

# Total Riders in a Month

The screenshot displays a web browser window with multiple tabs. The active tab is titled "Total riders in a month" and shows a datacamp exercise page. The page content includes a task description, a diagram of three data tables, and a code editor. The browser's address bar shows the URL "campus.datacamp.com/courses/joining-data-with-pandas/data-merging-basics/ex-9". The browser's taskbar at the bottom shows various application icons and the system clock indicating 14:48 on 02/12/2024.

**Exercise**

### Total riders in a month

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.

The `cal`, `ridership`, and `stations` DataFrames have been loaded for you. The relationship between the tables can be seen in the diagram below.

cal	ridership	stations
<ul style="list-style-type: none"><li>• year</li><li>• month</li><li>• day</li><li>• day_type</li></ul>	<ul style="list-style-type: none"><li>• station_id</li><li>• year</li><li>• month</li><li>• day</li><li>• rides</li></ul>	<ul style="list-style-type: none"><li>• station_id</li><li>• station_name</li><li>• location</li></ul>

**Instructions 1/3**

- 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.

[Take Hint \(-10 XP\)](#)

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

[Run Code](#) [Submit Answer](#)

**Python Shell** [Global](#)

```
In [1]:
```

## 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.