**MIS 502**

**SQL Project: Instacart DB and Data Analytics**

**Background:**

Instacart is an American company that operates as a same-day grocery delivery service. Customers select groceries through a web application from various retailers and delivered by a personal shopper. Instacart's service is mainly provided through a smartphone app, available on iOS and Android platforms, apart from its website.

**Data description:**

Instacart provides 6 tables: products, aisles, departments, orders, orders\_product\_train, and orders\_product\_prior. Please download the data set from the shared Box folder.

The data set include orders of 200,000 Instacart users with each user having between 4 and 100 orders. Instacart indicates each order in the data as prior, train or test in Eval\_set column.

orders.csv has all the information about the given order id, the user who has purchased the order, when was it purchased, days since prior order and so on. Each user has purchased various products during their orders.

In this dataset, 4 to 100 orders of a customer are given. The last order of the user has been taken out and placed into order\_products\_train file. All the prior order information of the customer is present in order\_products\_prior file. Eval\_set column in orders.csv tells us to which of the three datasets (prior, train or test) the given row goes to. Note that we don’t have order\_products\_test file. Not having this part of data will not influence our project.

The following is the information about the tables:

`orders` (>3.4m rows, 206k users):

\* `order\_id`: order identifier (primary key)

\* `user\_id`: customer identifier

\* `eval\_set`: which evaluation set this order belongs in (see `SET` described below)

\* `order\_number`: the order sequence number for this user (1 = first, n = nth)

\* `order\_dow`: the day of the week the order was placed on

\* `order\_hour\_of\_day`: the hour of the day the order was placed on

\* `days\_since\_prior`: days since the last order, capped at 30 (with NAs for `order\_number` = 1)

where `SET` is one of the three following evaluation sets (`eval\_set` in `orders`):

\* `"prior"`: orders prior to that user’s most recent order (~3.2m orders)

\* `"train"`: training data supplied to participants (~131k orders)

\* `"test"`: test data reserved for machine learning competitions (~75k orders)

Sample data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| order\_id | user\_id | eval\_set | order\_number | order\_dow | order\_hour\_of\_day | days\_since\_prior\_order | |
| 2539329 | 1 | prior | 1 | 2 | 8 |  |  |
| 2398795 | 1 | prior | 2 | 3 | 7 | 15 |  |
| 473747 | 1 | prior | 3 | 3 | 12 | 21 |  |
| 2254736 | 1 | prior | 4 | 4 | 7 | 29 |  |
| 431534 | 1 | prior | 5 | 4 | 15 | 28 |  |
| 3367565 | 1 | prior | 6 | 2 | 7 | 19 |  |
| 550135 | 1 | prior | 7 | 1 | 9 | 20 |  |
| 3108588 | 1 | prior | 8 | 1 | 14 | 14 |  |
| 2295261 | 1 | prior | 9 | 1 | 16 | 0 |  |
| 2550362 | 1 | prior | 10 | 4 | 8 | 30 |  |
| 1187899 | 1 | train | 11 | 4 | 8 | 14 |  |
| 2168274 | 2 | prior | 1 | 2 | 11 |  |  |
| 1501582 | 2 | prior | 2 | 5 | 10 | 10 |  |

Note: in order\_dow, 0 is Saturday and 1 stands for Sunday.

`products` (about 50k rows):

\* `product\_id`: product identifier (primary key)

\* `product\_name`: name of the product

\* `aisle\_id`: foreign key

\* `department\_id`: foreign key

Sample data

|  |  |  |  |
| --- | --- | --- | --- |
| product\_id | product\_name | aisle\_id | department\_id |
| 1 | Chocolate Sandwich Cookies | 61 | 19 |
| 2 | All-Seasons Salt | 104 | 13 |
| 3 | Robust Golden Unsweetened Oolong Tea | 94 | 7 |
| 4 | Smart Ones Classic Favorites Mini Rigatoni With Vodka Cream Sauce | 38 | 1 |
| 5 | Green Chile Anytime Sauce | 5 | 13 |
| 6 | Dry Nose Oil | 11 | 11 |
| 7 | Pure Coconut Water With Orange | 98 | 7 |
| 8 | Cut Russet Potatoes Steam N' Mash | 116 | 1 |
| 9 | Light Strawberry Blueberry Yogurt | 120 | 16 |
| 10 | Sparkling Orange Juice & Prickly Pear Beverage | 115 | 7 |
| 11 | Peach Mango Juice | 31 | 7 |
| 12 | Chocolate Fudge Layer Cake | 119 | 1 |
| 13 | Saline Nasal Mist | 11 | 11 |

`aisles` (134 rows):

\* `aisle\_id`: aisle identifier (primary key)

\* `aisle`: the name of the aisle

Sample data

|  |  |
| --- | --- |
| aisle\_id | aisle |
| 1 | prepared soups salads |
| 2 | specialty cheeses |
| 3 | energy granola bars |
| 4 | instant foods |
| 5 | marinades meat preparation |
| 6 | other |
| 7 | packaged meat |
| 8 | bakery desserts |
| 9 | pasta sauce |
| 10 | kitchen supplies |

`departments` (21 rows):

\* `department\_id`: department identifier (primary key)

\* `department`: the name of the department

Sample data

|  |  |
| --- | --- |
| department\_id | department |
| 1 | frozen |
| 2 | other |
| 3 | bakery |
| 4 | produce |
| 5 | alcohol |
| 6 | international |
| 7 | beverages |
| 8 | pets |
| 9 | dry goods pasta |
| 10 | bulk |
| 11 | personal care |
| 12 | meat seafood |
| 13 | pantry |
| 14 | breakfast |
| 15 | canned goods |
| 16 | dairy eggs |
| 17 | household |
| 18 | babies |
| 19 | snacks |
| 20 | deli |
| 21 | missing |

`order\_products\_train ` (1384617 rows):

\* `order\_id`: foreign key

\* `product\_id`: foreign key

\* `add\_to\_cart\_order`: order in which each product was added to cart

\* `reordered`: 1 if this product has been ordered by this user in the past, 0 otherwise

It has a composite primary key (order\_id and product\_id) and indicates whether a product in an order is a reorder or not (through the reordered variable).

Sample data

|  |  |  |  |
| --- | --- | --- | --- |
| order\_id | product\_id | add\_to\_cart\_order | reordered |
| 2 | 33120 | 1 | 1 |
| 2 | 28985 | 2 | 1 |
| 2 | 9327 | 3 | 0 |
| 2 | 45918 | 4 | 1 |
| 2 | 30035 | 5 | 0 |
| 2 | 17794 | 6 | 1 |
| 2 | 40141 | 7 | 1 |
| 2 | 1819 | 8 | 1 |
| 2 | 43668 | 9 | 0 |
| 3 | 33754 | 1 | 1 |
| 3 | 24838 | 2 | 1 |
| 3 | 17704 | 3 | 1 |
| 3 | 21903 | 4 | 1 |
| 3 | 17668 | 5 | 1 |
| 3 | 46667 | 6 | 1 |
| 3 | 17461 | 7 | 1 |
| 3 | 32665 | 8 | 1 |
| 4 | 46842 | 1 | 0 |
| 4 | 26434 | 2 | 1 |

`order\_products\_prior ` (32434489 rows):

\* `order\_id`: foreign key

\* `product\_id`: foreign key

\* `add\_to\_cart\_order`: order in which each product was added to cart

\* `reordered`: 1 if this product has been ordered by this user in the past, 0 otherwise

This table includes prior orders. It has a composite primary key (order\_id and product\_id) and indicates whether a product in an order is a reorder or not (through the reordered variable).

Sample data

|  |  |  |  |
| --- | --- | --- | --- |
| order\_id | product\_id | add\_to\_cart\_order | reordered |
| 1 | 49302 | 1 | 1 |
| 1 | 11109 | 2 | 1 |
| 1 | 10246 | 3 | 0 |
| 1 | 49683 | 4 | 0 |
| 1 | 43633 | 5 | 1 |
| 1 | 13176 | 6 | 0 |
| 1 | 47209 | 7 | 0 |
| 1 | 22035 | 8 | 1 |
| 36 | 39612 | 1 | 0 |
| 36 | 19660 | 2 | 1 |
| 36 | 49235 | 3 | 0 |
| 36 | 43086 | 4 | 1 |
| 36 | 46620 | 5 | 1 |
| 36 | 34497 | 6 | 1 |
| 36 | 48679 | 7 | 1 |
| 36 | 46979 | 8 | 1 |
| 38 | 11913 | 1 | 0 |
| 38 | 18159 | 2 | 0 |
| 38 | 4461 | 3 | 0 |
| 38 | 21616 | 4 | 1 |
| 38 | 23622 | 5 | 0 |

**Project Requirements:**

**This project contains two parts:**

**Part A:**

1. Create an Instacart DB to contain these tables and data,
2. After developing the database, create an EER diagram to show the relationships between tables,
3. You may also need to create index to improve the efficiency of queries, and
4. Save all SQL statements that you use to create tables, loading data, and create index in projectdb.sql file.

Note: You can create the database using Amazon cloud server or on your local MySQL server instance. It is better to use the local server instance because the network connection may not be reliable.

**Part B:** Explore the database and write queries to answer the following questions. For some questions, you may use multiple queries. Please paste a screen shot of the result set of each question in a word document. Save all your SQL queries for Part B in PartB.sql file.

Your project reports shall contain the following three items: projectdb.sql, partB.sql, and the word document containing the screen shots.

**Part B Query Questions:**

Q1. Which weekday (including weekends) has the highest number of orders? Which weekday has the lowest number of orders?

Q2. What percentage of orders are made during daytime (8am-5pm)? Round the result to 2 digits to decimal.

Q3. (a) If the company wants to give discount for customers’ reorders. At what time should the company launch the discount event? Find the top 3 prime times for re-orders. Prime time is measured by the reorder counts. Your results should look like Wednesday 3am, etc.

(b). The company wants to attract new customers by launching promotion events. At what time should the company launch the promotion to customers who place his/her first order? Find the top 3 prime times for customers’ first orders.

Q4. How often do the users reorder items? To answer this question, you need to show the number of users reorder items for each days\_since\_prior.

Q5. Show how many customers reorder once in every week, two weeks, three weeks, or once in every month, etc.

Q6. Use order\_products\_prior and order\_products\_train to answer this question. The company wants to know on average how many items and how many products users buy. Round the results to Integer. Do you see the distributions are comparable between the train and prior order set? Note that add\_to\_cart\_order show the purchased items. It is possible that a customer purchased multiple items of a product in an order.

Q7. What are the top 10 products most often ordered? Show the product names of these products. Note: You need to add the results order\_products\_prior and order\_products\_train tables.

### Q8. For each of the top 5 users who placed the highest number of orders, what is the average days interval of this user’s orders?

### Q9. Show days\_since\_prior and the average reorder rate of each days\_since\_prior. Round the average of reordered as 2 digits to decimal. Sort the result set by days\_since\_prior.

### Q10. We want to know which product people put into the cart first if they buy products? To answer this question, find the product\_id, product\_name, and the highest percentage of this product’s put-into-the-cart-first.

Q11. Are the top 5 products with the highest number of orders more likely to be reordered? Note that if the proportion of reordered is >70%, then this product is more likely to be reordered.

Hint: To answer this question, you need to write a query to show the number of orders per product

and the number of reorders per product. Then calculate the proportion of reordered.

Q12. Are organic products sold more often than non-organic products? You can solve this question by showing the percentage of orders that have organic products? Product\_name describes whether a product is organic or not.

Q13. How many unique products are offered in each department/aisle?

### Q14. Find the top 10 best-sellers in each department.

### Q15. Find the top 10 best-sellers in each aisle.

### Q16. Show the number of new users (i.e., customers place the first orders), and the number of existing users, and the ratio of new users to existing users in each weekday. Which day has the highest ratio?

### Q17. How many customers always reorder the same products all the time?

Hint: To search the users you need to look at all orders (excluding the first order), where the percentage of reordered items is exactly 1.

#### Q18. Segment the customers based on their average days of interval of reordering into 4 segments. Count the number of users in each segment.

#### Q19. For those customers who reordered within 7 days, what are the most frequently reordered products?

#### Show the top 5 products' product\_id and product\_name.

#### Q20. The manager thinks that the longer the length of days interval between reorders, the more users will purchase in the next reorder. Do you agree with the manager? Explain why.