# SQL TEST

# 3

DATA SCIENCE INFINITY

# THE DATA

For this test, we have 5 tables that can be accessed in the grocery_db schema of the DATA SCIENCE INFINITY database

Example data from each table can be seen below:

## customer_details

| customer_id | distance_from_store | gender | credit_score |
|---|---|---|---|
| 754 | 1.17 | M | 0.75 |
| 843 | 4.84 | | |
| 749 | 1.74 | M | 0.65 |
| 426 | 4.38 | F | 0.57 |
| 560 | | M | 0.54 |

## transactions

| customer_id | transaction_date | transaction_id | product_area_id | num_items | sales_cost |
|---|---|---|---|---|---|
| 642 | 2020-04-01 | 435561233435 | 4 | 3 | 9.44 |
| 642 | 2020-04-01 | 435561233435 | 3 | 5 | 23.82 |
| 493 | 2020-07-15 | 436618008621 | 4 | 1 | 6.83 |
| 493 | 2020-07-15 | 436618008621 | 3 | 9 | 9.33 |
| 493 | 2020-07-15 | 436618008621 | 5 | 1 | 8.50 |

## campaign_data

| customer_id | campaign_name | campaign_date | mailer_type | signup_flag |
|---|---|---|---|---|
| 74 | delivery_club | 2020-07-01 | Mailer2 | 1 |
| 655 | delivery_club | 2020-07-01 | Mailer2 | 0 |
| 607 | delivery_club | 2020-07-01 | Mailer2 | 1 |
| 788 | delivery_club | 2020-07-01 | Control | 0 |
| 405 | delivery_club | 2020-07-01 | Mailer1 | 0 |

## product_areas

| product_area_id | product_area_name | profit_margin |
|---|---|---|
| 1 | Non-Food | 0.25 |
| 2 | Vegetables | 0.18 |
| 3 | Fruit | 0.14 |
| 4 | Dairy | 0.19 |
| 5 | Meat | 0.11 |

## loyalty_scores

| customer_id | customer_loyalty_score |
|---|---|
| 104 | 0.587 |
| 69 | 0.156 |
| 525 | 0.959 |
| 181 | 0.418 |
| 796 | 0.57 |

# THE QUESTIONS

1) Return a list of customers from the *loyalty_scores* table who have a *customer_loyalty_score* of 0.77, 0.88, or 0.99

2) Return the average *customer_loyalty_score* for customers, split by *gender*

3) Return *customer_id*, *distance_from_store*, and a new column called *distance_category* that tags customers who are less than 1 mile from store as "Walking Distance", 1 mile or more from store as "Driving Distance" and "Unknown" for customers where we do not know their distance from the store

4) For the 400 customers with a *customer_loyalty_score*, divide them up into 10 deciles, and calculate the average *distance_from_store* for each decile

5) Return data showing, for each *product_area_name* - the total sales, and the percentage of overall sales that each product area makes up