Escribe un código para analizar los datos sobre el clima en Chicago en noviembre de 2017 desde el sitio web:

[https://practicum-content.s3.us-west-1.amazonaws.com/data-analyst-eng/moved_chicago_weather_2017.html

El nombre del DataFrame debe ser weather_records y tienes que especificarlo cuando buscas: attrs={"id": "weather_records"} . Imprime el DataFrame completo.

```python
import requests
from bs4 import BeautifulSoup
import pandas as pd
URL='https://practicum-content.s3.us-west-1.amazonaws.com/data-analyst-eng/moved_chicago_weather_2017.html'
req = requests.get(URL)
soup = BeautifulSoup(req.text, 'lxml')
table = soup.find('table', attrs={"id": "weather_records"})
heading_table=[]
for row in table.find_all('th'):
    heading_table.append(row.text)
content=[]
for row in table.find_all('tr'):
    if not row.find_all('th'):
        content.append([element.text for element in row.find_all('td')])
weather_records = pd.DataFrame(content, columns = heading_table)
print(weather_records)
```

Encuentra la cantidad de viajes para cada empresa de taxis cuyo nombre contenga las palabras "Yellow" o "Blue" del 1 al 7 de noviembre de 2017. Nombra la variable resultante *trips_amount.* Agrupa los resultados por el campo *company_name*.

```sql
SELECT
    cabs.company_name AS company_name,
    COUNT(trips.trip_id) AS trips_amount
FROM
    trips
JOIN
    cabs
ON
    trips.cab_id = cabs.cab_id
WHERE
    (cabs.company_name LIKE '%Yellow%' OR cabs.company_name LIKE '%Blue%')
    AND DATE(trips.start_ts) BETWEEN '2017-11-01' AND '2017-11-07'
GROUP BY
    cabs.company_name
ORDER BY
    trips_amount DESC;

```

Del 1 al 7 de noviembre de 2017, las empresas de taxis más populares fueron Flash Cab y Taxi Affiliation Services. Encuentra el número de viajes de estas dos empresas y asigna a la variable resultante el nombre *trips_amount.* Junta los viajes de todas las demás empresas en el grupo "Other". Agrupa los datos por nombres de empresas de taxis. Asigna el nombre *company* al campo con nombres de empresas de taxis. Ordena el resultado en orden descendente por *trips_amount*.

```sql
SELECT
    CASE
        WHEN cabs.company_name IN ('Flash Cab', 'Taxi Affiliation Services') THEN cabs.company_name
        ELSE 'Other'
    END AS company,
    COUNT(trips.trip_id) AS trips_amount
FROM
    trips
JOIN
    cabs
ON
    trips.cab_id = cabs.cab_id
WHERE
    DATE(trips.start_ts) BETWEEN '2017-11-01' AND '2017-11-07'
GROUP BY
    company
ORDER BY
    trips_amount DESC;

```

Recupera los identificadores de los barrios de O'Hare y Loop de la tabla *neighborhoods*.

```sql
SELECT
    neighborhood_id,
    name
FROM
    neighborhoods
WHERE
    name LIKE '%Hare' OR name LIKE 'Loop';

```

Para cada hora recupera los registros de condiciones meteorológicas de la tabla *weather_records*. Usando el operador CASE, divide todas las horas en dos grupos: Bad si el campo *description* contiene las palabras rain o storm, y Good para los demás. Nombra el campo resultante *weather_conditions*. La tabla final debe incluir dos campos: fecha y hora (*ts*) y *weather_conditions*.

```sql
SELECT
    ts,
    CASE
        WHEN description LIKE '%rain%' OR description LIKE '%storm%' THEN 'Bad'
        ELSE 'Good'
    END AS weather_conditions
FROM
    weather_records;

```

Recupera de la tabla de *trips* todos los viajes que comenzaron en el Loop (*pickup_location_id:* 50) el sábado y terminaron en O'Hare (*dropoff_location_id*: 63). Obtén las condiciones climáticas para cada viaje. Utiliza el método que aplicaste en la tarea anterior. Recupera también la duración de cada viaje. Ignora los viajes para los que no hay datos disponibles sobre las condiciones climáticas.

Las columnas de la tabla deben estar en el siguiente orden:

- *start_ts*
- *weather_conditions*
- *duration_seconds*

Ordena por *trip_id.*

```sql
SELECT
    trips.start_ts AS start_ts,
    T.weather_conditions,
    trips.duration_seconds AS duration_seconds
FROM
    trips
INNER JOIN (
    SELECT
        ts,
        CASE
            WHEN description LIKE '%rain%' OR description LIKE '%storm%' THEN 'Bad'
            ELSE 'Good'
        END AS weather_conditions
    FROM
        weather_records
) T on T.ts = trips.start_ts
WHERE
    pickup_location_id = 50 AND dropoff_location_id = 63 AND EXTRACT (DOW from trips.start_ts) = 6
ORDER BY trip_id;
```