

Employees With Missing Information

Question:

<https://leetcode.com/problems/employees-with-missing-information/>

EMPLOYEES	
Column	Type
emp_id	int
name	varchar

employee_id is the primary key for this table.

Each row of this table indicates the name of the employee whose ID is employee_id.

SALARIES	
Column	Type
emp_id	int
salary	int

employee_id is the primary key for this table.

Each row of this table indicates the salary of the employee whose ID is employee_id.

Write an SQL query to report the IDs of all the employees with missing information. The information of an employee is missing if:

The employee's name is missing, or

The employee's salary is missing.

Return the result table ordered by employee_id in ascending order.

Query 1

Tried using a **WHERE** clause with a condition to check for employees that are in **Employees DB** but not in **Salaries DB**.

```
SELECT  
  
    employee_id  
  
FROM Employees  
  
WHERE Employees.employee_id NOT IN (SELECT employee_id FROM Salaries)
```

The logic was partially correct. What we also need other than this condition is a back checking condition.

Employees that have an id in **Salaries DB** but not in **Employees DB**.

Query 2(Accepted)

Included the 2nd condition and connected the 2 conditions with a **UNION** clause.

```
SELECT  
  
    employee_id  
  
FROM Employees  
  
WHERE Employees.employee_id NOT IN (SELECT employee_id FROM Salaries)  
  
UNION  
  
SELECT  
  
    employee_id  
  
FROM Salaries  
  
WHERE Salaries.employee_id NOT IN (SELECT employee_id FROM Employees)  
  
ORDER BY 1 ASC
```

Query 3(Accepted)

Found this solution through the discussion forum

```
SELECT

    sub.employee_id

FROM ( SELECT

        e.employee_id,name,salary

        FROM Employees AS e

        LEFT JOIN Salaries AS s

        ON e.employee_id = s.employee_id

        UNION

        SELECT

            s.employee_id,name,salary

            FROM Employees AS e

            RIGHT JOIN Salaries AS s

            ON e.employee_id = s.employee_id ) AS sub

WHERE (sub.name IS NULL) OR (sub.salary IS NULL)

ORDER BY sub.employee_id
```

Although it is doing the same thing as **QUERY 2**, but here we are traversing the database 3 times whereas in **QUERY 2**, we traversed it twice.

PS: Query 2 was the fastest and the best accepted solution!!!