✓ **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?

```
SELECT
  ROUND(avg(total), 2) as avg_total
from
  fastkitchen.orders
```

The average total amount including tips per order spent is **$22.22.**

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

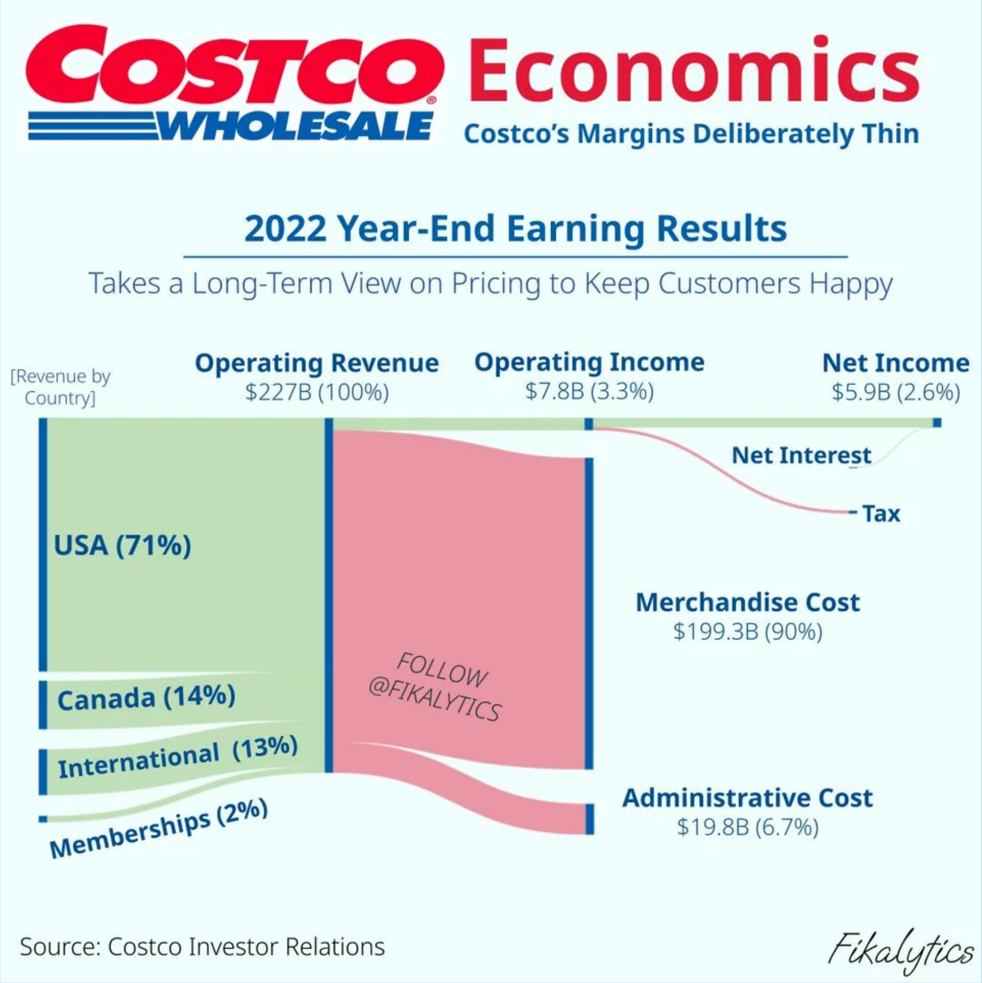
```
SELECT
  order_type,
  ROUND(AVG(subtotal), 2) AS avg_subtotal,
  ROUND(AVG(tip), 2) AS avg_tips,
  ROUND(AVG(total), 2) AS avg_total
FROM
  fastkitchen.orders
GROUP BY
  order_type
ORDER BY
  avg_subtotal
```

There is a major difference between order types. **Deliveries** have the **highest average subtotal** at a total of $20.61 and **Carry Out** has the lowest average subtotal at a total of $20.16 for a difference of $0.45.

There is a **2.19%** difference in pricing from **Deliveries** to **Carry Out.** This is only in reference to Pizza; however if this was a business like Costco that does 227B dollars in Operating Revenue, **2.19%** of **$227B** would be **$6.066B.**

So to answer the question, "Is there any major difference between order types," I would stand on the side of Yes, that **$0.45** relative to the **total amounts** is a major factor.

Here is a graph to reference Costcos 2022 Revenue (*And while I don't know how accurate the graph being portrayed is, the point still stands that I'm making that 2.19% of anything is a major difference.)*:



Source: Costco Investor Relations

(Another Example is ***Eliud Kipchoge*** *breaking the 2 Hour barrier running a full marathon. His time was* ***1:59:40****, and to break that number down further, if he ran the marathon* ***2.19% slower****, his time would be* ***2:13:11.*** This number is a whole **13 minutes away** from being under 2 hours!)

C. Write two different queries to count the number of orders made by registered users and the number of orders made by non-registered customers. Remember, non-registered customers don't have a user id. Which group is larger?

```
SELECT
  user_id
FROM
  fastkitchen.orders
WHERE
  user_id IS NULL
```

```
SELECT
  user_id
FROM
  fastkitchen.orders
WHERE
  user_id IS NOT NULL
```

The number of orders made by **Registered Customers** is **1932** orders. The number of orders made by **Non-Registered Customers** is **2088.**

If the goal of the business was to get as many people registered as possible, it would be important to see these numbers to have a visual as to how many people who order on their website are actually registered compared to those who are not registered.

In this example, Non-Registered Customers outweigh the Registered customers 174:161. This gives insight that somehow, some way, and for some reason, customers actually find it EASIER not to register and have an account with a company.

My personal experience with signing into websites for food is met with annoying sign-ins and typing in a password and email every time I want to order.

Thus, if a company wants more registered accounts, they need to make signing up and signing in, a task that would be a waste NOT to do.

Take for example a company like Jersey Mikes, where its promotion of adding a phone number allows for a customer to earn “points” toward a free sandwich. It makes more sense to spend the extra 60 seconds signing in to get points to earn a free food item, then to waste the money buying a sandwich that could have been purchased for free.

More companies are starting to implement a point system method like Chipotle or Buffalo Wild Wings. These are all just food places I know about that have point systems, and for me personally, they work towards having me sign into their website, even if I did not plan on spending my time to do it in the first place.

(*Also, about the code itself, I’m not proud of the solutions and I feel as though there is a better solution I could not come up with*).

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

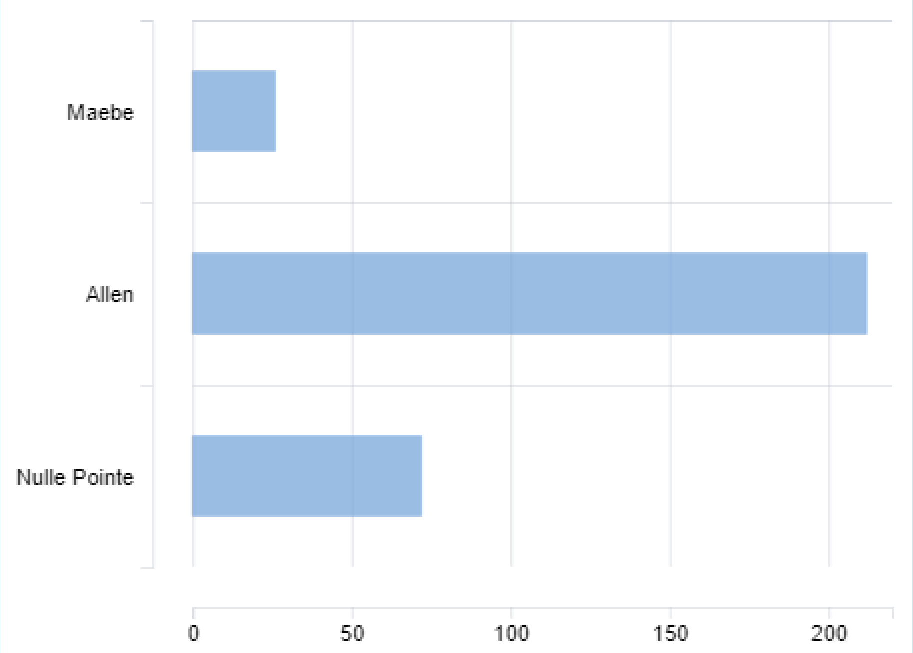
Next, we’ll check out the `users` table.

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?

```
SELECT
  city,
  count(user_id) as n_users
FROM
```

```
  fastkitchen.users
GROUP BY
  city
```

Allen City has the highest number at 212 users. Nulle Pointe has the second highest at 72 users. Maebe has the lowest number of users at 26.



Visually the difference is drastic.

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

```
SELECT
  city,
  zip,
  count(user_id) as n_users
FROM
  fastkitchen.users
GROUP BY
  city,
  zip
ORDER BY
  city DESC
```

This new data information provides a reason why Allen has more users. There are three zip codes within this city. This helps me reasonably assume that Allen is a bigger city than Maebe and Nulle Pointe.

Thus, if we were to assume it’s a bigger city, it makes sense that it also has the most users.

So it would make sense that each zip code within Allen has roughly the same amount as the Nulle Pointe zip code.

Statistically, while these numbers are skewed and far apart from each other, mathematically none of these pointers are outliers from one another.

Here is a link to a Google Drive Document to review the Math: https://docs.google.com/document/d/1WqXb5W0A0oWXiVLddn8Oa4WIRkk9n2ip-3IMoyZMkv4/edit?usp=sharing

– **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.

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

No question is being asked.

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?

```
SELECT
  b.total,
  a.zip
FROM
  fastkitchen.users AS a FULL
  OUTER JOIN fastkitchen.orders AS b ON a.user_id = b.user_id
```

```
ORDER BY
  total DESC
```

The zip code with the user with the highest amount of money spent lives in 63216 with a total of $92.15.

Please note that this is the first recorded total with a zip code attached to the total amount. The highest total without a zip code is $95.80.

It is also worth noting that within the top 5 totals where the zip code is connected to the amount, 4 out of the 5 zip codes are 63216.

It is also worth noting that the zip code 63216 is located in Allen. Which was previously determined to be the city with the most users. There is a correlation between the most users within the city and the highest total amount spent on an order.

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?

```
SELECT
  a.zip,
  ROUND(avg(b.total), 2) AS avg_total
FROM
  fastkitchen.users AS a FULL
  OUTER JOIN fastkitchen.orders AS b ON a.user_id = b.user_id
GROUP BY
  a.zip
```

```
ORDER BY
  avg_total DESC
```

The number of zip codes that spend more on average than non-registered guest customers is 3 zip codes.

The three listed zip codes below **spend more** on average compared to non-registered guest customers:

63216 → Allen
63215 → Allen
63222 → Nulle Pointe

The 2 listed zip codes below **spend less** on average compared to non-registered guest customers:

63225 → Maebe
63218 → Allen

– 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.