Easy

Table: Employees

```
+-------------+----------+
| Column Name | Type     |
+-------------+----------+
| employee_id | int      |
| name        | varchar  |
| reports_to  | int      |
| age         | int      |
+-------------+----------+
```

employee_id is the column with unique values for this table.
This table contains information about the employees and the id of the manager they report to. Some employees do not report to anyone (reports_to is null).

For this problem, we will consider a **manager** an employee who has at least 1 other employee reporting to them.

Write a solution to report the ids and the names of all **managers**, the number of employees who report **directly** to them, and the average age of the reports rounded to the nearest integer.

Return the result table ordered by employee_id.

The result format is in the following example.

**Example 1:**

**Input:**
Employees table:

```
+-------------+---------+------------+-----+
| employee_id | name    | reports_to | age |
+-------------+---------+------------+-----+
| 9           | Hercy   | null       | 43  |
| 6           | Alice   | 9          | 41  |
| 4           | Bob     | 9          | 36  |
| 2           | Winston | null       | 37  |
+-------------+---------+------------+-----+
```

**Output:**

```
+-------------+-------+---------------+-------------+
| employee_id | name  | reports_count | average_age |
+-------------+-------+---------------+-------------+
| 9           | Hercy | 2             | 39          |
+-------------+-------+---------------+-------------+
```

**Explanation:** Hercy has 2 people report directly to him, Alice and Bob. Their average age is (41+36)/2 = 38.5, which is 39 after rounding it to the nearest integer.

**Example 2:**

**Input:**
Employees table:

```
+-------------+---------+------------+-----+
| employee_id | name    | reports_to | age |
|-------------|---------|------------|-----|
| 1           | Michael | null       | 45  |
| 2           | Alice   | 1          | 38  |
| 3           | Bob     | 1          | 42  |
| 4           | Charlie | 2          | 34  |
| 5           | David   | 2          | 40  |
| 6           | Eve     | 3          | 37  |
| 7           | Frank   | null       | 50  |
| 8           | Grace   | null       | 48  |
+-------------+---------+------------+-----+
```

**Output:**

```
+-------------+---------+---------------+-------------+
| employee_id | name    | reports_count | average_age |
| ----------- | ------- | ------------- | ----------- |
| 1           | Michael | 2             | 40          |
| 2           | Alice   | 2             | 37          |
| 3           | Bob     | 1             | 37          |
+-------------+---------+---------------+-------------+
```

```
# Write your MySQL query statement below
SELECT e.reports_to as employee_id,
       m.name,
       COUNT(*) AS reports_count,
       ROUND(AVG(e.age)) AS average_age
FROM Employees e
JOIN Employees m
   ON e.reports_to=m.employee_id
WHERE e.reports_to IS NOT NULL
GROUP BY e.reports_to
ORDER BY employee_id;

-- STEP 1:
-- SELECT e.reports_to as employee_id
--          m.name,
-- FROM Employees e
-- JOIN Employees m
--     ON e.reports_to=m.employee_id
-- WHERE e.reports_to IS NOT NULL;

-- STEP 2:
-- SELECT e.reports_to as employee_id
--        m.name,
--        COUNT(*) AS reports_count
-- FROM Employees e
-- JOIN Employees m
--     ON e.reports_to=m.employee_id
-- WHERE e.reports_to IS NOT NULL
-- GROUP BY e.reports_to;

-- STEP 3:
-- SELECT e.reports_to as employee_id,
--        m.name,
--        COUNT(*) AS reports_count,
--        ROUND(AVG(e.age)) AS average_age
-- FROM Employees e
-- JOIN Employees m
--     ON e.reports_to=m.employee_id
-- WHERE e.reports_to IS NOT NULL
-- GROUP BY e.reports_to
-- ORDER BY employee_id;
```