Project Three
On-Line Grocery

Timeline

- Due Date: May 13, 2020
- Cutoff Date: May 18, 2020 (without penalty)
- No projects will be accepted after the cutoff date.

Objective

- Modify the on-line grocery database to include product ratings, delivery satisfaction survey and tips to deliveries.
- Create reports
- Manage concurrency
- Manage security

Database requirements

Include all the On-Line Grocery database requirements from project two. In addition, add the following:

- Customers can rate products offered by the grocery. Track the customer who provides the rating, product rated, method of rating (number, stars, letter grade, etc.), date of rating and rating comment. Customers can enter open ended narrative text for each product sold by the grocery. For instance, "*The Kellogg's Raisin Bran cereal is great, although it doesn't include as many raisins as the Post Raisin Bran cereal."* One customer can rate many products.
- Customers can complete a survey of the order/delivery. Track the customer who completes the survey, date of survey, method of rating the survey (number, stars, letter grade, etc.) and rating comments. A customer can only complete one survey for each order.
- Customers can add an optional tip to the delivery.
- Security
    - Staff are restricted from accessing customer credit card numbers, expiration dates.
    - Customers can view past orders, but NOT change past orders.
    - Staff can't delete purchases after they are entered in the system.
- Include at least two sequence numbers to create unique values.

In addition to the data entered in project two, you must enter at least the following new data:

- At least 5 customer product surveys
- At least 5 purchase surveys
- At least 5 tips to purchases

Questions

- Disable the auto commit flag before performing all operations.
- Replace underlined items with values of your own choosing. For instance, replace the underlined terms last year with your own date range.
- 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.

- Be very descriptive explaining your results. For instance, use appropriate terminology, print screens and SQL

1. Identify customers who have not completed a purchase/delivery survey in the last 6 months. Display the customer name and email. Use a nested select to answer this question.
2. Identify the most popular product purchased in the last month. Display four columns: warehouse, product name, product type and number of orders. Display one distinct row for each warehouse, product and product type. Display the product with the most orders first.
3. Identify customers with the most purchases of fruit in the last year by customer location. Display five rows in your output – one row for each borough. Display three columns: borough, number of orders, total dollar amount of order. The borough with the most orders is displayed first. You may need multiple SQL to answer this question.
4. Identify customers with no comments in the product survey. Display the customer name.
5. Search the open-ended narrative text/comments in the product and delivery comments to identify personally identifiable information (PII). This includes any data that could potentially be used to identify a person. For instance, examples of PII include email address, date of birth, Social Security number, bank account number, home address, and full name. Display the customer who created the comment, date of comment and the comment. Order the output by customer name.
6. Customers can view, but not change past orders. Create SQL to implement. Demonstrate your implementation will not edit past orders by attempting to change data.
7. Staff are restricted from accessing customer credit card number and expiration dates. Create SQL to implement. Demonstrate your implementation will prevent staff from viewing customer credit card data.
8. Staff can't delete purchases after they are entered in the database. Create SQL to implement. Demonstrate your implementation will prevent staff from deleting purchases.
9. The product Raisin Bran is no longer being offered by the grocery store and being available for 3 years. Identify the SQL to implement.
10. In one SQL window, delete all customers. Don't commit. In another SQL window, add five new customers. Don't commit. In each SQL window, identify the number of customers. Explain your results. Disable the auto commit flag at the top of the window before performing this operation. Show all SQL to perform these operations. Demonstrate the functionally of your SQL by displaying the before and after results.
11. In one SQL window, delete all dairy products. Don't commit. In another SQL window, delete all products Don't commit. Explain your results. Resolve the problem. Create a backup of your table before implementing. To create a backup table, enter CREATE TABLE <NEWTABLE> AS SELECT * FROM <ORIGINALTABLE>; COMMIT; Then you can rename a table using the RENAME TABLE commit. Disable the auto commit flag at the top of the window before performing this operation. Show all SQL to perform these operations. Demonstrate the functionally of your SQL by displaying the before and after results.

12. In one SQL window, change the password for the customer Bo Li. Don't commit. In another SQL window, change the last name of customer Bo Li (use the same name as above). Don't commit. Quit both Oracle sessions. Login to Oracle again and display all columns for the customer Bo Li . Explain your results. Disable the auto commit flag at the top of the windows before performing this operation. Show all SQL to perform these operations. Demonstrate the functionally of your SQL by displaying the before and after results.
13. Use the SQL DESCRIBE operation to display the structure for all tables.
14. 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 11g or 18c. 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.
- 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.
- 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.
- 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] Project3.docx. For example, *Smith Sally Project3.docx.*
- Submit one MS Word file. 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

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. You are required to comply with the CUNY Policy on Academic Integrity available at
http://www.cuny.edu/about/administration/offices/la/Academic_Integrity_Policy.pdf