SQL Cheat Sheet

Basic functions

- Window function.
  - RANK(), assign a rank to each row within a partition with gaps, e.g. 1, 1, 3, 4.
    - SELECT department_id, salary, RANK() OVER (PARTITION BY company, department_id ORDER BY salary DESC) AS salary_rank, get Nth highest salary.
  - DENSE_RANK(), without gaps, ranks are assigned in a consecutive manner, e.g. 1, 1, 2, 3.
- Date function.
  - DATEDIFF(), return # of days between two date values.
  - EXTRACT(unit FROM date), return a single part of a date/time, such as year, month, day, hour minute, etc.
    - EXTRACT(month FROM OrderDate);
  - DATE_ADD(date, INTERVAL expr type), adds a specified time interval to a date
    - DATE_ADD(OrderDate, INTERVAL 30 DAY) AS OrderPayDate.
  - MAKEDATE(), create and return a date based on a year and a number of days value.
  - Find the first day of a month.
    - SELECT DATE_ADD(@date, interval -DAY(@date)+1 DAY) AS first_day; The logic is to find the day part of the date, add 1 to it, and subtract it from the date.
  - Find the first day of a week (if the week starts on Sunday)
    - DATE_ADD(mydate, INTERVAL (-DAYOFWEEK(mydate)+1) DAY);
  - Find the first day of a year
    - SELECT DATE(CONCAT(YEAR(CURDATE()),"-01-01"));
- Advanced function.
  - CAST(), converts a value (of any type) into a specified data type.
    - CAST('2022-05-30' AS date).
  - COALESCE(), returns the first non-null value in a list.

Sample questions

SQL leetcode question list: https://leetcode.com/problem-list/e97a9e5m/
SQL Leetcode questions and solutions: https://github.com/mrinal1704/SQL-Leetcode-Challenge

Top travelers

Write a solution to report the distance traveled by each user.
Return the result table ordered by travelled_distance in descending order, if two or more users traveled the same distance, order them by their name in ascending order.
Input:
Users table:

```
+------+-----------+
| id   | name      |
+------+-----------+
| 1    | Alice     |
| 2    | Bob       |
| 3    | Alex      |
| 4    | Donald    |
| 7    | Lee       |
| 13   | Jonathan  |
| 19   | Elvis     |
+------+-----------+
```

Rides table:

```
+------+----------+----------+
| id   | user_id  | distance |
+------+----------+----------+
| 1    | 1        | 120      |
| 2    | 2        | 317      |
| 3    | 3        | 222      |
| 4    | 7        | 100      |
| 5    | 13       | 312      |
| 6    | 19       | 50       |
| 7    | 7        | 120      |
| 8    | 19       | 400      |
| 9    | 7        | 230      |
+------+----------+----------+
```

Output:

```
+----------+--------------------+
| name     | travelled_distance |
+----------+--------------------+
| Elvis    | 450                |
| Lee      | 450                |
| Bob      | 317                |
| Jonathan | 312                |
| Alex     | 222                |
| Alice    | 120                |
| Donald   | 0                  |
+----------+--------------------+
```

Solution:

```sql
SELECT
    u.name,
    SUM(COALESCE(r.distance, 0)) AS travelled_distance
FROM users u
LEFT JOIN rides r ON u.id = r.user_id
GROUP BY u.name
ORDER BY travelled_distance DESC, u.name;
```

Second Highest Salary

Write a solution to find the second highest salary from the Employee table. If there is no second highest salary, return null (return None in Pandas).

The result format is in the following example.

Example 1:

Input:
Employee table:

```
+----+--------+
| id | salary |
+----+--------+
| 1  | 100    |
| 2  | 200    |
| 3  | 300    |
+----+--------+
```

Output:

```
+---------------------+
| SecondHighestSalary |
+---------------------+
| 200                 |
+---------------------+
```

Example 2:

Input:
Employee table:

```
+----+--------+
| id | salary |
+----+--------+
| 1  | 100    |
+----+--------+
```

Output:

```
+--------------------+
| SecondHighestSalary |
+--------------------+
| null               |
+--------------------+
```

Solution:

```sql
SELECT
    min(a.Salary) AS SecondHighestSalary
FROM
    (SELECT
        Id,
        Salary,
        RANK() OVER (ORDER BY Salary DESC) AS rank
    FROM
        Employee) a
WHERE a.rank = 2;
```