Table: `Person`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| personId    | int     |
| lastName    | varchar |
| firstName   | varchar |
+-------------+---------+
```

personId is the primary key column for this table.

This table contains information about the ID of some persons and their first and last names.
```

Table: `Address`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| addressId   | int     |
| personId    | int     |
| city        | varchar |
| state       | varchar |
+-------------+---------+
```

addressId is the primary key column for this table.

Each row of this table contains information about the city and state of one person with ID = PersonId.

```

Write an SQL query to report the first name, last name, city, and state of each person in the `Person` table. If the address of a `personId` is not present in the `Address` table, report `null` instead.

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

The query result format is in the following example.

**Example 1:**\*\*

```

Input:

Person table:

+----------+----------+-----------+
| personId | lastName | firstName |
+----------+----------+-----------+
| 1 | Wang | Allen |
| 2 | Alice | Bob |
+----------+----------+-----------+

Address table:

+-----------+----------+---------------+------------+
| addressId | personId | city | state |
+-----------+----------+---------------+------------+

| 1 | 2 | New York City | New York |
| 2 | 3 | Leetcode | California |

+-----------+----------+---------------+------------+

Output:

+-----------+----------+---------------+----------+
| firstName | lastName | city | state |
+-----------+----------+---------------+----------+
| Allen | Wang | Null | Null |
| Bob | Alice | New York City | New York |
+-----------+----------+---------------+----------+

Explanation:

There is no address in the address table for the personId = 1 so we return null in their city and state.

addressId = 1 contains information about the address of personId = 2.

```

**True**

Table: `Employee`

```

+-------------+---------+
| Column Name | Type |
+-------------+---------+
id	int
name	varchar
salary	int
managerId	int
+-------------+---------+

id is the primary key column for this table.

Each row of this table indicates the ID of an employee, their name, salary, and the ID of their manager.

```
```

Write an SQL query to find the employees who earn more than their managers.

Return the result table in any order.

The query result format is in the following example.

Example 1:**

```
```

Input:

Employee table:

```
+----+-------+--------+-----------+
| id | name  | salary | managerId |
+----+-------+--------+-----------+
| 1  | Joe   | 70000  | 3         |
| 2  | Henry | 80000  | 4         |
| 3  | Sam   | 60000  | Null      |
| 4  | Max   | 90000  | Null      |
+----+-------+--------+-----------+
```

Output:

```
+----------+
| Employee |
```

```
+----------+
| Joe      |
+----------+
```

Explanation: Joe is the only employee who earns more than his manager.

```