CS 532 Database System Project 1

**Team Members:**

Name: Akshay Kumar Anvekar

B No: B00813848

[aanveka2@binghamton.edu](mailto:aanveka2@binghamton.edu)

Name: Sanjay Bettadapura Ganesha

B No: B00816568

[sbettad1@binghamton.edu](mailto:sbettad1@binghamton.edu)

Name: Akshay Kumar Anvekar

B No: B00813848

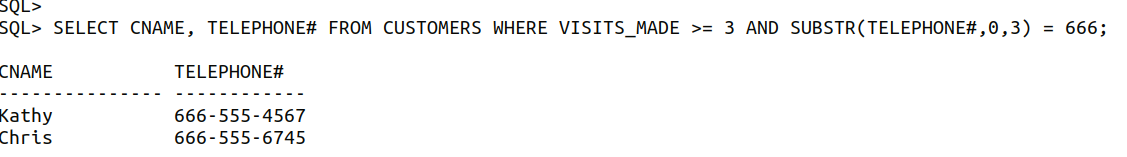
Name: Sanjay Bettadapura Ganesha

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:

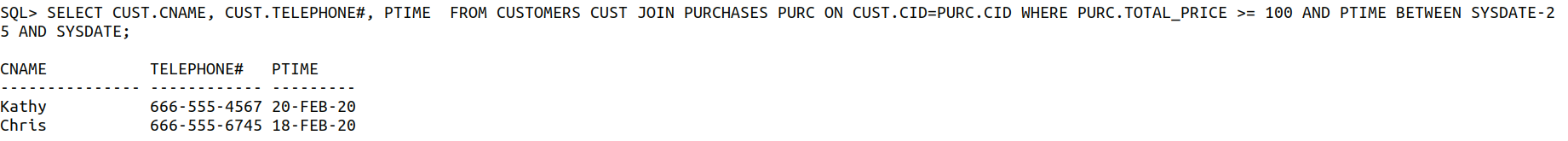
SELECT CNAME, TELEPHONE# FROM CUSTOMERS WHERE VISITS\_MADE >= 3 AND SUBSTR(TELEPHONE#,0,3) = 666;



1. 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:

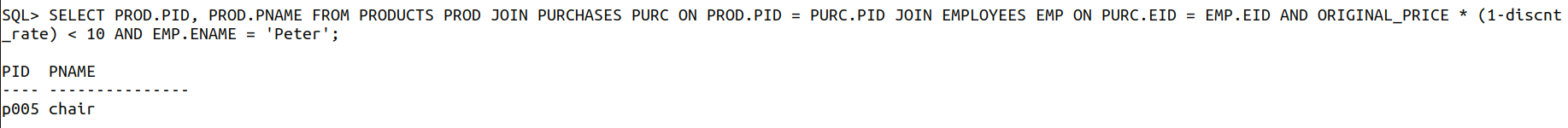
SELECT CUST.CNAME, CUST.TELEPHONE#, PTIME FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID=PURC.CID WHERE PURC.TOTAL\_PRICE >= 100 AND PTIME BETWEEN SYSDATE-25 AND SYSDATE;



1. 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:

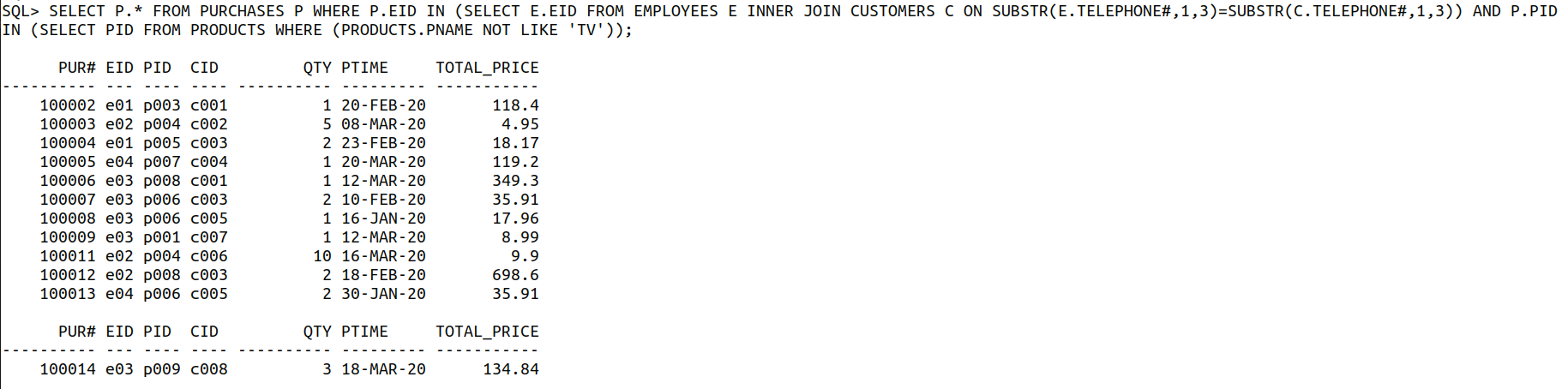
SELECT PROD.PID, PROD.PNAME FROM PRODUCTS PROD JOIN PURCHASES PURC ON PROD.PID = PURC.PID JOIN EMPLOYEES EMP ON PURC.EID = EMP.EID AND ORIGINAL\_PRICE \* (1-discnt\_rate) < 10 AND EMP.ENAME = 'Peter';



1. 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:

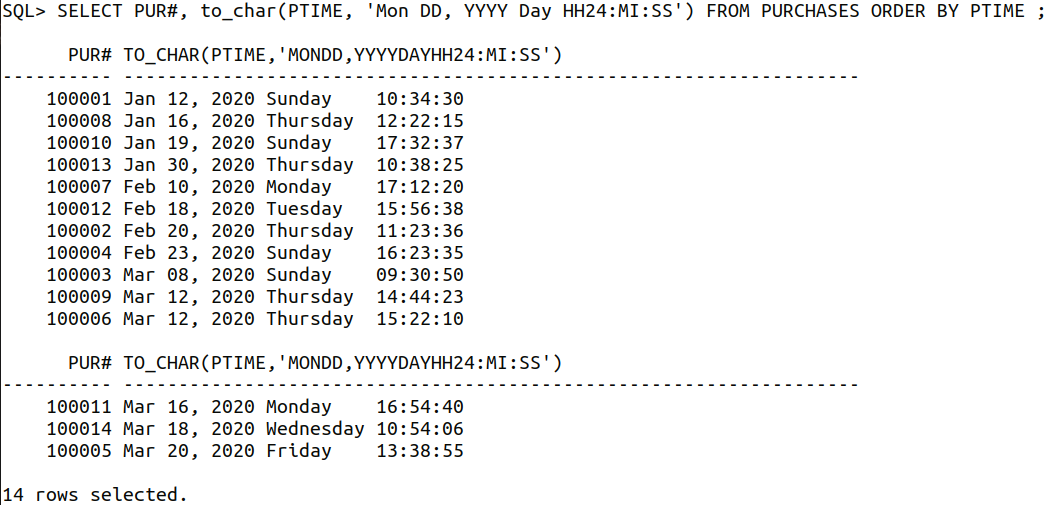
SELECT P.\* FROM PURCHASES P WHERE P.EID IN (SELECT E.EID FROM EMPLOYEES E INNER JOIN CUSTOMERS C ON SUBSTR(E.TELEPHONE#,1,3)=SUBSTR(C.TELEPHONE#,1,3)) AND P.PID IN (SELECT PID FROM PRODUCTS WHERE (PRODUCTS.PNAME NOT LIKE 'TV'));



1. 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:

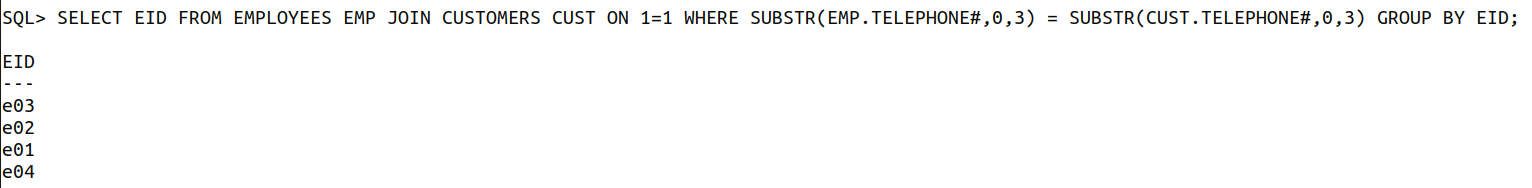
SELECT PUR#, to\_char(PTIME, 'Mon DD, YYYY Day HH24:MI:SS') FROM PURCHASES ORDER BY PTIME ;



1. 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:

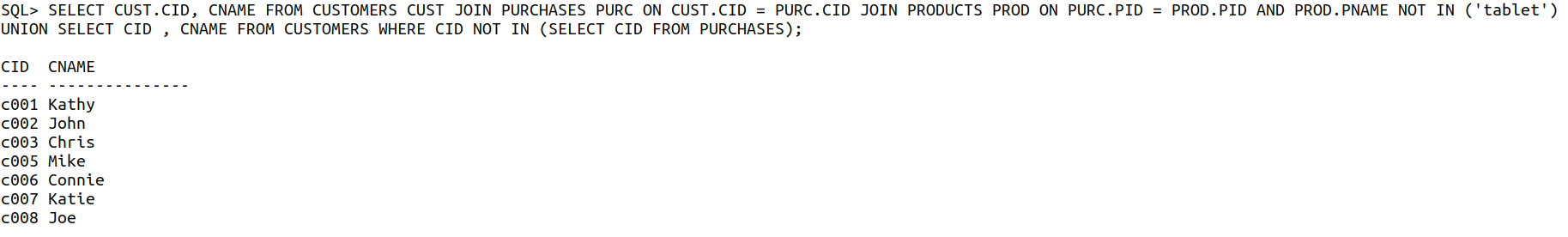
SELECT EID FROM EMPLOYEES EMP JOIN CUSTOMERS CUST ON 1=1 WHERE SUBSTR(EMP.TELEPHONE#,0,3) = SUBSTR(CUST.TELEPHONE#,0,3) GROUP BY EID;



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

Query:

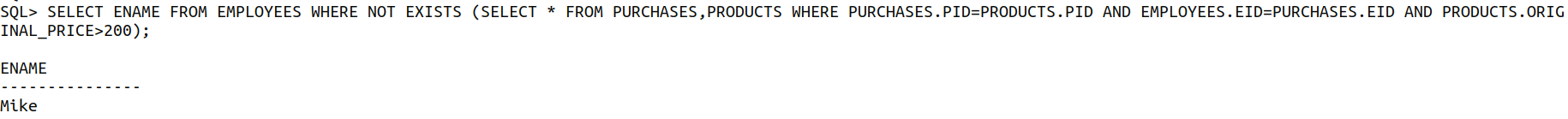
SELECT CUST.CID, CNAME FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID = PURC.CID JOIN PRODUCTS PROD ON PURC.PID = PROD.PID AND PROD.PNAME NOT IN ('tablet') UNION SELECT CID , CNAME FROM CUSTOMERS WHERE CID NOT IN (SELECT CID FROM PURCHASES);



1. 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:

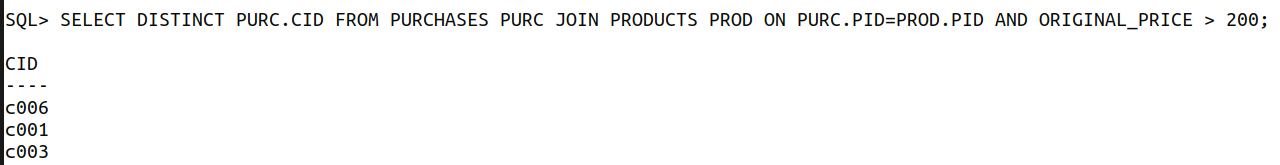
SELECT ENAME FROM EMPLOYEES WHERE NOT EXISTS (SELECT \* FROM PURCHASES,PRODUCTS WHERE PURCHASES.PID=PRODUCTS.PID AND EMPLOYEES.EID=PURCHASES.EID AND PRODUCTS.ORIGINAL\_PRICE>200);



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

Query:

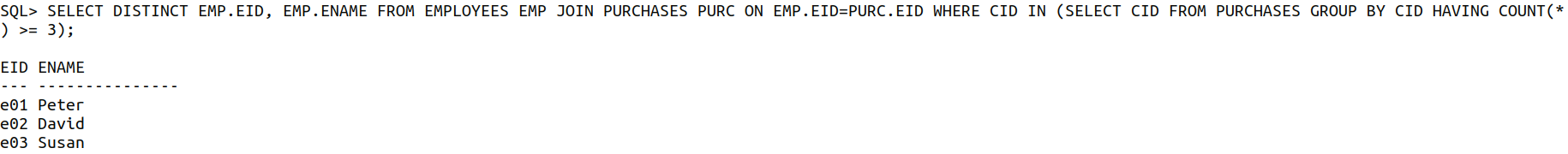
SELECT DISTINCT PURC.CID FROM PURCHASES PURC JOIN PRODUCTS PROD ON PURC.PID=PROD.PID AND ORIGINAL\_PRICE > 200;



1. 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:

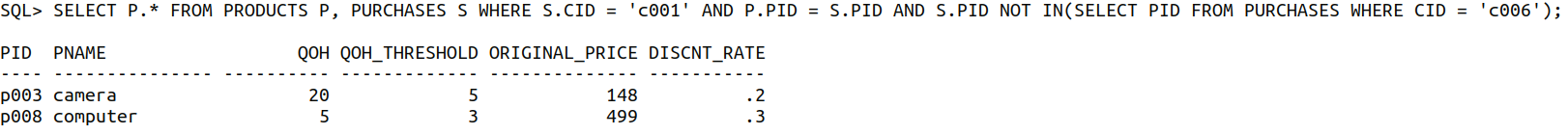
SELECT DISTINCT EMP.EID, EMP.ENAME FROM EMPLOYEES EMP JOIN PURCHASES PURC ON EMP.EID=PURC.EID WHERE CID IN (SELECT CID FROM PURCHASES GROUP BY CID HAVING COUNT(\*) >= 3);



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

Query:

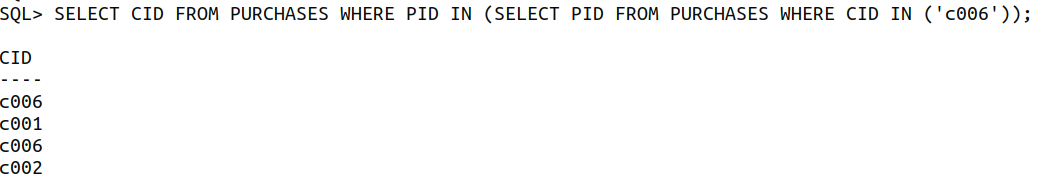
SELECT P.\* FROM PRODUCTS P, PURCHASES S WHERE S.CID = 'c001' AND P.PID = S.PID AND S.PID NOT IN(SELECT PID FROM PURCHASES WHERE CID = 'c006');



1. 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:

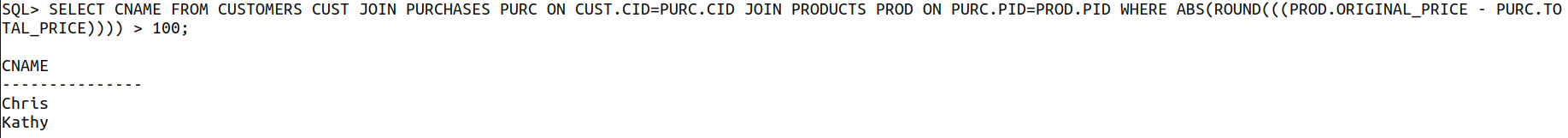
SELECT CID FROM PURCHASES WHERE PID IN (SELECT PID FROM PURCHASES WHERE CID IN ('c006'));



1. 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:

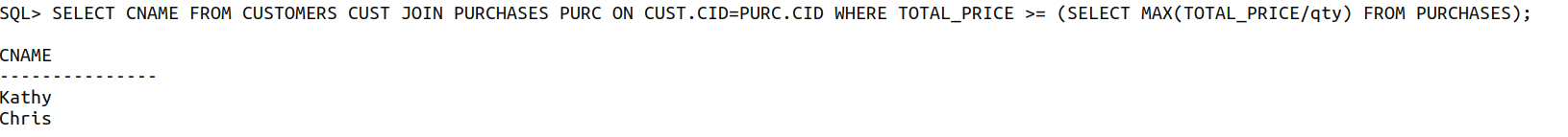
SELECT CNAME FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID=PURC.CID JOIN PRODUCTS PROD ON PURC.PID=PROD.PID WHERE ABS(ROUND(((PROD.ORIGINAL\_PRICE - PURC.TOTAL\_PRICE)))) > 100;



1. 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:

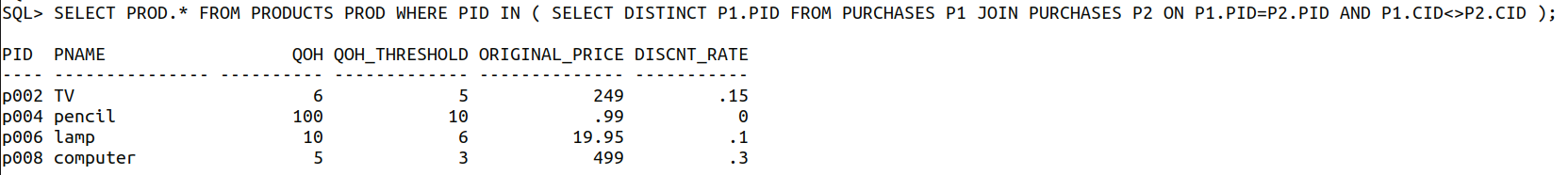
SELECT CNAME FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID=PURC.CID WHERE TOTAL\_PRICE >= (SELECT MAX(TOTAL\_PRICE/QTY) FROM PURCHASES);



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

Query:

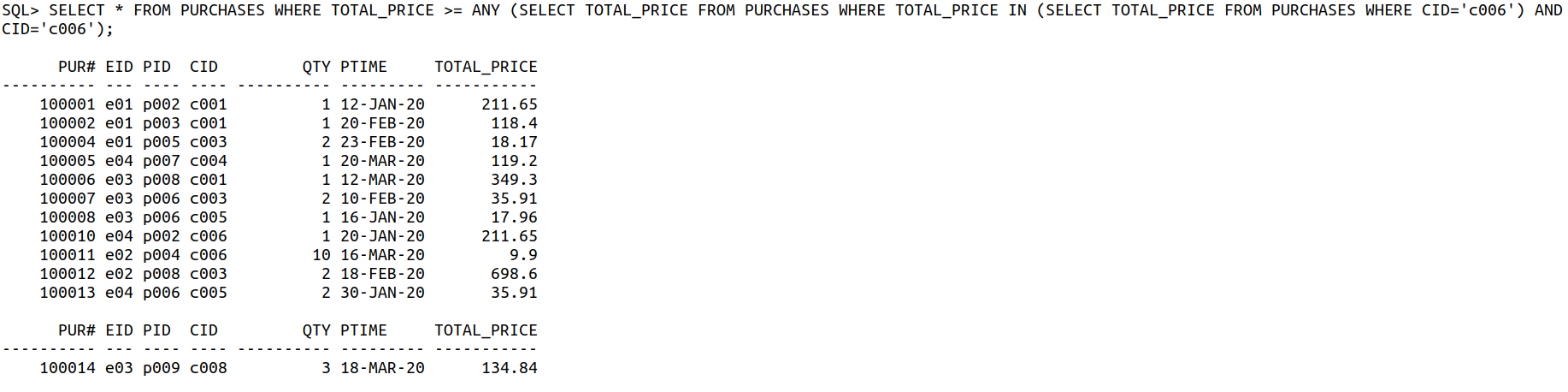
SELECT PROD.\* FROM PRODUCTS PROD WHERE PID IN ( SELECT DISTINCT P1.PID FROM PURCHASES P1 JOIN PURCHASES P2 ON P1.PID=P2.PID AND P1.CID<>P2.CID );



1. 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:

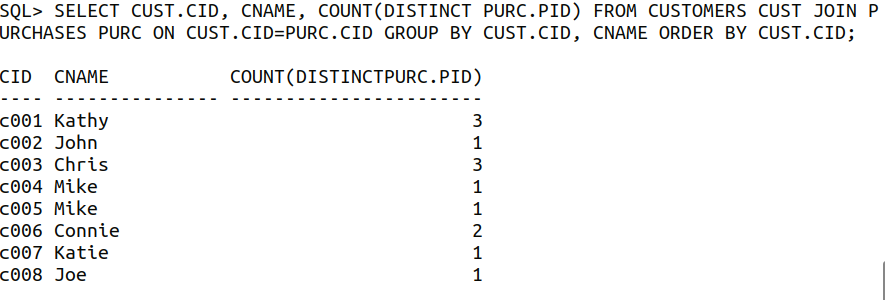
SELECT \* FROM PURCHASES WHERE TOTAL\_PRICE >= ANY (SELECT TOTAL\_PRICE FROM PURCHASES WHERE TOTAL\_PRICE IN (SELECT TOTAL\_PRICE FROM PURCHASES WHERE CID='c006') AND CID='c006');



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

Query:

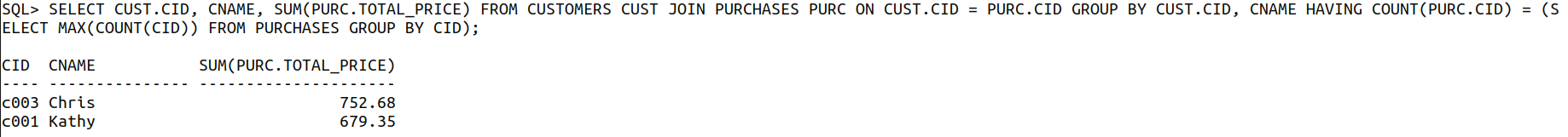
SELECT CUST.CID, CNAME, COUNT(DISTINCT PURC.PID) FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID=PURC.CID GROUP BY CUST.CID, CNAME ORDER BY CUST.CID;



1. 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:

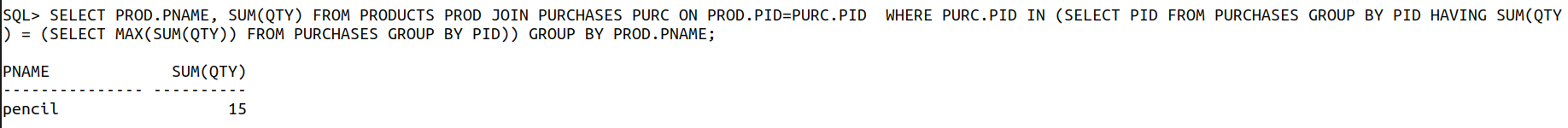
SELECT CUST.CID, CNAME, SUM(PURC.TOTAL\_PRICE) FROM CUSTOMERS CUST JOIN PURCHASES PURC ON CUST.CID = PURC.CID GROUP BY CUST.CID, CNAME HAVING COUNT(PURC.CID) = (SELECT MAX(COUNT(CID)) FROM PURCHASES GROUP BY CID);



1. 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;



1. 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;

