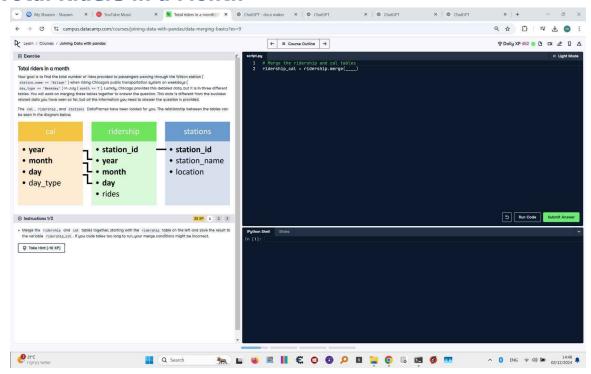
## **Total Riders in a Month**



# **Question:**

Your goal is to find the total number of rides provided to passengers passing through the Wilson station (station\_name == 'Wilson') when riding Chicago's public transportation system on weekdays ('day\_type == Weekday') in July (month == 7). Luckily, Chicago provides this detailed data, but it is in three different tables. You will work on merging these tables together to answer the question.

#### **Tables Provided:**

- `cal`: Contains columns `year`, `month`, `day`, and `day\_type` (e.g., Weekday, Weekend).
- `ridership`: Contains columns `station\_id`, `year`, `month`, `day`, and `rides`.
- `stations`: Contains columns `station\_id`, `station\_name`, and `location`.

### **Instructions:**

- 1. Merge the `ridership` and `cal` tables together, starting with the `ridership` table on the left, and save the result to a variable `ridership\_cal`.
- 2. Merge the `ridership\_cal` table with the `stations` table, and filter for rows where `station name` is 'Wilson'.

3. Further filter the data for weekdays in July and calculate the total number of rides.

### **Answer:**

# **Explanation of the Code:**

(filtered data['month'] == 7)]

print(total rides)

total rides = filtered data['rides'].sum()

- 1. `ridership.merge(cal, on=['year', 'month', 'day'])`: This merges the `ridership` table with the `cal` table on common columns `year`, `month`, and `day`, combining date-specific information.
- 2. `ridership\_cal.merge(stations, on='station\_id')`: This merges the result with the `stations` table on the `station\_id` column to add station-specific details. The result is filtered for rows where `station name` is 'Wilson'.
- 3. The final filter applies conditions for `day\_type` (Weekday) and `month` (July), and the total number of rides is calculated using the `.sum()` method on the `rides` column.