Solution

Approach 1: Using `WHERE` clause [Accepted]

Algorithm

As this table has the employee's manager information, we probably need to select information from it twice.

```
SELECT *
FROM Employee AS a, Employee AS b
;
```

> Note: The keyword 'AS' is optional.

Id	Name	Salary	ManagerId	Id	Name	Salary	ManagerId
1	Joe	70000	3	1	Joe	70000	3
2	Henry	80000	4	1	Joe	70000	3
3	Sam	60000		1	Joe	70000	3
4	Max	90000		1	Joe	70000	3
1	Joe	70000	3	2	Henry	80000	4
2	Henry	80000	4	2	Henry	80000	4
3	Sam	60000		2	Henry	80000	4
4	Max	90000		2	Henry	80000	4
1	Joe	70000	3	3	Sam	60000	
2	Henry	80000	4	3	Sam	60000	
3	Sam	60000		3	Sam	60000	
4	Max	90000		3	Sam	60000	
1	Joe	70000	3	4	Max	90000	
2	Henry	80000	4	4	Max	90000	
3	Sam	60000		4	Max	90000	
4	Max	90000		4	Max	90000	

> The first 3 columns are from a and the last 3 ones are from b.

Select from two tables will get the Cartesian product of these two tables. In this case, the output will be 4*4 = 16 records. However, what we interest is the employee's salary higher than his/her manager. So we should add two conditions in a `WHERE` clause like below.

```
SELECT
    *
FROM
    Employee AS a,
    Employee AS b
WHERE
    a.ManagerId = b.Id
        AND a.Salary > b.Salary
;
```

Id	Name	Salary	ManagerId	Id	Name	Salary	ManagerId
1	Joe	70000	3	3	Sam	60000	

As we only need to output the employee's name, so we modify the above code a little to get a solution.

MySQL

```
SELECT
    a.Name AS 'Employee'
FROM
    Employee AS a,
    Employee AS b
WHERE
    a.ManagerId = b.Id
        AND a.Salary > b.Salary
;
```

Approach 2: Using `JOIN` clause [Accepted]

Algorithm

Actually, `JOIN` is a more common and efficient way to link tables together, and we can use `ON` to specify some conditions.

```
SELECT
     a.NAME AS Employee
FROM Employee AS a JOIN Employee AS b
     ON a.ManagerId = b.Id
     AND a.Salary > b.Salary
;
```