✓ **Milestone 7** | FastKitchen Customers

INTRODUCTION: In this SkillBuilder, you've learned additional ways of joining tables together, with three different types of outer join: the left join, right join, and the full outer join. While an inner join retains only information when there's a match between the joined tables, an outer join will also output information that can only be found in one table.

FastKitchen is a fictitious restaurant and the dataset you'll be working with is constructed. While this dataset might not represent real data, it emulates characteristics of real data. When you're interviewing for a job, you might be asked to look at this kind of data to show off your skills in a context related to the company and the job position!

HOW IT WORKS: Follow the prompts in the questions below to investigate your data. Post your answers in the provided boxes: the **yellow boxes** for the queries you write, and **blue boxes** for text-based answers. When you're done, export your document as a pdf file and submit it on the Milestone page – see instructions for creating a PDF at the end of the Milestone.

RESOURCES: If you need hints on the Milestone or are feeling stuck, there are multiple ways of getting help. Attend Drop-In Hours to work on these problems with your peers, or reach out to the HelpHub if you have questions. Good luck!

PROMPT: In this Milestone, you'll step into the shoes of a data contractor who is helping a new fast-food restaurant understand their customer base. You will need to make use of one type of outer join to help the restaurant manager combine information about their customers. These customers include registered customers who have accounts on the restaurant's website, and guest customers who do not register for accounts.

SQL App: Here's that link to our specialized SQL app, where you'll write your SQL queries and interact with the data.

– Data Set **Description**

The data in this Milestone (`fastkitchen.*`) depicts orders made at a fictional takeout-only fast food restaurant in the Midwestern United States. The restaurant has an online site where customers can put in orders for carryout or delivery; customers can also make orders offline at the restaurant's storefront. You will be working with two tables in this Milestone: orders and users.

Each row in the orders table is a single order that was placed at the restaurant. This table has seven columns:

- **`order_id`** - unique order id, primary key
- **`timestamp`** - when the order was made
- **`user_id`** **-** user_id for registered accounts, blank if guest customer
- **`order_type`** - whether the order was made onsite, online carryout, or online delivery
- **`subtotal`** - base amount for the order
- **`tip`** - amount of tip, if any, left by the customer
- **`total`** - subtotal + tip

Customers have the option of creating a user account, which can be used both in person and online. The users table has five columns:

- **`user_id`** - unique user_id value, primary key
- **`reg_timestamp`** - when the user registered their account
- **`city`** - user city
- **`state`** - two-letter code for state
- **`zip`** - zip code

– **Task 1:** Explore information about orders.

To start off, let's warm up with some questions on the individual tables, before we ask questions that require joining the two tables together. Let's look at the `orders` table first.

A. What is the average total amount (including tips) spent per order?

(paste your query below ≣)

```
SELECT
  AVG(total)
FROM
  fastkitchen.orders
```

(write your **answer** below ≣)

The average total amount spent per order is 22.170.

B. Compare the average subtotals, tips, and totals spent by each order type (onsite, carryout, delivery). Are there any major differences between order types?

(paste your query below ≣)

```
SELECT
  order_type,
  AVG(subtotal) AS subtotal,
  AVG(tip) AS tips,
  AVG(total) AS total
FROM
  fastkitchen.orders
GROUP BY
  order_type
```

(write your **answer** below ≣)

There are no major differences between order types. The biggest difference between any two order types is less than 0.5.

C. Write a query to count the number of orders made by **registered users**.

(paste your query below ≣)

```
SELECT
  (SELECT COUNT(order_id) FROM fastkitchen.orders
WHERE user_id IS NOT NULL) AS orders_by_rUsers
```

D. Now, write a query to return the number of orders made by **non-registered customers**. Remember, non-registered customers don't have a user id.

Compared alongside the returned count from the query in C, which group is larger?

(paste your query below ≣)

```
SELECT
  (SELECT COUNT(order_id) FROM fastkitchen.orders
WHERE user_id IS NULL) AS orders_by_rUsers
```

(write your **answer** below ≣)

The total orders made by non-registered customers are 2088 and 1932 for registered users. Non-registered customers made more orders.

– **Task 2:** Explore information about registered users.

Next, we'll check out the `users` table.

A. Write a query that counts the number of users by city. Which city has the highest number of users, and how many users are there?

(paste your query below ≣)

```
SElECT
  city,
  COUNT(user_id)
FROM
  fastkitchen.users
GROUP BY
  city
```

(write your **answer** below ≣)

Allen has the most with 212 users.

B. Expand the query so that you group by zip code as well. Does this help explain what you found in part 2A?

(paste your query below ≣)

```
SElECT
  city,
  zip,
  COUNT(user_id)
FROM
  fastkitchen.users
GROUP BY
```

```
  city,
  zip
```

(write your **answer** below ☰)

Yes, assuming each city's zip code is part of the data set, Allen having three zip codes probably means its area or population is larger than the other cities.

– **Task 3:** How do orders compare between zip codes and cities?

Finally, we'll combine the `user` and `orders` tables into a single, joined table.

A. To start, simply write a query that returns all of the columns, joining the two tables on the **`user_id`** column.

Make sure that you choose a join that keeps all of the orders, even when there isn't a matching registered user.

(paste your query below ☰)

```
SELECT
  a.*,
  b.*
FROM
  fastkitchen.orders AS a
  LEFT OUTER JOIN fastkitchen.users AS b ON
a.user_id = b.user_id
```

B. Add to the query from 3A to answer the following question: in which zip code is the user with the highest amount of money spent?

(paste your query below ≡)

```
SELECT
  SUM(a.total) AS total_spent,
  a.user_id,
  b.zip
FROM
  fastkitchen.orders AS a
  LEFT OUTER JOIN fastkitchen.users AS b ON a.user_id =
b.user_id
WHERE
  a.user_id IS NOT NULL
GROUP BY
  b.zip,
  a.user_id
ORDER BY
  total_spent DESC
```

(write your **answer** below ≡)

The zip code of the user who spent the most money is 63222 (whose id and zip is known).

C. Write a query that returns the average total amount spent per order by zip code.

How many of the zip codes spend more on average than non-registered guest customers?

HINT: The null zip code represents non-registered guests!

(paste your query below ≣)

```
SELECT
  b.zip,
  AVG(a.total) AS avg_tSpent,
  COUNT(a.order_id) AS num_orders
FROM
  fastkitchen.orders AS a
  LEFT OUTER JOIN fastkitchen.users AS b ON a.user_id =
b.user_id
GROUP BY
  b.zip
ORDER BY
  avg_tSpent DESC
```

(write your **answer** below ≣)

Three zip codes spend more on average than non-registered users. Those are 63216, 63215, and 63222.

– Submission

Great work completing this Milestone! To submit your completed Milestone, you will need to download / export this document as a PDF and then upload it to the Milestone submission page. You can find the option to download as a PDF from the File menu in the upper-left corner of the Google Doc interface.