

# Total Riders in a Month

The screenshot shows a web browser with multiple tabs, including 'Total riders in a month'. The active page is a DataCamp exercise titled 'Total riders in a month'. The exercise text states: '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. This data is different from the business related data you have seen so far, but all the information you need to answer the question is provided.' Below the text, three DataFrames are listed: 'cal' (columns: year, month, day, day\_type), 'ridership' (columns: station\_id, year, month, day, rides), and 'stations' (columns: station\_id, station\_name, location). A diagram shows 'cal' and 'ridership' connected by 'year', 'month', and 'day', and 'ridership' and 'stations' connected by 'station\_id'. The instructions section says: 'Merge the ridership and cal tables together, starting with the ridership table on the left and save the result to the variable ridership\_cal. If you code takes too long to run, your merge conditions might be incorrect.' There is a 'Take Hint (-10 XP)' button. On the right, a code editor shows the following code: 

```
1 # Merge the ridership and cal tables
2 ridership_cal = ridership.merge(_____)
```

 Below the code editor is a 'Python Shell' area with a prompt 'In [1]:'. The interface also shows a 'Course Outline' button, 'Daily XP 452', and a 'Run Code' button.

## 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:

# Step 1: Merge `ridership` and `cal` tables

```
ridership_cal = ridership.merge(cal, on=['year', 'month', 'day'])
```

# Step 2: Merge `ridership\_cal` with `stations` and filter for 'Wilson'

```
ridership_cal_stations = ridership_cal.merge(stations, on='station_id')
```

```
filtered_data = ridership_cal_stations[ridership_cal_stations['station_name']  
== 'Wilson']
```

# Step 3: Filter for weekdays in July and calculate total rides

```
filtered_data = filtered_data[(filtered_data['day_type'] == 'Weekday') &
```

```
(filtered_data['month'] == 7)]
```

```
total_rides = filtered_data['rides'].sum()
```

```
print(total_rides)
```

### Explanation of the Code:

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.