

Project 1: Writing SQL Queries

(Due by 11:59 pm on November 06, 2024, on Gradescope)

The following six tables will be used for this project. These tables are either directly from or modified from the relations obtained after applying normalization to the tables in Homework 2 solution.

- Employees(eid, name, telephone#, email)
- Customers(cid, first_name, last_name, phone#, visits_made, last_visit_date)
- Products(pid, name, qoh, qoh_threshold, orig_price, discent_category)
- Prod_Discent(discent_category, discent_rate)
- Purchases(pur#, eid, pid, cid, pur_time, quantity, unit_price, payment, saving)

The meaning of each attribute is the same as that in the Requirements Document for RBMS.

The SQL statements for creating and populating these tables are provided in the file RBMSTablesScript.txt, which is uploaded to Brightspace. Copy RBMSTablesScript.txt file to your harveyv account and save it as RBMSTablesScript.sql. Run RBMSTablesScript.sql file to create a database required for this project. **No changes are allowed to these tables for this project.**

There are 20 statements in this project. If not otherwise specified in a question, you are asked to write a single SQL query for each statement. Some questions may ask you to write more than one SQL query in different styles. You are not allowed to create views or other (temporary) tables for this project. Inline views (i.e., select in the “from clause”) are allowed. Your query should take into consideration that the tuples currently in the tables may change. In other words, your solution to each question must be correct no matter what valid tuples are in the tables.

The query (or queries) for each statement is worth 5 points.

It is very important that you understand each query statement correctly. If you have any doubt about the correct interpretation of a statement, please ask the instructor for clarification through the Brightspace discussion forum for this project.

I suggest that you first test each query individually and save each query in a different file (with extension .sql) in your harveyv account (e.g., SQL> save query1). After all queries have been tested to your satisfaction and saved, 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. Follow the steps below to generate the spool file after you have logged into your Oracle account.

```
SQL> set echo on
```

```
SQL> spool project1.txt
```

```
SQL> start query1
```

```
.....
```

SQL> start query20

SQL> spool off

For each of the 20 statements, your output needs to show both the query and the result of the query. This can be achieved by “set echo on” as shown above. Before you submit the file project1.txt, you need to add your name, B-number, and the following statement *“I have done this assignment completely on my own. I have not copied it, nor have I given my solution to anyone else. I understand that if I am involved in plagiarism or cheating I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of 0 for the involved assignment and my grade will be reduced by one level for my first offense, and that I will receive a grade of “F” for the course for any additional offense of any kind.”* to the top of the file. Remember to (digitally) sign the statement before you submit the file.

The following are the 20 statements for this project:

1. Find the pid and name of each product that has a discount rate between 10% and 20% (i.e., including 10% and 20%) and can still sell 5 without the need to restock (i.e., $(qoh - 5) \geq qoh_threshold$).
2. Find the name of each customer who visited the retail business in October 2022 and whose phone# has an area code 888. The name is a concatenation of the first_name and the last_name. You can use to_char(pur_time, 'Month') and to_char(pur_time, 'YYYY') to extract month (e.g., October) and year (e.g., 2022) from pur_time, respectively.
3. Find the first_name and phone# of each customer who has made at least one purchase with a total price (i.e., payment) of at least 100 dollars in October 2022.
4. Find the pid and name of each product that is priced below 10 dollars (based on unit_price) and is purchased in December 2022 or October 2022.
5. Find the pur# of each purchase where a customer whose first name starts with “K” purchased a product whose original price is lower than 15 from an employee whose telephone’s area code is 888.
6. Find the pur#, product name (with header “product name”), pur_time, payment and saving of each purchase. pur_time must be displayed in a format as illustrated by the following example: March 10, 2023 Friday.
7. Find the eid and name of each employee who has not sold any products, i.e., not involved in any purchases. Write two queries for this statement with one having an uncorrelated subquery and another one having a correlated subquery.
8. Find the discount categories and their corresponding discount rates that have not been used on any product.

9. Find the name (i.e., the concatenation of the first name and last name) of each customer who has visited the retail business multiple times (i.e., more than once) but has not visited the business (i.e., has not made any purchase) in the most recent 200 days.
10. Find the eid and name of each employee who has sold all the products that have discount category 3.
11. Find the pid and name of each product that has an original price over \$180 and has been sold by all employees whose name starts with either an A or a D.
12. Find the cid of each customer who visited the retail business in August 2022 and has purchased all the products whose original price is between \$15 and \$20 (consider all purchases made by the customer, not just those made in August 2022).
13. Find the cid and name (i.e., the concatenation of the first name and last name) of each customer who has made at least one purchase that has the highest payment among all individual purchases. Note that it is possible for multiple purchases to have the same highest payment.
14. Find the pid and name of the product that has the highest discount per item in terms of absolute dollar amount (not necessarily the highest discount_rate). Also show this dollar amount (header: "discount amount"). It is possible that multiple products have the same highest discount amount. Use column format to change the header to "discount amount" (SQL> column discount_amount heading "discount amount").
15. For each month in chronological order (header: "Month"), output the total sale (header: "Total Sale") of the month (which is the sum of all payment for the month). Under "Month", show both month and year in the format as illustrated by "2022/08" for August 2022.
16. Find the cid and first name of each customer as well as the number of different types of products (i.e., products with different pids) purchased by the customer.
17. Find the cid and name (i.e., the concatenation of the first name and last name) of each customer who has visited the retail business more times than all customers whose phone# has an area code 777. Write an SQL query for this question without using any aggregate function.
18. Find the pid and name of each product that has been sold the most in terms of the total quantity. Also display the corresponding total quantity (header: "total quantity sold"). Note that it is possible that more than one product has been sold the same (highest) total quantity.
19. Find the cids and first names of the top-4 customers who have spent the most amounts of money at the retail business. Also show the total amount of money spent (header: "total amount spent") by each of these customers. If more than one customer is tied at the fourth place, showing anyone (just one) of them is acceptable.
20. Find the pid and name of the product that has not sold for the longest time (in number of days) among the products that have been sold before.