Table: `Project`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| project_id  | int     |
| employee_id | int     |
+-------------+---------+
(project_id, employee_id) is the primary key of this table.
employee_id is a foreign key to Employee table.
Each row of this table indicates that the employee with employee_id is working on the proje
```

Table: `Employee`

```
+------------------+---------+
| Column Name      | Type    |
+------------------+---------+
| employee_id      | int     |
| name             | varchar |
| experience_years | int     |
+------------------+---------+
employee_id is the primary key of this table.
Each row of this table contains information about one employee.
```

Write an SQL query that reports the **most experienced** employees in each project. In case of a tie, report all employees with the maximum number of experience years.

Return the result table in **any order**.

The query result format is in the following example.

Example 1:**

```
Input:
Project table:
+-------------+-------------+
| project_id  | employee_id |
+-------------+-------------+
| 1           | 1           |
| 1           | 2           |
| 1           | 3           |
| 2           | 1           |
```

1

```
| 2          | 4           |
+------------+-------------+
```

Employee table:

```
+------------+--------+------------------+
| employee_id | name   | experience_years |
+------------+--------+------------------+
| 1          | Khaled | 3                |
| 2          | Ali    | 2                |
| 3          | John   | 3                |
| 4          | Doe    | 2                |
+------------+--------+------------------+
```

Output:

```
+------------+--------------+
| project_id | employee_id  |
+------------+--------------+
| 1          | 1            |
| 1          | 3            |
| 2          | 1            |
+------------+--------------+
```

Explanation: Both employees with id 1 and 3 have the most experience among the employees of