

Project Two

On-Line Grocery

Important Dates

- Due Date: April 15, 2020
- Cutoff Date: April 20, 2020
- Projects are submitted on or before the due date.
- No projects will be accepted after the cutoff date.
- 5 points will be deducted for each calendar day a project is submitted after the due date.

Objective

- Assist at risk New Yorkers to perform food shopping without visiting a crowded store. Hopefully, this will reduce the risk of infection.
- Create a database to track purchases, products and delivery for an on-line grocery store.
- Create searches and output using SQL

Database Requirements

Your design must include the following requirements:

- Customers purchase products from the grocery store. Identify the customer name, billing address, account, password and phone.
- Staff assist in the checkout, packing and delivery of customer orders. Identify the staff name, address, email and department. Staff can be assigned to many departments. In addition, identify current and all previous job titles and corresponding salary.
- Grocery products are saved in warehouses around NYC. Identify the warehouse location and warehouse type (refrigerated, kitchen used to prepare foods, etc.)
- Products are stored in warehouses. Identify the product name, product type (produce, fish, meat, dairy, etc.), calories, sodium, quantity, expiration date and warehouse where the product is stored. One product can be stored in multiple warehouses.
- Customers purchase products. Identify the product purchased, price, method of payment and delivery address.
- Customer orders are delivered. Identify the delivery address, staff who assisted with the delivery and delivery date.

Data

Enter *at least* the following data:

- 15 customers
- 5 staff
- 5 warehouses
- 20 products
- 10 orders

Questions

- Create SQL to answer the following queries.
 - Replace underlined items with values of your own choosing. For instance, replace the underlined terms Bo Li, \$1000, dairy in the last month with your own values.
 - Create data to ensure all questions generate output.
 - Format all output. For instance, all numbers will display with commas, dollar values will display with a \$ prefix and create descriptive labels for all columns.
1. Identify purchases for customer Bo Li this month. Display the customer name, products ordered, price and delivery date. Order the output by date and product.
 2. Identify customers who have not placed an order in the last year. Display the customer name and email. Order the output the name.
 3. Identify customers who made orders over \$1000 in the last month. Display the customer name and email. Display the customers who spent the most first.
 4. Identify the inventory of products by all locations. Display two columns: product name and number of products. Display one row for each distinct product. Display the output alphabetically by product.
 5. Identify dairy products not purchased in the last week. Display the product name, quantity and expiration date. Order the output by product name.
 6. Identify the most purchased products in the last month. Display three columns: product type, product name and number of purchases. Display one row for each distinct product type and product name. Display the product most purchased first.
 7. Using purchases made in the last month, identify customers with children. Display the customer name and email. Order the output by customer name. Note, you can replace children with other demographic characteristics. For instance, dog owners, seniors, vegetarians, Tesla car owners, etc.

8. Using purchases made in the last month, identify customers who are ill. Display two columns: customer zip code and number of customers. Display one row for each distinct zip code. Order the output by zip code.
9. Identify staff with the most deliveries in the last month. Display two columns: staff and number of deliveries. Display one row for each distinct staff. Display the staff with the most deliveries first.
10. Identify products with low inventory. Display the product name, warehouse location and quantity. Order the output by product name.
11. Display the structure of ALL tables using SQL Describe.
12. Display the version of Oracle. Enter:

```
SELECT *  
FROM v$version;
```

Additional Design Requirements

- Include all SQL commands to create your database and answer the questions including create tables, select, update, insert data, alter column names and alter column types.
- Create your database using Oracle version 18c. It will be discussed in class the scenarios when Oracle 11g can be used to complete the project. Projects created with other databases will be rejected and not graded. Utilizing other databases requires prior instructor approval.
- Normalize your database to third normal form.
- All multi-value columns must be saved to their own table.
- Output for all questions must include at least one row displayed.
- Identify and create primary keys for all tables.
- Create foreign keys to enforce referential integrity. For instance, you must have foreign keys with references to at least the following:
 - a. Staff and title
 - b. Product and warehouse
 - c. Staff and department
- Include the question, SQL command to answer the question and output from the SQL command.
- Create descriptive column labels for all output. For instance, don't display a column label named *count(*)*

Formatting

- The column output should be displayed in a non-proportional font such as courier. This will display the columns vertically straight.
- All columns in your search must display on one line. Don't wrap columns to two lines.
- Your project must be typed.
- All pages of your output must include the following in the header: name, class, date and project number.
- The first page of your project must include your name, the last four digits of your student id, class, submission date and the project number.

Submission

- Review the grading rubric on Blackboard to identify how the project will be evaluated and graded.
- Projects are due on the due date. No projects will be accepted after the cutoff date. Five points will be deducted for each calendar day, including weekends a project is submitted after the due date.
- An electronic copy of your project will be submitted to Blackboard on or before the due date. The file name uploaded to Blackboard will be in the format: [last name] [first name] Project2.docx. For example, *Smith Sally Project2.docx*.
- Submit one MS Word. For instance, don't submit separate files for create tables, insert and output.
- No projects will be accepted if sent to my email, left in my office mailbox or delivered to any other member of the department.
- If you submit multiple versions of the project, the last submitted project will be graded. Unless you receive prior approval, a project submitted before the due date and re-submitted after the due date is late.
- Unless you receive prior approval, projects submitted after the due date is late.
- Projects not in compliance with the submission requirements will be rejected and not graded.

Academic Integrity

You are required to comply with the *CUNY Policy on Academic Integrity* (<https://www.cuny.edu/wp-content/uploads/sites/4/page-assets/about/administration/offices/legal-affairs/policies-procedures/academic-integrity-policy/Academic-Integrity-Policy-051911.pdf>)

Projects and examinations must represent your own work. Group projects and exams are not permitted. Although you are encouraged to ask other students for information, you should neither copy another student's project nor permit another student to see your work. You can be asked to perform specific procedures and operations in the presence of the instructor. A student who submits a project that is too similar to another student's work will receive a ZERO for the project. Additional penalties may be imposed. Students found guilty of any form of academic dishonesty such as plagiarism or cheating on an exam or computer project are subject to discipline, including, but not limited to, failure in the course and suspension or dismissal from the College.