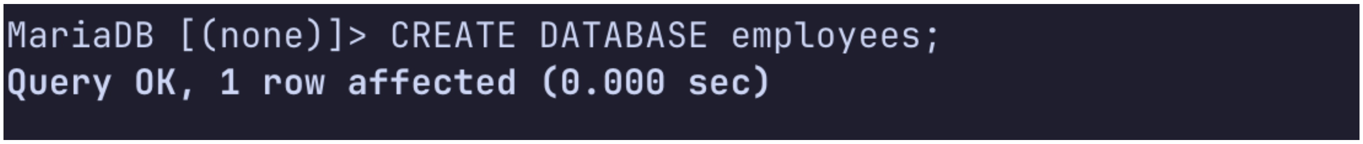
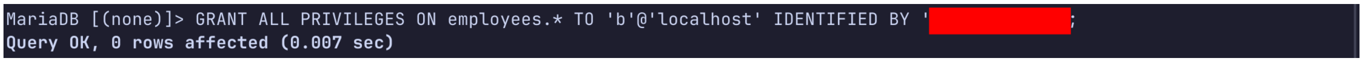
For this week's exercises, we are tasked to load a file into a database called 'employees'.

Let's log in as root and create a new database:



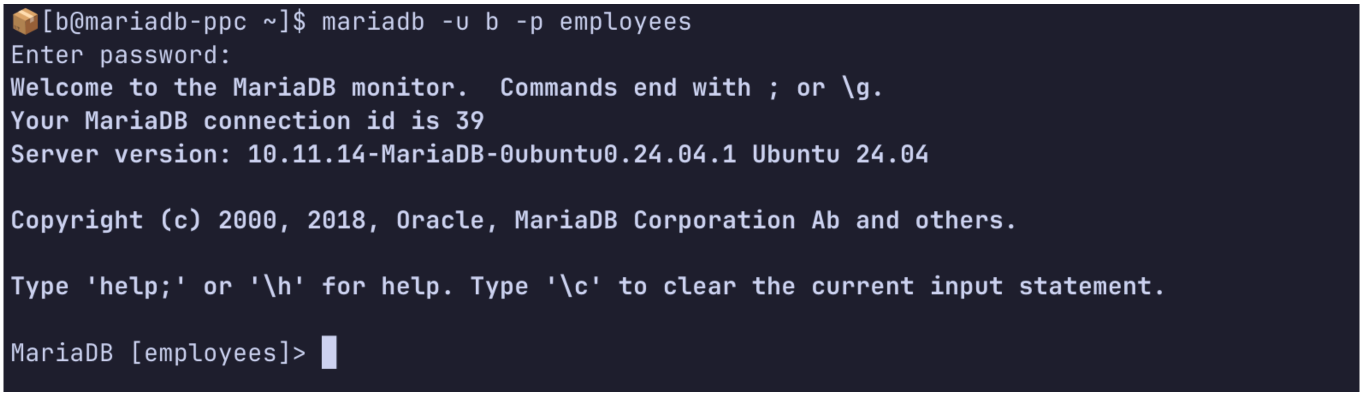
```
MariaDB [(none)]> CREATE DATABASE employees;
Query OK, 1 row affected (0.000 sec)
```

Next, let's give my user permissions.



```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON employees.* TO 'b'@'localhost' IDENTIFIED BY '[redacted];
Query OK, 0 rows affected (0.007 sec)
```

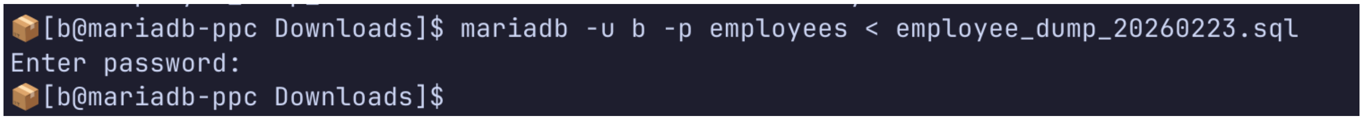
Now, I'll login with my user b.



```
📦[b@mariadb-ppc ~]$ mariadb -u b -p employees
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.11.14-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [employees]> 
```

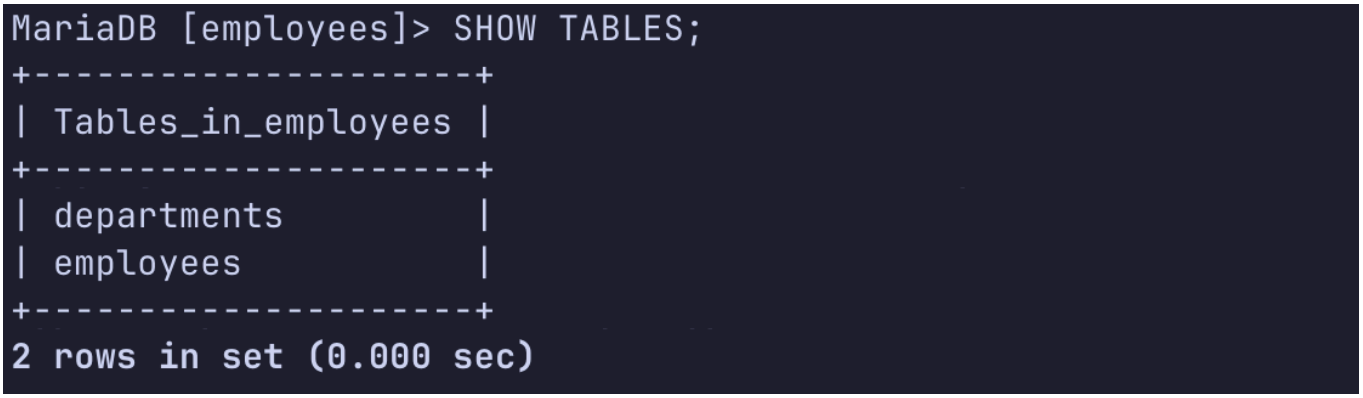
Looks good!

Next step is to read in the dump file:



```
📦[b@mariadb-ppc Downloads]$ mariadb -u b -p employees < employee_dump_20260223.sql
Enter password:
📦[b@mariadb-ppc Downloads]$
```

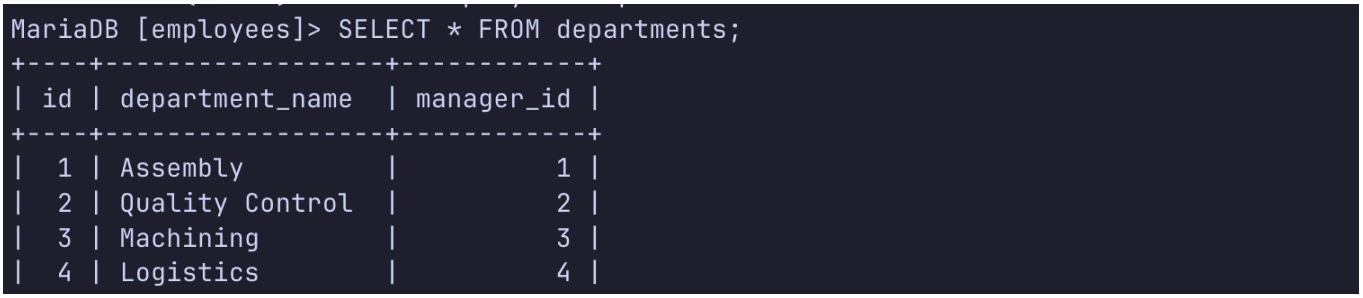
And then test that the file import succeeded:



```
MariaDB [employees]> SHOW TABLES;
+---------------------+
| Tables_in_employees |
+---------------------+
| departments         |
| employees           |
+---------------------+
2 rows in set (0.000 sec)
```

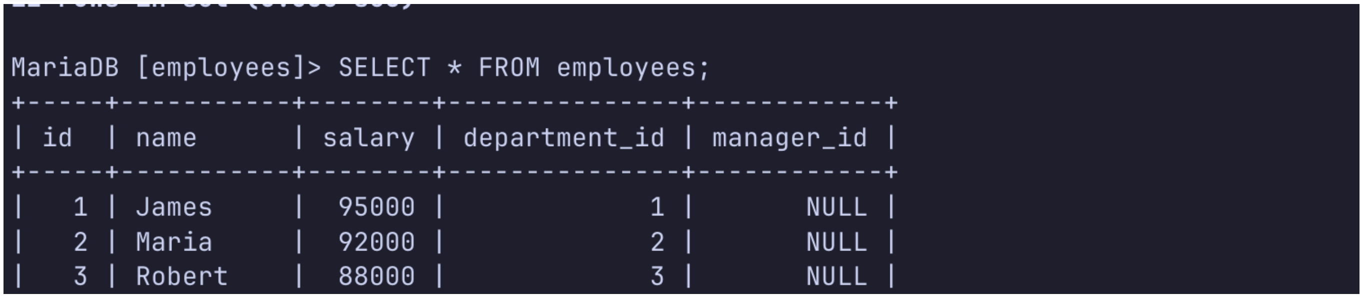
And let's check out both tables to see what we're working with:

departments:



```
MariaDB [employees]> SELECT * FROM departments;
+----+-----------------+------------+
| id | department_name | manager_id |
+----+-----------------+------------+
|  1 | Assembly        |          1 |
|  2 | Quality Control |          2 |
|  3 | Machining       |          3 |
|  4 | Logistics       |          4 |
```

employees:



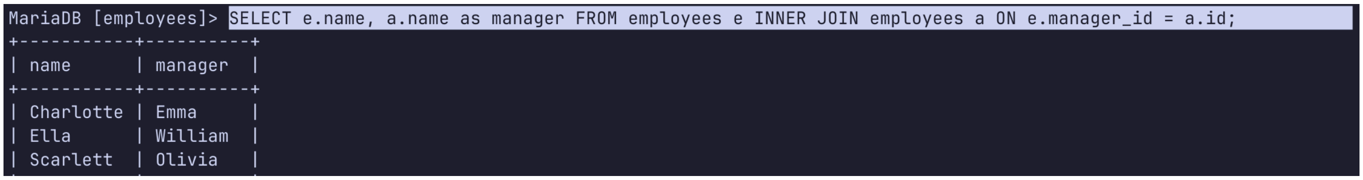
```
MariaDB [employees]> SELECT * FROM employees;
+-----+-----------+--------+---------------+------------+
| id  | name      | salary | department_id | manager_id |
+-----+-----------+--------+---------------+------------+
|   1 | James     |  95000 |             1 |       NULL |
|   2 | Maria     |  92000 |             2 |       NULL |
|   3 | Robert    |  88000 |             3 |       NULL |
```

Great! Now we can get started on our exercises.

1. List employees and their managers

A simple inner join can solve this:

```
SELECT e.name, a.name as manager FROM employees e INNER JOIN employees a ON
e.manager_id = a.id;
```

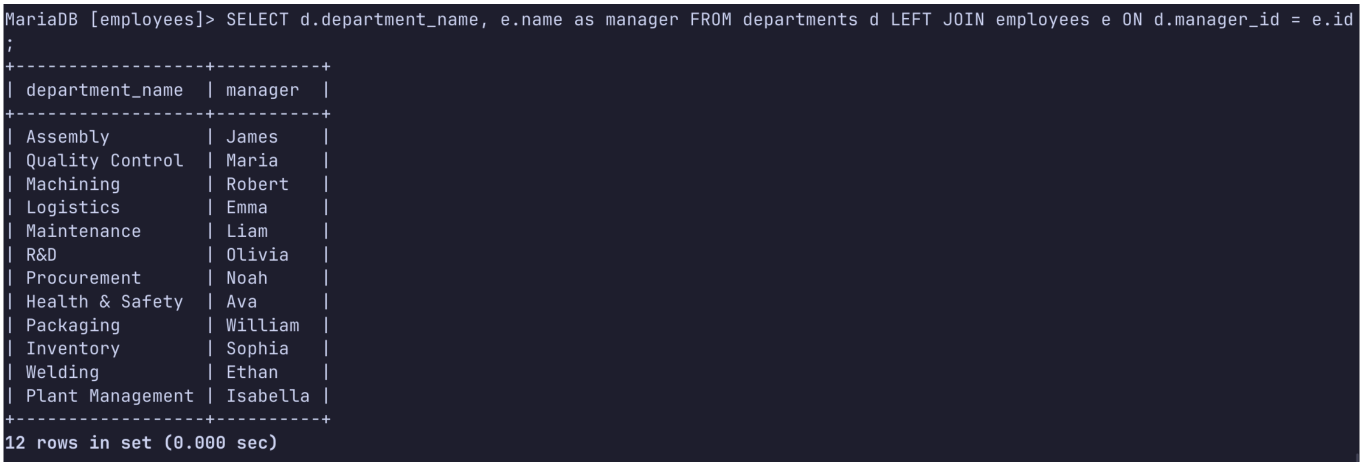


```
MariaDB [employees]> SELECT e.name, a.name as manager FROM employees e INNER JOIN employees a ON e.manager_id = a.id;
+-----------+----------+
| name      | manager  |
+-----------+----------+
| Charlotte | Emma     |
| Ella      | William  |
| Scarlett  | Olivia   |
```

2. List departments and their managers

Simple left join:

```
SELECT d.department_name, e.name as manager FROM departments d LEFT JOIN
employees e ON d.manager_id = e.id;
```

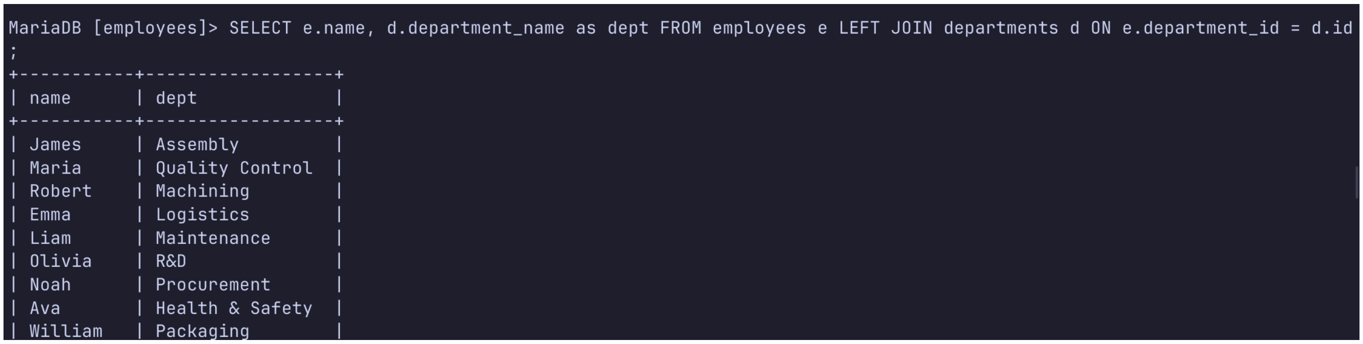


```
MariaDB [employees]> SELECT d.department_name, e.name as manager FROM departments d LEFT JOIN employees e ON d.manager_id = e.id
;
+------------------+----------+
| department_name  | manager  |
+------------------+----------+
| Assembly         | James    |
| Quality Control  | Maria    |
| Machining        | Robert   |
| Logistics        | Emma     |
| Maintenance      | Liam     |
| R&D              | Olivia   |
| Procurement      | Noah     |
| Health & Safety  | Ava      |
| Packaging        | William  |
| Inventory        | Sophia   |
| Welding          | Ethan    |
| Plant Management | Isabella |
+------------------+----------+
12 rows in set (0.000 sec)
```

3. List department each person works in

Left join again:

```
SELECT e.name, d.department_name as dept FROM employees e LEFT JOIN
departments d ON e.department_id = d.id;
```

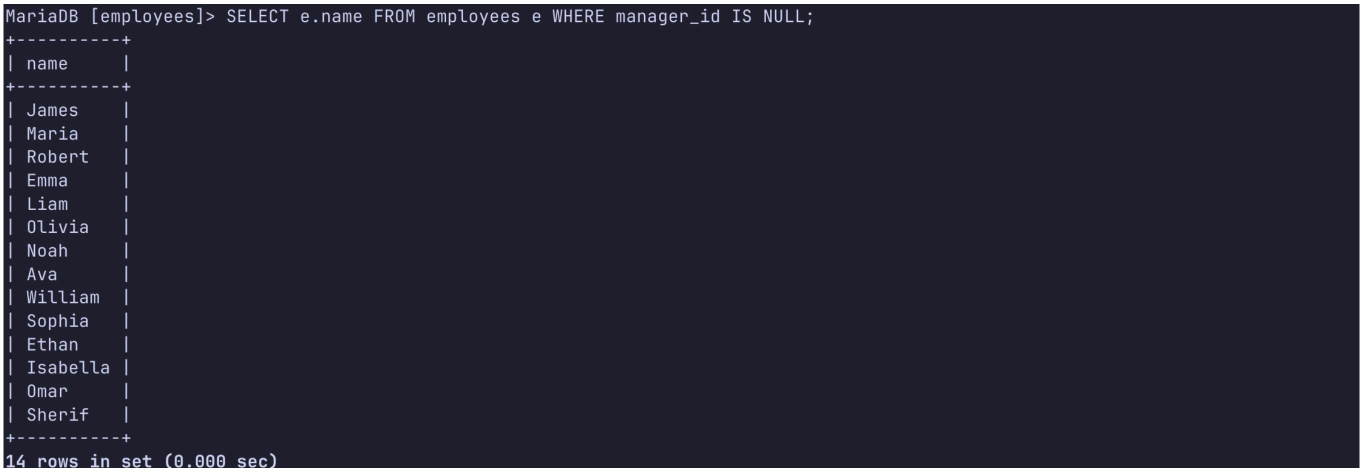


```
MariaDB [employees]> SELECT e.name, d.department_name as dept FROM employees e LEFT JOIN departments d ON e.department_id = d.id
;
+-----------+------------------+
| name      | dept             |
+-----------+------------------+
| James     | Assembly         |
| Maria     | Quality Control  |
| Robert    | Machining        |
| Emma      | Logistics        |
| Liam      | Maintenance      |
| Olivia    | R&D              |
| Noah      | Procurement      |
| Ava       | Health & Safety  |
| William   | Packaging        |
```

4. Find employees who have no manager

Since we know an employee with no manager has `manager_id` NULL, we can use `IS NULL`:

```
SELECT e.name FROM employees e WHERE manager_id IS NULL;
```

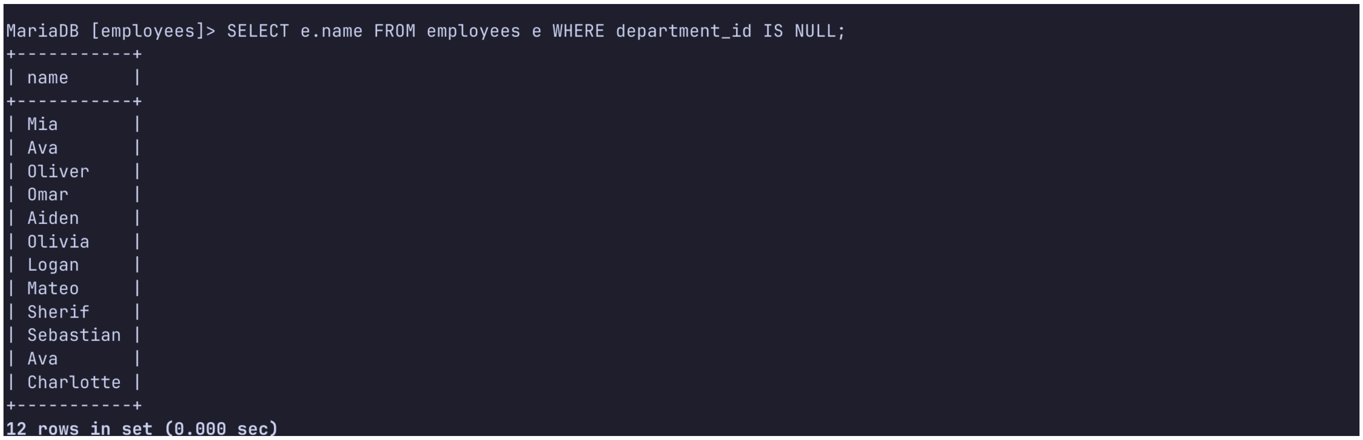


```
MariaDB [employees]> SELECT e.name FROM employees e WHERE manager_id IS NULL;
+----------+
| name     |
+----------+
| James    |
| Maria    |
| Robert   |
| Emma     |
| Liam     |
| Olivia   |
| Noah     |
| Ava      |
| William  |
| Sophia   |
| Ethan    |
| Isabella |
| Omar     |
| Sherif   |
+----------+
14 rows in set (0.000 sec)
```

5. Find employees who have no departments

Similar idea as #4, using `IS NULL`.

```
SELECT e.name FROM employees e WHERE department_id IS NULL;
```

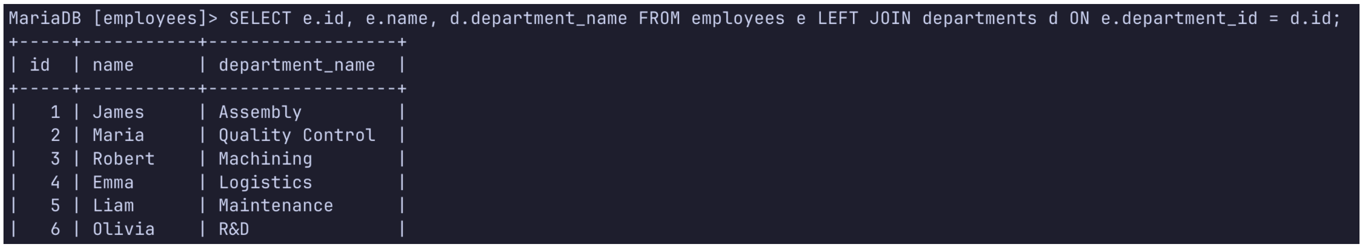


```
MariaDB [employees]> SELECT e.name FROM employees e WHERE department_id IS NULL;
+-----------+
| name      |
+-----------+
| Mia       |
| Ava       |
| Oliver    |
| Omar      |
| Aiden     |
| Olivia    |
| Logan     |
| Mateo     |
| Sherif    |
| Sebastian |
| Ava       |
| Charlotte |
+-----------+
12 rows in set (0.000 sec)
```

6. Show a count of employees in each department, including those who have no departments.

First, I'll run a query to show all employees and their departments, even if they have no department.

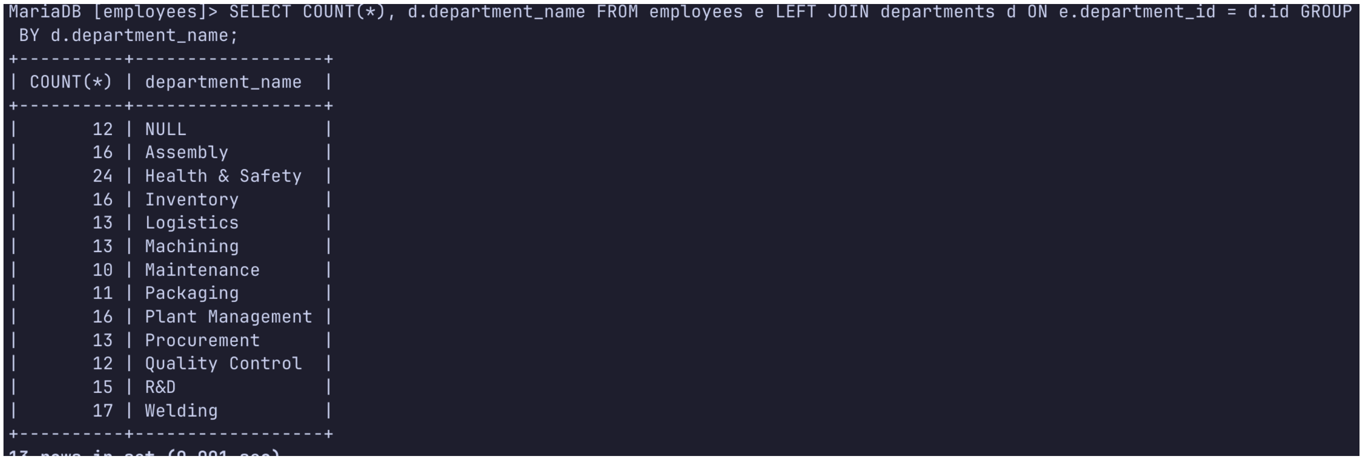
```
SELECT e.id, e.name, d.department_name FROM employees e LEFT JOIN
departments d ON e.department_id = d.id;
```



```
MariaDB [employees]> SELECT e.id, e.name, d.department_name FROM employees e LEFT JOIN departments d ON e.department_id = d.id;
+-----+-----------+-----------------+
| id  | name      | department_name |
+-----+-----------+-----------------+
|   1 | James     | Assembly        |
|   2 | Maria     | Quality Control |
|   3 | Robert    | Machining       |
|   4 | Emma      | Logistics       |
|   5 | Liam      | Maintenance     |
|   6 | Olivia    | R&D             |
```

Now, let's alter it to give us the count per department and group by department name.

```
SELECT COUNT(*), d.department_name FROM employees e LEFT JOIN departments d
ON e.department_id = d.id GROUP BY d.department_name;
```



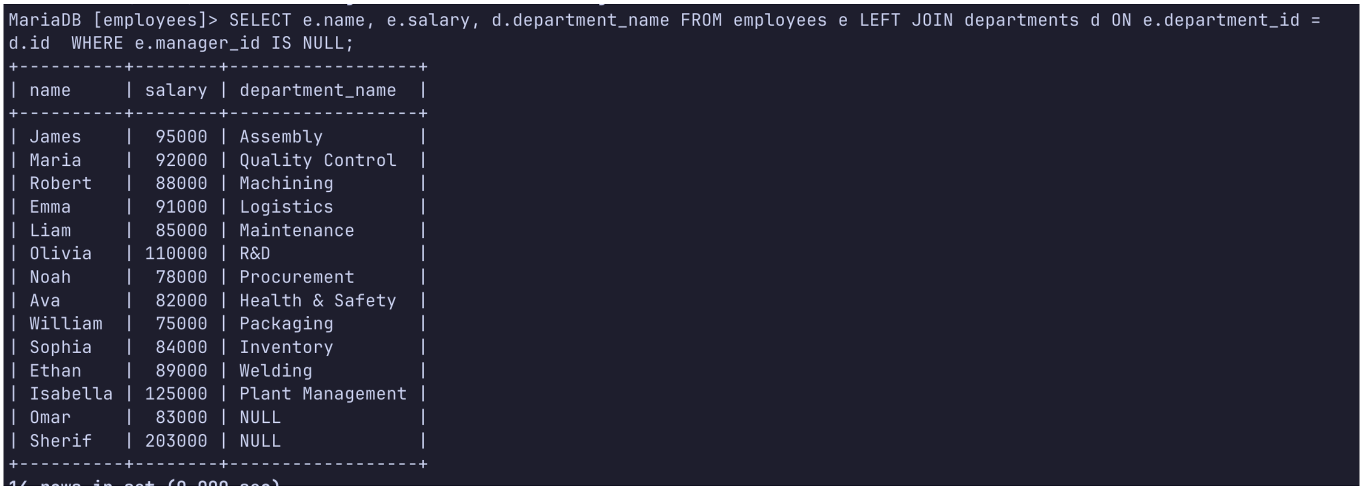
```
MariaDB [employees]> SELECT COUNT(*), d.department_name FROM employees e LEFT JOIN departments d ON e.department_id = d.id GROUP
 BY d.department_name;
+----------+------------------+
| COUNT(*) | department_name  |
+----------+------------------+
|       12 | NULL             |
|       16 | Assembly         |
|       24 | Health & Safety  |
|       16 | Inventory        |
|       13 | Logistics        |
|       13 | Machining        |
|       10 | Maintenance      |
|       11 | Packaging        |
|       16 | Plant Management |
|       13 | Procurement      |
|       12 | Quality Control  |
|       15 | R&D              |
|       17 | Welding          |
+----------+------------------+
13 rows in set (0.001 sec)
```

Looks great!

7. List name, salary, and department name of managers (those who have a department but no manager)

For this question, we are looking for people from `employees` who have no manager_id, meaning they are managers themselves. We'll combine `IS NULL` and a left join here:

```
SELECT e.name, e.salary, d.department_name FROM employees e LEFT JOIN
departments d ON e.department_id =
d.id  WHERE e.manager_id IS NULL;
```



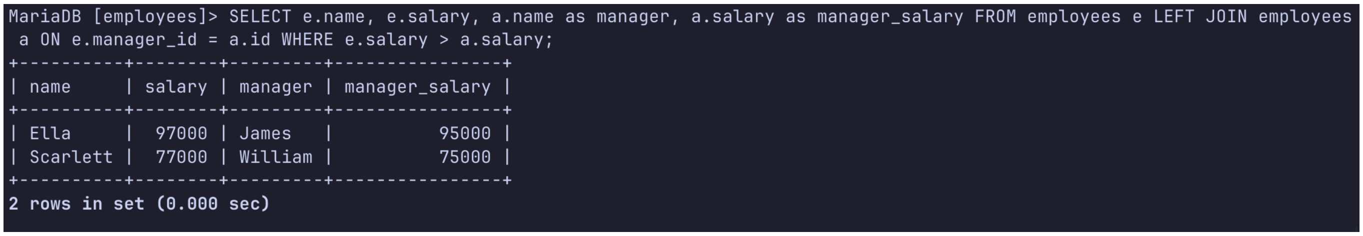
```
MariaDB [employees]> SELECT e.name, e.salary, d.department_name FROM employees e LEFT JOIN departments d ON e.department_id =
d.id  WHERE e.manager_id IS NULL;
+----------+--------+------------------+
| name     | salary | department_name  |
+----------+--------+------------------+
| James    |  95000 | Assembly         |
| Maria    |  92000 | Quality Control  |
| Robert   |  88000 | Machining        |
| Emma     |  91000 | Logistics        |
| Liam     |  85000 | Maintenance      |
| Olivia   | 110000 | R&D              |
| Noah     |  78000 | Procurement      |
| Ava      |  82000 | Health & Safety  |
| William  |  75000 | Packaging        |
| Sophia   |  84000 | Inventory        |
| Ethan    |  89000 | Welding          |
| Isabella | 125000 | Plant Management |
| Omar     |  83000 | NULL             |
| Sherif   | 203000 | NULL             |
+----------+--------+------------------+
14 rows in set (0.000 sec)
```

Looks good! Interesting note: there are people with no department or manager.

8. List employees whose salaries are higher than their manager's salaries.

For this, we need to self join the employees table to compare salaries of employees and their managers:

```
SELECT e.name, e.salary, a.name as manager, a.salary as manager_salary FROM
employees e LEFT JOIN employees a ON e.manager_id = a.id WHERE e.salary >
a.salary;
```



```
MariaDB [employees]> SELECT e.name, e.salary, a.name as manager, a.salary as manager_salary FROM employees e LEFT JOIN employees
 a ON e.manager_id = a.id WHERE e.salary > a.salary;
+----------+--------+---------+----------------+
| name     | salary | manager | manager_salary |
+----------+--------+---------+----------------+
| Ella     |  97000 | James   |          95000 |
| Scarlett |  77000 | William |          75000 |
+----------+--------+---------+----------------+
2 rows in set (0.000 sec)
```

It looks like there aren't many employees with salaries greater than their manager's salary.