TOTAL_PRICE >= 100 AND PTIME BETWEEN SYSDATE-25 AND SYSDATE;

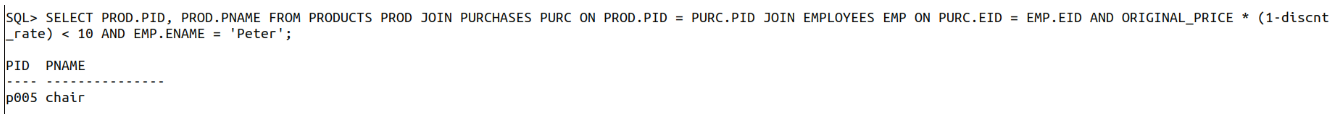


```
SQL> SELECT CUST.CNAME, CUST.TELEPHONE#, PTIME  FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID=PURC.CID WHERE PURC.TOTAL_PRICE >= 100 AND PTIME BETWEEN SYSDATE-2
5 AND SYSDATE;

CNAME           TELEPHONE#   PTIME
--------------- ------------ ---------
Kathy           666-555-4567 20-FEB-20
Chris           666-555-6745 18-FEB-20
```

3. Find the pids and names of those products that are priced below 10 dollars (based on discount price) and are purchased through an employee named Peter. The discount price or sale price of a product is computed by original_price * (1 – discnt_rate).

Query:

SELECT PROD.PID, PROD.PNAME FROM PRODUCTS PROD JOIN PURCHASES PURC ON PROD.PID = PURC.PID JOIN EMPLOYEES EMP ON PURC.EID = EMP.EID AND ORIGINAL_PRICE * (1-discnt_rate) < 10 AND EMP.ENAME = 'Peter';

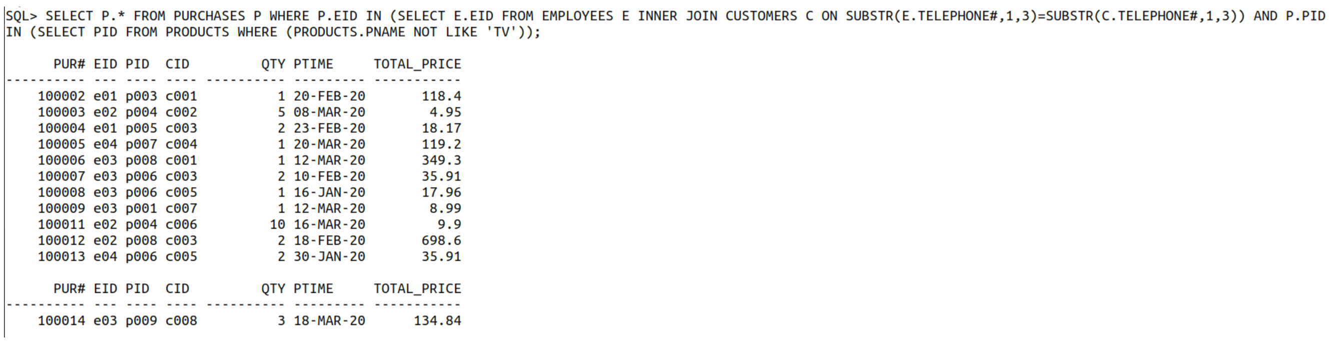


```
SQL> SELECT PROD.PID, PROD.PNAME FROM PRODUCTS PROD JOIN PURCHASES PURC ON PROD.PID = PURC.PID JOIN EMPLOYEES EMP ON PURC.EID = EMP.EID AND ORIGINAL_PRICE * (1-discnt
_rate) < 10 AND EMP.ENAME = 'Peter';

PID  PNAME
---- ---------------
p005 chair
```

4. Find each purchase that involves an employee whose telephone number has the same area code as that of the customer who purchased a non-TV product. All attributes of qualified purchases should be returned.

Query:

SELECT P.* FROM PURCHASES P WHERE P.EID IN (SELECT E.EID FROM EMPLOYEES E INNER JOIN CUSTOMERS C ON SUBSTR(E.TELEPHONE#,1,3)=SUBSTR(C.TELEPHONE#,1,3)) AND P.PID IN (SELECT PID FROM PRODUCTS WHERE (PRODUCTS.PNAME NOT LIKE 'TV'));

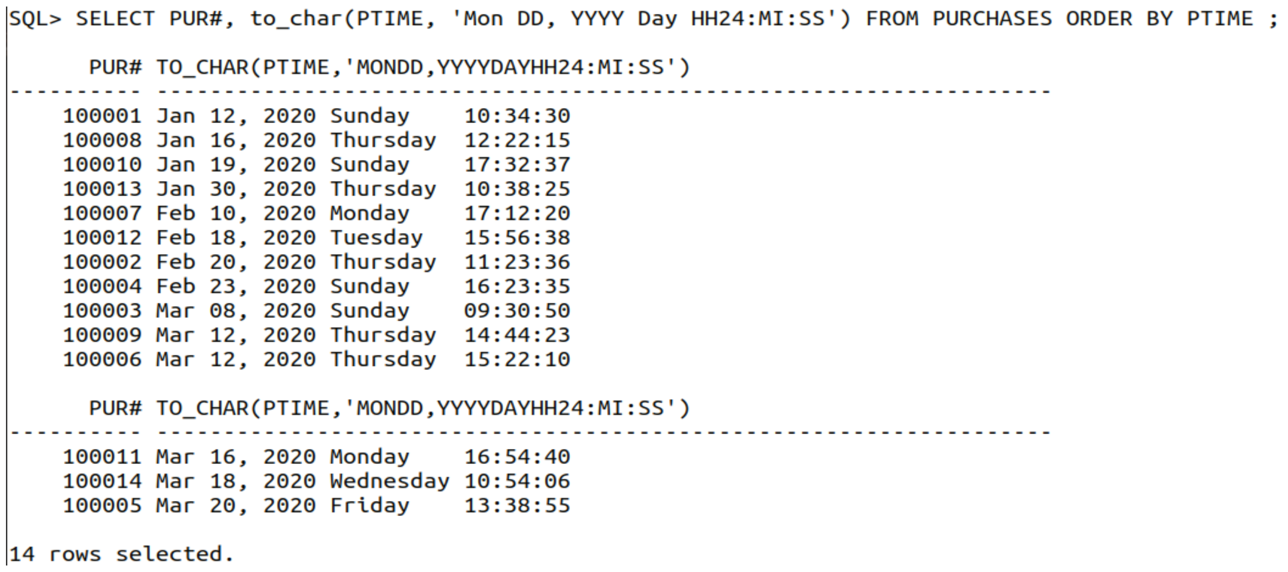


```
SQL> SELECT P.* FROM PURCHASES P WHERE P.EID IN (SELECT E.EID FROM EMPLOYEES E INNER JOIN CUSTOMERS C ON SUBSTR(E.TELEPHONE#,1,3)=SUBSTR(C.TELEPHONE#,1,3)) AND P.PID
IN (SELECT PID FROM PRODUCTS WHERE (PRODUCTS.PNAME NOT LIKE 'TV'));

      PUR# EID PID  CID         QTY PTIME     TOTAL_PRICE
---------- --- ---- ---- ---------- --------- -----------
    100002 e01 p003 c001          1 20-FEB-20       118.4
    100003 e02 p004 c002          5 08-MAR-20        4.95
    100004 e01 p005 c003          2 23-FEB-20       18.17
    100005 e04 p007 c004          1 20-MAR-20       119.2
    100006 e03 p008 c001          1 12-MAR-20       349.3
    100007 e03 p006 c003          2 10-FEB-20       35.91
    100008 e03 p006 c005          1 16-JAN-20       17.96
    100009 e03 p001 c007          1 12-MAR-20        8.99
    100011 e02 p004 c006         10 16-MAR-20         9.9
    100012 e02 p008 c003          2 18-FEB-20       698.6
    100013 e04 p006 c005          2 30-JAN-20       35.91

      PUR# EID PID  CID         QTY PTIME     TOTAL_PRICE
---------- --- ---- ---- ---------- --------- -----------
    100014 e03 p009 c008          3 18-MAR-20      134.84
```

5. Find the purchase number (pur#) and ptime of each purchase. It is required that ptime be displayed in a format as illustrated by the following example: March 23, 2020 Friday 08:33:46. Furthermore, the results must be displayed in increasing ptime order.

Query:

SELECT PUR#, to_char(PTIME, 'Mon DD, YYYY Day HH24:MI:SS') FROM PURCHASES ORDER BY PTIME ;

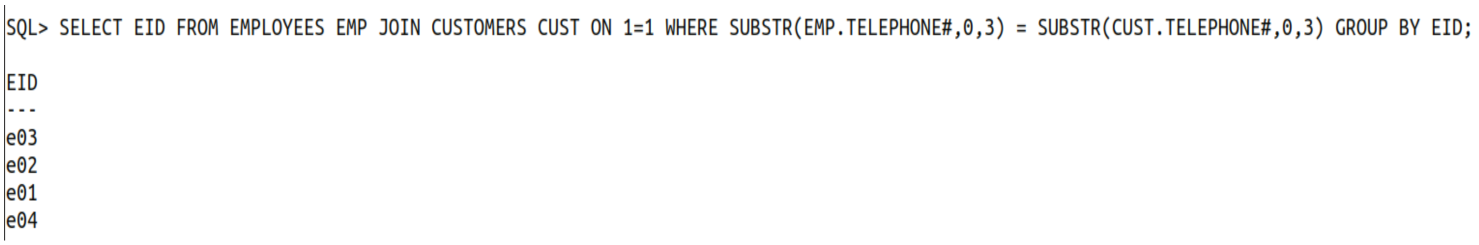


```
SQL> SELECT PUR#, to_char(PTIME, 'Mon DD, YYYY Day HH24:MI:SS') FROM PURCHASES ORDER BY PTIME ;

      PUR# TO_CHAR(PTIME,'MONDD,YYYYDAYHH24:MI:SS')
---------- ----------------------------------------------------------------
    100001 Jan 12, 2020 Sunday    10:34:30
    100008 Jan 16, 2020 Thursday  12:22:15
    100010 Jan 19, 2020 Sunday    17:32:37
    100013 Jan 30, 2020 Thursday  10:38:25
    100007 Feb 10, 2020 Monday    17:12:20
    100012 Feb 18, 2020 Tuesday   15:56:38
    100002 Feb 20, 2020 Thursday  11:23:36
    100004 Feb 23, 2020 Sunday    16:23:35
    100003 Mar 08, 2020 Sunday    09:30:50
    100009 Mar 12, 2020 Thursday  14:44:23
    100006 Mar 12, 2020 Thursday  15:22:10

      PUR# TO_CHAR(PTIME,'MONDD,YYYYDAYHH24:MI:SS')
---------- ----------------------------------------------------------------
    100011 Mar 16, 2020 Monday    16:54:40
    100014 Mar 18, 2020 Wednesday 10:54:06
    100005 Mar 20, 2020 Friday    13:38:55

14 rows selected.
```

6. Find the eids of those employees whose telephone number has the same area code as that of at least one customer. Note that the customer is not necessarily the employee's customer; that is, the customer may not necessarily have made a purchase from the employee. For this query, make sure that no duplicate results are returned and you are not allowed to use “select distinct”.

Query:

SELECT EID FROM EMPLOYEES EMP JOIN CUSTOMERS CUST ON 1=1 WHERE SUBSTR(EMP.TELEPHONE#,0,3) = SUBSTR(CUST.TELEPHONE#,0,3) GROUP BY EID;

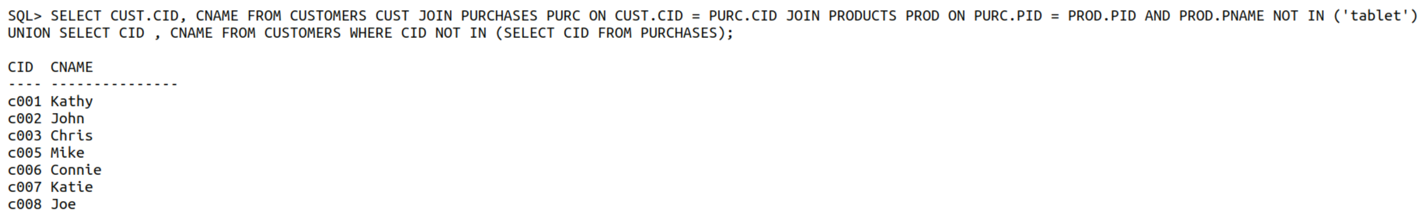


```
SQL> SELECT EID FROM EMPLOYEES EMP JOIN CUSTOMERS CUST ON 1=1 WHERE SUBSTR(EMP.TELEPHONE#,0,3) = SUBSTR(CUST.TELEPHONE#,0,3) GROUP BY EID;

EID
---
e03
e02
e01
e04
```

7. Find the names of those customers who did not purchase a tablet in their last visit to the retail business.

Query:

SELECT CUST.CID, CNAME FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID = PURC.CID JOIN PRODUCTS PROD ON PURC.PID = PROD.PID AND PROD.PNAME NOT IN ('tablet') UNION SELECT CID , CNAME FROM CUSTOMERS WHERE CID NOT IN (SELECT CID FROM PURCHASES);

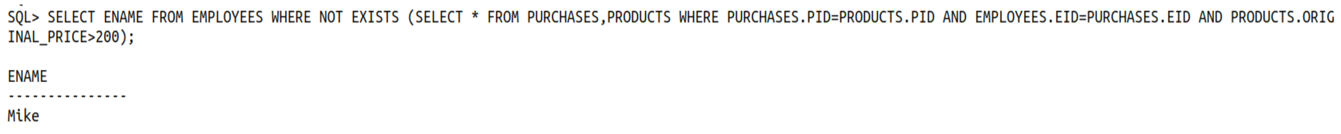


```
SQL> SELECT CUST.CID, CNAME FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID = PURC.CID JOIN PRODUCTS PROD ON PURC.PID = PROD.PID AND PROD.PNAME NOT IN ('tablet')
UNION SELECT CID , CNAME FROM CUSTOMERS WHERE CID NOT IN (SELECT CID FROM PURCHASES);

CID  CNAME
---- ---------------
c001 Kathy
c002 John
c003 Chris
c005 Mike
c006 Connie
c007 Katie
c008 Joe
```

8. Find the names of those employees who have not sold any product whose original price is $200 or higher. Use “not exists” to answer this query.

Query:

SELECT ENAME FROM EMPLOYEES WHERE NOT EXISTS (SELECT * FROM PURCHASES,PRODUCTS WHERE PURCHASES.PID=PRODUCTS.PID AND EMPLOYEES.EID=PURCHASES.EID AND PRODUCTS.ORIGINAL_PRICE>200);

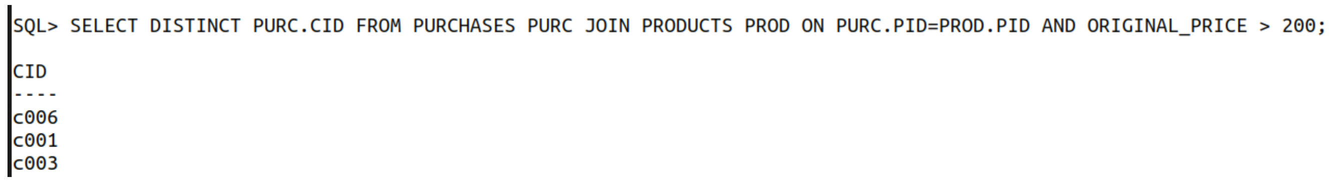


```
SQL> SELECT ENAME FROM EMPLOYEES WHERE NOT EXISTS (SELECT * FROM PURCHASES,PRODUCTS WHERE PURCHASES.PID=PRODUCTS.PID AND EMPLOYEES.EID=PURCHASES.EID AND PRODUCTS.ORIG
INAL_PRICE>200);

ENAME
---------------
Mike
```

9. Find the cids of those customers who have purchased all the products whose original prices are above $200.

Query:

SELECT DISTINCT PURC.CID FROM PURCHASES PURC JOIN PRODUCTS PROD ON PURC.PID=PROD.PID AND ORIGINAL_PRICE > 200;

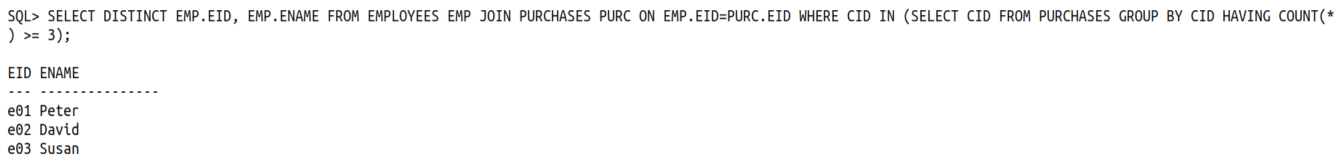


```
SQL> SELECT DISTINCT PURC.CID FROM PURCHASES PURC JOIN PRODUCTS PROD ON PURC.PID=PROD.PID AND ORIGINAL_PRICE > 200;

CID
----
c006
c001
c003
```

10. Find the eids and names of those employees who have made sale to all the customers who have visited the retail business at least 3 times.

Query:

SELECT DISTINCT EMP.EID, EMP.ENAME FROM EMPLOYEES EMP JOIN PURCHASES PURC ON EMP.EID=PURC.EID WHERE CID IN (SELECT CID FROM PURCHASES GROUP BY CID HAVING COUNT(*) >= 3);

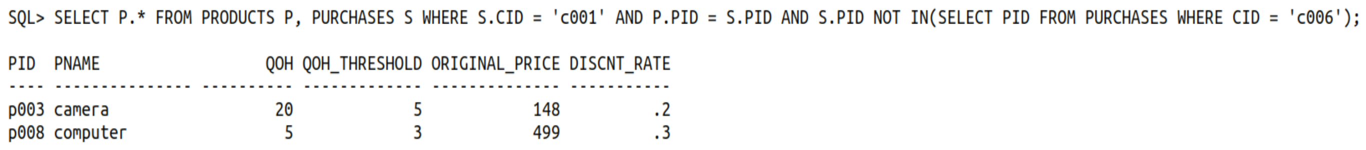


```
SQL> SELECT DISTINCT EMP.EID, EMP.ENAME FROM EMPLOYEES EMP JOIN PURCHASES PURC ON EMP.EID=PURC.EID WHERE CID IN (SELECT CID FROM PURCHASES GROUP BY CID HAVING COUNT(*
) >= 3);

EID ENAME
--- ---------------
e01 Peter
e02 David
e03 Susan
```

11. Find those products that are purchased by customer c001 but not by customer c006. All attributes of qualified products should be returned.

Query:

SELECT P.* FROM PRODUCTS P, PURCHASES S WHERE S.CID = 'c001' AND P.PID = S.PID AND S.PID NOT IN(SELECT PID FROM PURCHASES WHERE CID = 'c006');

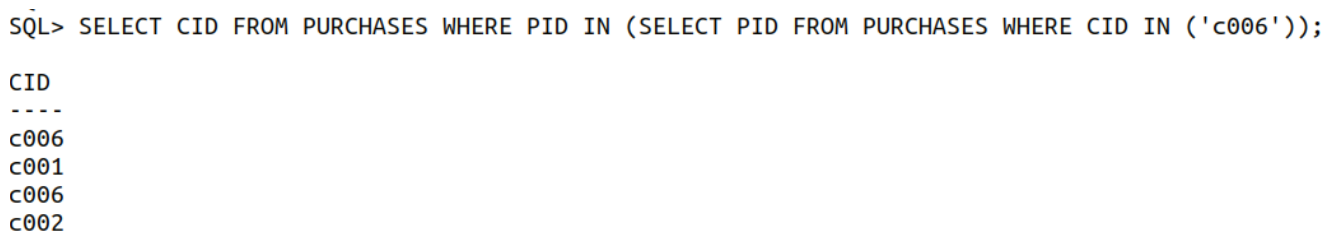


```
SQL> SELECT P.* FROM PRODUCTS P, PURCHASES S WHERE S.CID = 'c001' AND P.PID = S.PID AND S.PID NOT IN(SELECT PID FROM PURCHASES WHERE CID = 'c006');

PID  PNAME                  QOH QOH_THRESHOLD ORIGINAL_PRICE DISCNT_RATE
---- --------------- ---------- ------------- -------------- -----------
p003 camera                  20             5            148          .2
p008 computer                 5             3            499          .3
```

12. Find the cids of those customers who purchased at least one product that is also purchased by customer c006. Customer c006 should be included in the result.

Query:

SELECT CID FROM PURCHASES WHERE PID IN (SELECT PID FROM PURCHASES WHERE CID IN ('c006'));

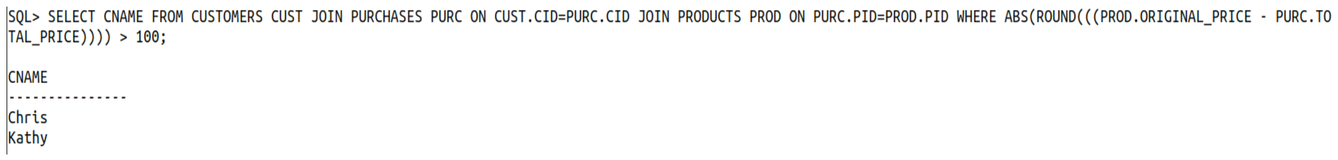


```
SQL> SELECT CID FROM PURCHASES WHERE PID IN (SELECT PID FROM PURCHASES WHERE CID IN ('c006'));

CID
----
c006
c001
c006
c002
```

13. Find the names of those customers who saved more than $100 in a single purchase (i.e., based on one record in the purchases table) from the original price of the product.

Query:

SELECT CNAME FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID=PURC.CID JOIN PRODUCTS PROD ON PURC.PID=PROD.PID WHERE ABS(ROUND(((PROD.ORIGINAL_PRICE - PURC.TOTAL_PRICE)))) > 100;

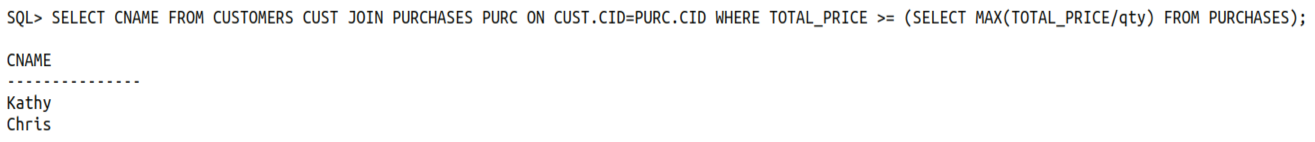


```
SQL> SELECT CNAME FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID=PURC.CID JOIN PRODUCTS PROD ON PURC.PID=PROD.PID WHERE ABS(ROUND(((PROD.ORIGINAL_PRICE - PURC.TO
TAL_PRICE)))) > 100;

CNAME
---------------
Chris
Kathy
```

14. Find the names of those customers who have made at least one purchase that has the highest total price among all purchases. Note that it is possible for multiple purchases to have the same highest total price.

Query:

SELECT CNAME FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID=PURC.CID WHERE TOTAL_PRICE >= (SELECT MAX(TOTAL_PRICE/QTY) FROM PURCHASES);

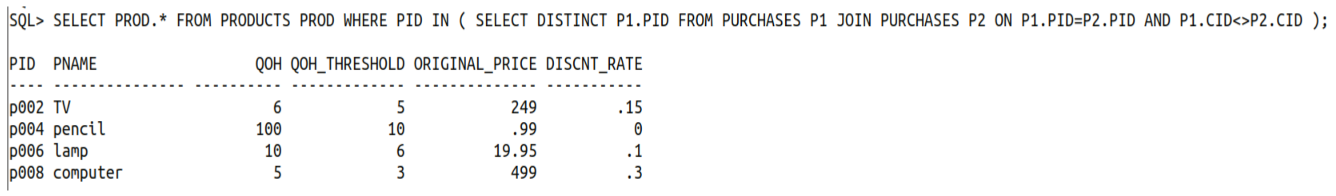


```
SQL> SELECT CNAME FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID=PURC.CID WHERE TOTAL_PRICE >= (SELECT MAX(TOTAL_PRICE/qty) FROM PURCHASES);

CNAME
--------------
Kathy
Chris
```

15. Find those products that are purchased by at least two different customers. All attributes of qualified products should be returned.

Query:

SELECT PROD.* FROM PRODUCTS PROD WHERE PID IN (SELECT DISTINCT P1.PID FROM PURCHASES P1 JOIN PURCHASES P2 ON P1.PID=P2.PID AND P1.CID<>P2.CID);

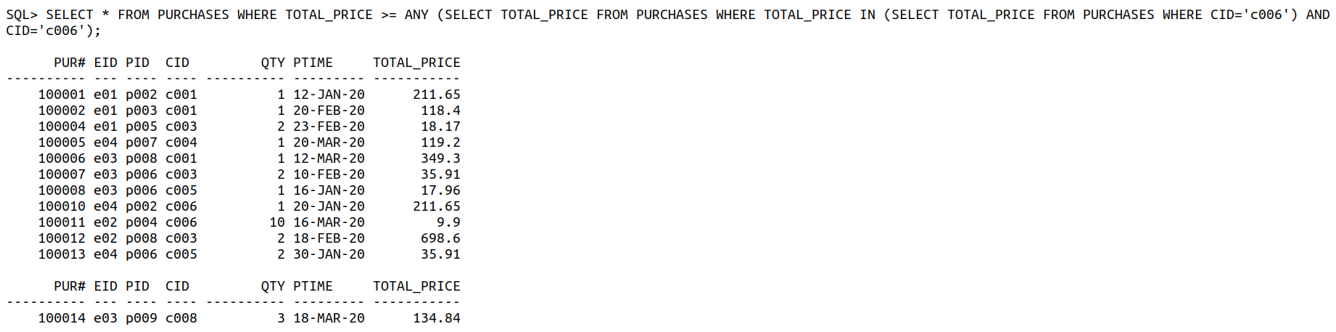


```
SQL> SELECT PROD.* FROM PRODUCTS PROD WHERE PID IN ( SELECT DISTINCT P1.PID FROM PURCHASES P1 JOIN PURCHASES P2 ON P1.PID=P2.PID AND P1.CID<>P2.CID );

PID  PNAME                  QOH QOH_THRESHOLD ORIGINAL_PRICE DISCNT_RATE
---- --------------- ---------- ------------- -------------- -----------
p002 TV                       6             5            249         .15
p004 pencil                 100            10            .99           0
p006 lamp                    10             6          19.95          .1
p008 computer                 5             3            499          .3
```

16. Find the purchase number (pur#) of each purchase whose total price is greater than or equal to each of the total prices of those purchases placed by customer c006. Do not use any aggregate function.

Query:

SELECT * FROM PURCHASES WHERE TOTAL_PRICE >= ANY (SELECT TOTAL_PRICE FROM PURCHASES WHERE TOTAL_PRICE IN (SELECT TOTAL_PRICE FROM PURCHASES WHERE CID='c006') AND CID='c006');

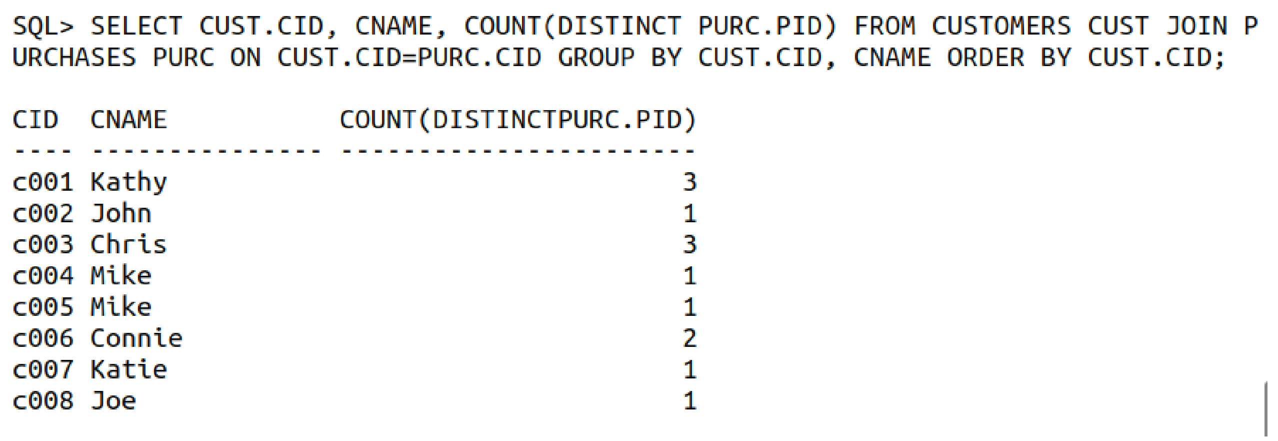


```
SQL> SELECT * FROM PURCHASES WHERE TOTAL_PRICE >= ANY (SELECT TOTAL_PRICE FROM PURCHASES WHERE TOTAL_PRICE IN (SELECT TOTAL_PRICE FROM PURCHASES WHERE CID='c006') AND
CID='c006');

      PUR# EID PID  CID         QTY PTIME     TOTAL_PRICE
---------- --- ---- ---- ---------- --------- -----------
    100001 e01 p002 c001          1 12-JAN-20      211.65
    100002 e01 p003 c001          1 20-FEB-20       118.4
    100004 e01 p005 c003          2 23-FEB-20       18.17
    100005 e04 p007 c004          1 20-MAR-20       119.2
    100006 e03 p008 c001          1 12-MAR-20       349.3
    100007 e03 p006 c003          2 10-FEB-20       35.91
    100008 e03 p006 c005          1 16-JAN-20       17.96
    100010 e04 p002 c006          1 20-JAN-20      211.65
    100011 e02 p004 c006         10 16-MAR-20         9.9
    100012 e02 p008 c003          2 18-FEB-20       698.6
    100013 e04 p006 c005          2 30-JAN-20       35.91

      PUR# EID PID  CID         QTY PTIME     TOTAL_PRICE
---------- --- ---- ---- ---------- --------- -----------
    100014 e03 p009 c008          3 18-MAR-20      134.84
```

17. Find the cid and name of each customer as well as the number of different types of products purchased by the customer.

Query:

SELECT CUST.CID, CNAME, COUNT(DISTINCT PURC.PID) FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID=PURC.CID GROUP BY CUST.CID, CNAME ORDER BY CUST.CID;

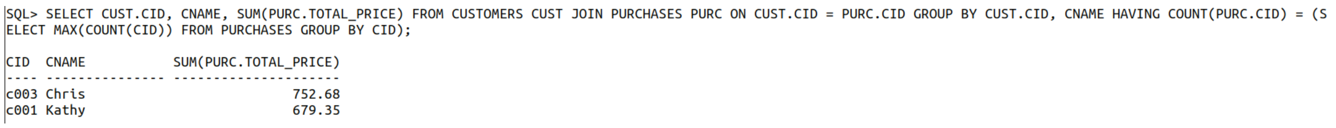


```
SQL> SELECT CUST.CID, CNAME, COUNT(DISTINCT PURC.PID) FROM CUSTOMERS CUST JOIN P
URCHASES PURC ON CUST.CID=PURC.CID GROUP BY CUST.CID, CNAME ORDER BY CUST.CID;

CID  CNAME           COUNT(DISTINCTPURC.PID)
---- --------------- -----------------------
c001 Kathy                                 3
c002 John                                  1
c003 Chris                                 3
c004 Mike                                  1
c005 Mike                                  1
c006 Connie                                2
c007 Katie                                 1
c008 Joe                                   1
```

18. Find the cid and name of each customer who has visited the retail business the most number of times and also display the total amount of money such a customer has spent at the retail business. Note that it is possible that multiple customers have visited the retail business for the (same) largest number of times.

Query:

SELECT CUST.CID, CNAME, SUM(PURC.TOTAL_PRICE) FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID = PURC.CID GROUP BY CUST.CID, CNAME HAVING COUNT(PURC.CID) = (SELECT MAX(COUNT(CID)) FROM PURCHASES GROUP BY CID);

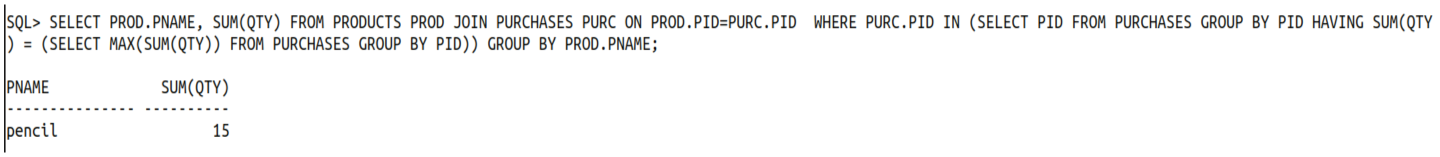


```
SQL> SELECT CUST.CID, CNAME, SUM(PURC.TOTAL_PRICE) FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID = PURC.CID GROUP BY CUST.CID, CNAME HAVING COUNT(PURC.CID) = (S
ELECT MAX(COUNT(CID)) FROM PURCHASES GROUP BY CID);

CID  CNAME           SUM(PURC.TOTAL_PRICE)
---- --------------- --------------------
c003 Chris                         752.68
c001 Kathy                         679.35
```

19. Find the name and the total quantity sold of each product that has sold the most units (i.e., with the largest total quantity sold). You may assume that all products have different names. It is possible that multiple products have sold the same highest total units.

Query:

SELECT PROD.PNAME, SUM(QTY) FROM PRODUCTS PROD JOIN PURCHASES PURC ON PROD.PID=PURC.PID WHERE PURC.PID IN (SELECT PID FROM PURCHASES GROUP BY PID HAVING SUM(QTY) = (SELECT MAX(SUM(QTY)) FROM PURCHASES GROUP BY PID)) GROUP BY PROD.PNAME;

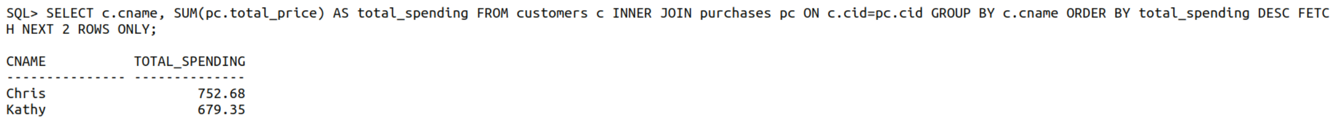


```
SQL> SELECT PROD.PNAME, SUM(QTY) FROM PRODUCTS PROD JOIN PURCHASES PURC ON PROD.PID=PURC.PID  WHERE PURC.PID IN (SELECT PID FROM PURCHASES GROUP BY PID HAVING SUM(QTY
) = (SELECT MAX(SUM(QTY)) FROM PURCHASES GROUP BY PID)) GROUP BY PROD.PNAME;

PNAME            SUM(QTY)
--------------- ----------
pencil                 15
```

20. Find the names of the top two customers in terms of their spending at the retail business. For each such customer, also display the total amount of money he or she has spent. (The top two customers may have spent the same amount or different amounts of money. If more than two customers have the same highest expenditure at the retail store, return any two of those customers.)

Query:

SELECT C.CNAME, SUM(PC.TOTAL_PRICE) AS TOTAL_SPENDING FROM CUSTOMERS C INNER JOIN PURCHASES PC ON C.CID=PC.CID GROUP BY C.CNAME ORDER BY TOTAL_SPENDING DESC FETCH NEXT 2 ROWS ONLY;



```
SQL> SELECT c.cname, SUM(pc.total_price) AS total_spending FROM customers c INNER JOIN purchases pc ON c.cid=pc.cid GROUP BY c.cname ORDER BY total_spending DESC FETC
H NEXT 2 ROWS ONLY;

CNAME           TOTAL_SPENDING
--------------- --------------
Chris                   752.68
Kathy                   679.35
```