Well Paid Employees

Companies often perform salary analyses to ensure fair compensation practices. One useful analysis is to check if there are any employees earning more than their direct managers.

As a HR Analyst, you're asked to identify all employees who earn more than their direct managers. The result should include the employee's ID and name.

`employee` Schema:

column_name	type	description
employee_id	integer	The unique ID of the employee.
name	string	The name of the employee.
salary	integer	The salary of the employee.
department_id	integer	The department ID of the employee.
manager_id	integer	The manager ID of the employee.

`employee` Example Input:

employee_id	name	salary	department_id	manager_id
1	Emma Thompson	3800	1	6
2	Daniel Rodriguez	2230	1	7
3	Olivia Smith	7000	1	8
4	Noah Johnson	6800	2	9
5	Sophia Martinez	1750	1	11
6	Liam Brown	13000	3	NULL
7	Ava Garcia	12500	3	NULL
8	William Davis	6800	2	NULL

```
SELECT e.employee_id, e.name as employee_name FROM employee e
inner join employee m
on e.manager_id = m.employee_id
where e.salary>m.salary
```

Average Post Hiatus (Part 1)

Given a table of Facebook posts, for each user who posted at least twice in 2021, write a query to find the number of days between each user's first post of the year and last post of the year in the year 2021. Output the user and number of the days between each user's first and last post.

`posts` Table:

Column Name	Type
user_id	integer
post_id	integer
post_content	text
post_date	timestamp

`posts` Example Input:

user_id	post_id	post_content	post_date
151652	599415	Need a hug	07/10/2021 12:00:00
661093	624356	Bed. Class 8-12. Work 12-3. Gym 3-5 or 6. Then class 6-10. Another day that's gonna fly by. I miss my girlfriend	07/29/2021 13:00:00
004239	784254	Happy 4th of July!	07/04/2021 11:00:00
661093	442560	Just going to cry myself to sleep after watching Marley and Me.	07/08/2021 14:00:00
151652	111766	I'm so done with covid - need travelling ASAP!	07/12/2021 19:00:00

```
WITH count_user AS (
   SELECT user_id, COUNT(post_id) AS post_count
   FROM posts
   WHERE post_date < '2022-01-01'
   GROUP BY user_id
   HAVING COUNT(post_id) > 1
)

SELECT p.user_id,
      EXTRACT(DAY FROM (MAX(p.post_date) - MIN(p.post_date))) AS days_between
FROM posts p
INNER JOIN count_user cu ON p.user_id = cu.user_id
GROUP BY p.user_id
ORDER BY p.user_id;
```