Project 1: Writing SQL Queries Using Oracle's SQL*Plus

Upload your source code by 8:59 am on March 2
Submit a hard copy in class on March 2, 2020

The following four tables are simplified from the tables for the Retail Business Management System project:

employees(eid, ename, telephone#)
customers(cid, cname, telephone#, visits_made, last_visit_date)
products(pid, pname, qoh, qoh_threshold, original_price, discnt_rate)
purchases(pur#, eid, pid, cid, ptime, qty, total_price)

We assume telephone numbers are of format: 607-777-1234, where the first 3 digits (e.g., 607) indicate the area code. visits_made is the number of visits to the retail business a customer has made. last_visit_date is the date of the most recent visit made by a customer. It is assumed that all visits made on the same day are counted as one visit. qoh is the quantity on hand and qoh_threshold is the threshold for quantity on hand for adding supplies. original_price is the original price of a product. discnt_rate is the discount percentage over the original price of each product. ptime is the time when a purchase is made. qty is the quantity of product purchased, and total_price is the total dollar amount of each purchase. Each tuple in table purchases represents a purchase made by a customer for a product through an employee. eid, cid, pid and pur# are the primary keys of employees, customers, products and purchases, respectively.

Please copy and paste the SQL statements below into a file (say project1.sql) in your bingsuns account and save it. After you log into your Oracle account, use SQL> **start project1** to create these tables in your account.

```
create table employees
(eid char(3) primary key,
ename varchar2(15),
telephone# char(12));

create table customers
(cid char(4) primary key,
cname varchar2(15),
telephone# char(12),
visits_made number(4),
last_visit_date date);

create table products
(pid char(4) primary key,
pname varchar2(15),
qoh number(5),
qoh_threshold number(4),
original_price number(6,2),
discnt_rate number(3,2));

create table purchases
(pur# number(6) primary key,
eid char(3) references employees(eid),
pid char(4) references products(pid),
cid char(4) references customers(cid),
qty number(5),
ptime date,
total_price number(7,2));
```

```
insert into employees values ('e01', 'Peter', '666-555-1234');
insert into employees values ('e02', 'David', '777-555-2341');
insert into employees values ('e03', 'Susan', '888-555-3412');
insert into employees values ('e04', 'Anne', '666-555-4123');
insert into employees values ('e05', 'Mike', '444-555-4231');

insert into customers values ('c001', 'Kathy', '666-555-4567', 3, '12-MAR-20');
insert into customers values ('c002', 'John', '888-555-7456', 1, '08-MAR-20');
insert into customers values ('c003', 'Chris', '666-555-6745', 3, '18-FEB-20');
insert into customers values ('c004', 'Mike', '999-555-5674', 1, '20-MAR-20');
insert into customers values ('c005', 'Mike', '777-555-4657', 2, '30-JAN-20');
insert into customers values ('c006', 'Connie', '777-555-7654', 2, '16-MAR-20');
insert into customers values ('c007', 'Katie', '888-555-6574', 1, '12-MAR-20');
insert into customers values ('c008', 'Joe', '666-555-5746', 1, '18-MAR-20');

insert into products values ('p001', 'stapler', 60, 20, 9.99, 0.1);
insert into products values ('p002', 'TV', 6, 5, 249, 0.15);
insert into products values ('p003', 'camera', 20, 5, 148, 0.2);
insert into products values ('p004', 'pencil', 100, 10, 0.99, 0.0);
insert into products values ('p005', 'chair', 10, 8, 12.98, 0.3);
insert into products values ('p006', 'lamp', 10, 6, 19.95, 0.1);
insert into products values ('p007', 'tablet', 50, 10, 149, 0.2);
insert into products values ('p008', 'computer', 5, 3, 499, 0.3);
insert into products values ('p009', 'powerbank', 20, 5, 49.95, 0.1);

insert into purchases values (100001, 'e01', 'p002', 'c001', 1, to_date('12-JAN-
2020 10:34:30', 'DD-MON-YYYY HH24:MI:SS'), 211.65);
insert into purchases values (100002, 'e01', 'p003', 'c001', 1, to_date('20-FEB-
2020 11:23:36', 'DD-MON-YYYY HH24:MI:SS'), 118.40);
insert into purchases values (100003, 'e02', 'p004', 'c002', 5, to_date('08-MAR-
2020 09:30:50', 'DD-MON-YYYY HH24:MI:SS'), 4.95);
insert into purchases values (100004, 'e01', 'p005', 'c003', 2, to_date('23-FEB-
2020 16:23:35', 'DD-MON-YYYY HH24:MI:SS'), 18.17);
insert into purchases values (100005, 'e04', 'p007', 'c004', 1, to_date('20-MAR-
2020 13:38:55', 'DD-MON-YYYY HH24:MI:SS'), 119.20);
insert into purchases values (100006, 'e03', 'p008', 'c001', 1, to_date('12-MAR-
2020 15:22:10', 'DD-MON-YYYY HH24:MI:SS'), 349.30);
insert into purchases values (100007, 'e03', 'p006', 'c003', 2, to_date('10-FEB-
2020 17:12:20', 'DD-MON-YYYY HH24:MI:SS'), 35.91);
insert into purchases values (100008, 'e03', 'p006', 'c005', 1, to_date('16-JAN-
2020 12:22:15', 'DD-MON-YYYY HH24:MI:SS'), 17.96);
insert into purchases values (100009, 'e03', 'p001', 'c007', 1, to_date('12-MAR-
2020 14:44:23', 'DD-MON-YYYY HH24:MI:SS'), 8.99);
insert into purchases values (100010, 'e04', 'p002', 'c006', 1, to_date('20-JAN-
2020 17:32:37', 'DD-MON-YYYY HH24:MI:SS'), 211.65);
insert into purchases values (100011, 'e02', 'p004', 'c006', 10, to_date('16-
MAR-2020 16:54:40', 'DD-MON-YYYY HH24:MI:SS'), 9.90);
insert into purchases values (100012, 'e02', 'p008', 'c003', 2, to_date('18-FEB-
2020 15:56:38', 'DD-MON-YYYY HH24:MI:SS'), 698.60);
insert into purchases values (100013, 'e04', 'p006', 'c005', 2, to_date('30-JAN-
2020 10:38:25', 'DD-MON-YYYY HH24:MI:SS'), 35.91);
insert into purchases values (100014, 'e03', 'p009', 'c008', 3, to_date('18-MAR-
2020 10:54:06', 'DD-MON-YYYY HH24:MI:SS'), 134.84);
```

There are 20 statements in this project and you are asked to write a single SQL query for each statement. As the tuples currently in the tables may change (e.g., add, deleted, update tuples), **your query to each question must be correct no matter what valid tuples are in the tables**. Each correct SQL query is worth 5 points. It is very important that you understand each query statement correctly. If you have any question about the correct interpretation of a statement, please ask the instructor for clarification through the blackboard discussion forum for this project. Unless it is explicitly required in a statement, it is up to you to decide whether you want to keep or remove duplicate tuples in the result of each query and no points will be taken off either way.

I suggest that you first test each query individually and save each query in a different file (with extension .sql). After all queries have been tested to your satisfaction, you can run all the queries in a sequence and save the entire session in a spool file. Suppose you have saved your queries in files query1.sql, ..., query20.sql. Following the steps below to generate the spool file after you have logged on Oracle.

```
SQL> set echo on
SQL> spool project1.txt
SQL> start query1
.......
SQL> start query20
SQL> spool off
```

Your printout must list each query followed by the result of executing the query.

The following are the 20 statements:

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.

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).

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).

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.

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.

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”.

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

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.

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

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.

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

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.

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.

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.

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

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.

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

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.

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.

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.)