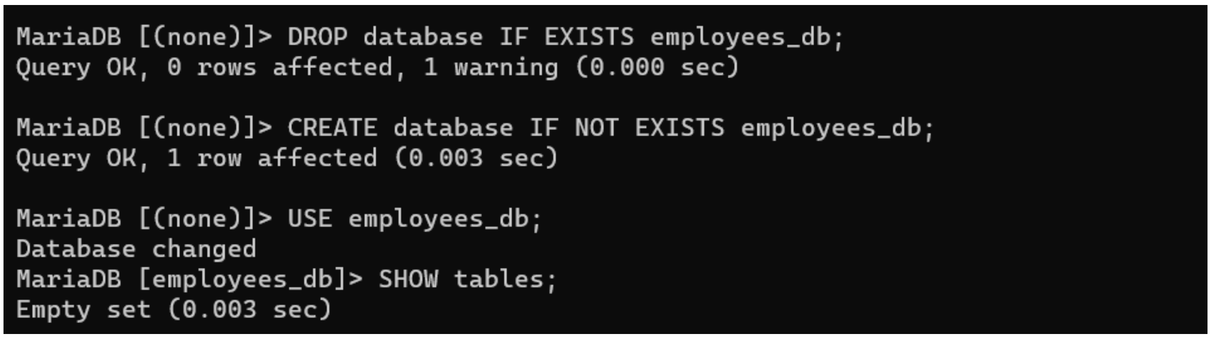
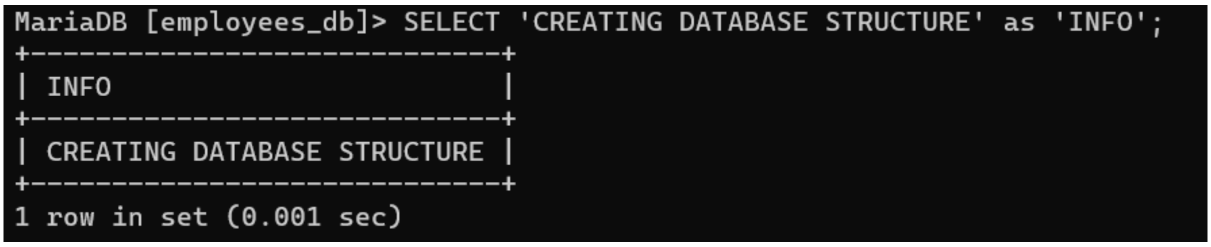
MySQL Assessment 11/02/24

Create database:



```
MariaDB [(none)]> DROP database IF EXISTS employees_db;
Query OK, 0 rows affected, 1 warning (0.000 sec)

MariaDB [(none)]> CREATE database IF NOT EXISTS employees_db;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]> USE employees_db;
Database changed
MariaDB [employees_db]> SHOW tables;
Empty set (0.003 sec)
```



```
MariaDB [employees_db]> SELECT 'CREATING DATABASE STRUCTURE' as 'INFO';
+-----------------------------+
| INFO                        |
+-----------------------------+
| CREATING DATABASE STRUCTURE |
+-----------------------------+
1 row in set (0.001 sec)
```

Building the tables:

Normalisation of the database structure:

1. Employees table:
    a. Table consistent but number of characters might be limited as names can be longer.
    b. We add an alias ‘emp_pk’ to primary key to establish relations.

2. Departments table:
    a. Table consistent, using UNIQUE KEY or just UNIQUE to define ‘dept_name’ can be considered as a difference of style and achieves the same result.
    b. We change the number of characters of the dept_no to 8 to allow for updating them to ‘old.’. This will allow us not to delete any department, as explained in the dept_manager table normalization.
    c. We add an alias ‘dept_pk’ to primary key to establish relations.

3. Dept_manager table:
    a. Here we have an issue as we have a composite PRIMARY KEY (emp_no, dept_no). An employee can manage twice the same department in their career, so I will remove the primary key as it is not used in the structure.

    b. Since the dept_no is set as NOT NULL, if we have a deletion triggered by the deletion of a department, we run into another violation as the corresponding rows will be null. To avoid that and data loss, we will replace the ON DELETE CASCADE to ON UPDATE CASCADE. A department cannot be deleted from the departments table without causing a problem in the child table. Therefore, we change the dept_no to a different code to mark that it is no longer active, by example with prefix 'old.'. To allow for this we first change the number of characters of the dept_no in all relevant tables to 8.

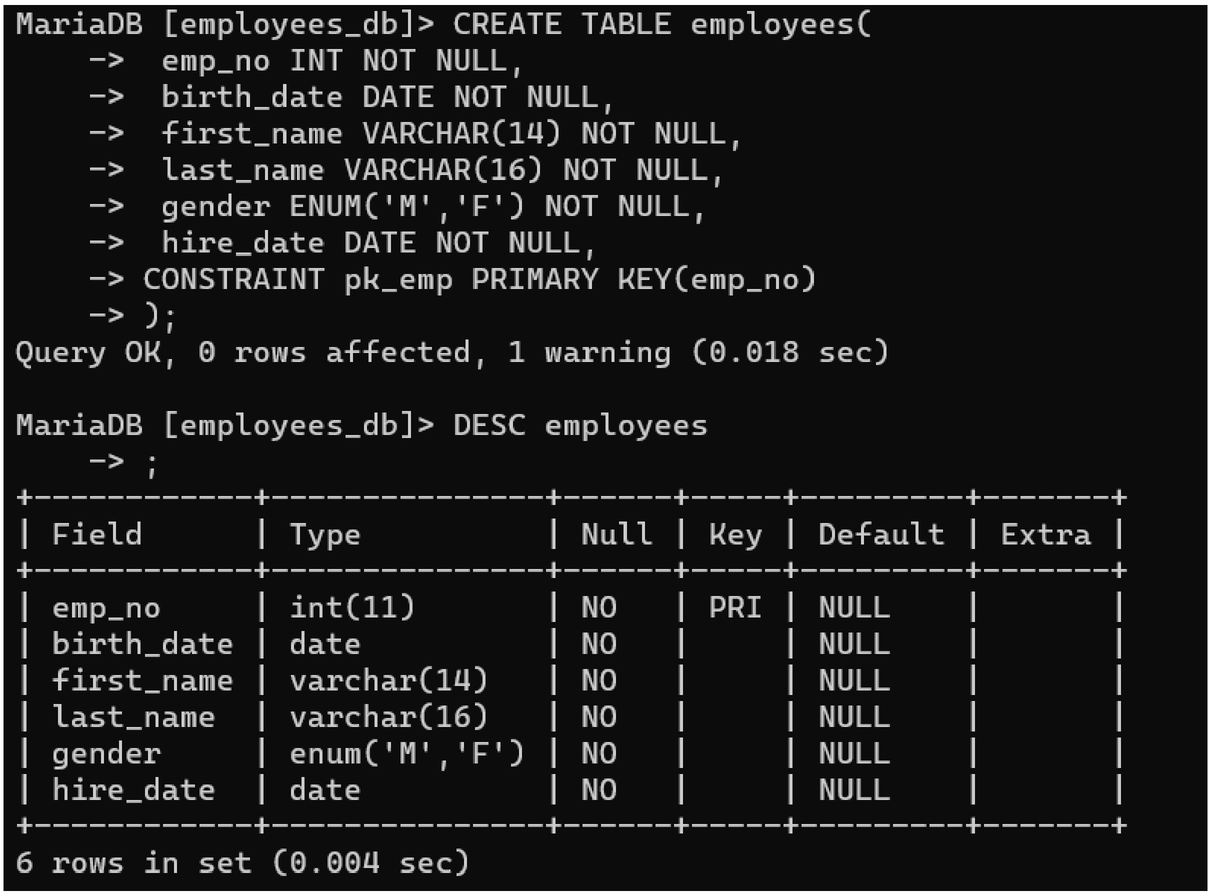
4. Dept_emp table:
    a. We perform the same changes as in the dept_manager table to remove PRIMARY KEY (an employee can work twice in the same department at different times in their careers. And we ON UPDATE CASCADE the dept_no to keep historical data and avoid violations in the NOT NULL column.

5. Titles table:
    a. We can see this table cannot have a title if not associated with an employee, and it means titles can have typos and inconsistencies which might impede our analysis. I think it makes more sense to have a table with all the titles, and a title_id. We can then add the title_id as a column to a new emp_title table which refers to it.
    b. The new table emp_title will have emp_no, title_id, from_date, to_date, and the titles table will only have title_id and title_name columns.

6. Salaries table:
    a. Here we perform the same normalization: removal of the composite primary key

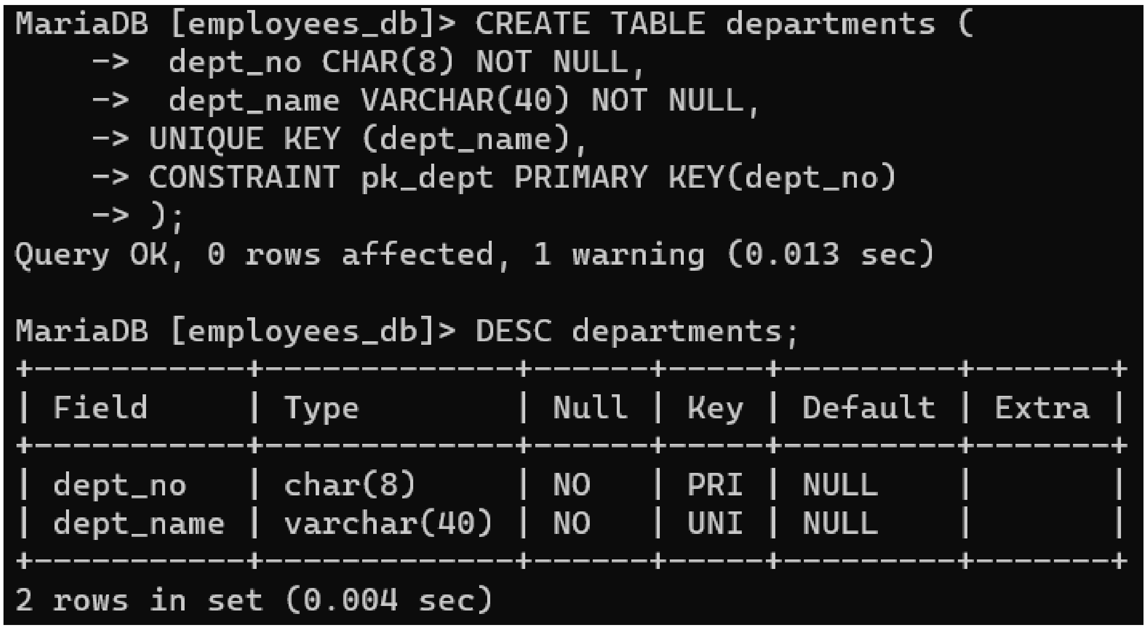
Creating the tables in the DATABASE

- Table employees:



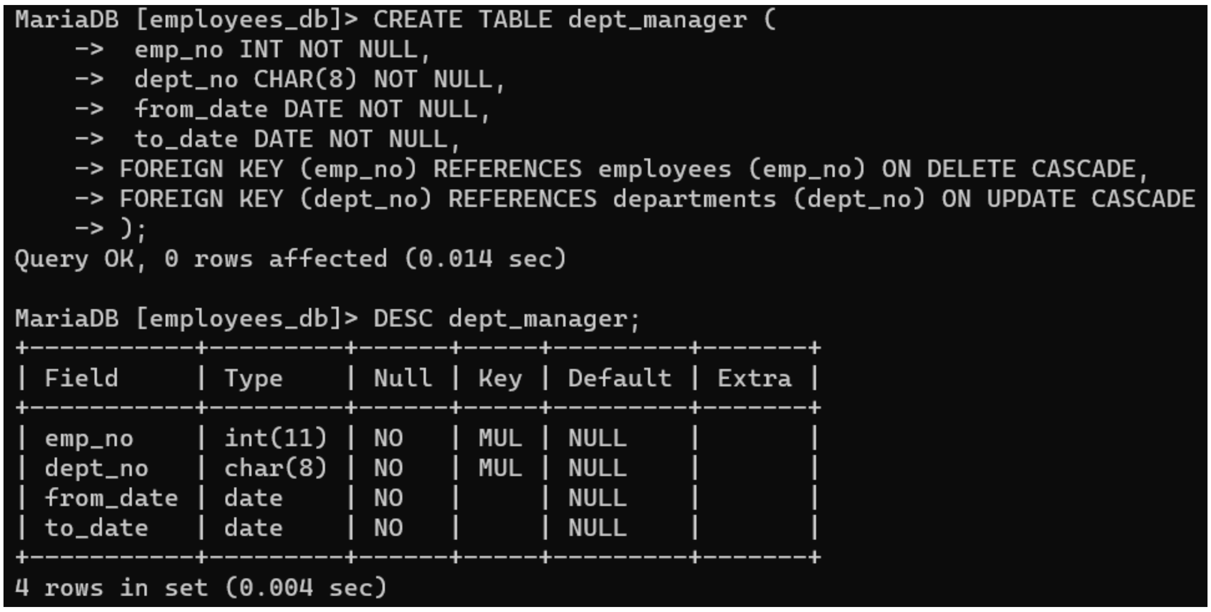
```
MariaDB [employees_db]> CREATE TABLE employees(
    ->  emp_no INT NOT NULL,
    ->  birth_date DATE NOT NULL,
    ->  first_name VARCHAR(14) NOT NULL,
    ->  last_name VARCHAR(16) NOT NULL,
    ->  gender ENUM('M','F') NOT NULL,
    ->  hire_date DATE NOT NULL,
    -> CONSTRAINT pk_emp PRIMARY KEY(emp_no)
    -> );
Query OK, 0 rows affected, 1 warning (0.018 sec)

MariaDB [employees_db]> DESC employees
    -> ;
+------------+---------------+------+-----+---------+-------+
| Field      | Type          | Null | Key | Default | Extra |
+------------+---------------+------+-----+---------+-------+
| emp_no     | int(11)       | NO   | PRI | NULL    |       |
| birth_date | date          | NO   |     | NULL    |       |
| first_name | varchar(14)   | NO   |     | NULL    |       |
| last_name  | varchar(16)   | NO   |     | NULL    |       |
| gender     | enum('M','F') | NO   |     | NULL    |       |
| hire_date  | date          | NO   |     | NULL    |       |
+------------+---------------+------+-----+---------+-------+
6 rows in set (0.004 sec)
```

- Departments table:



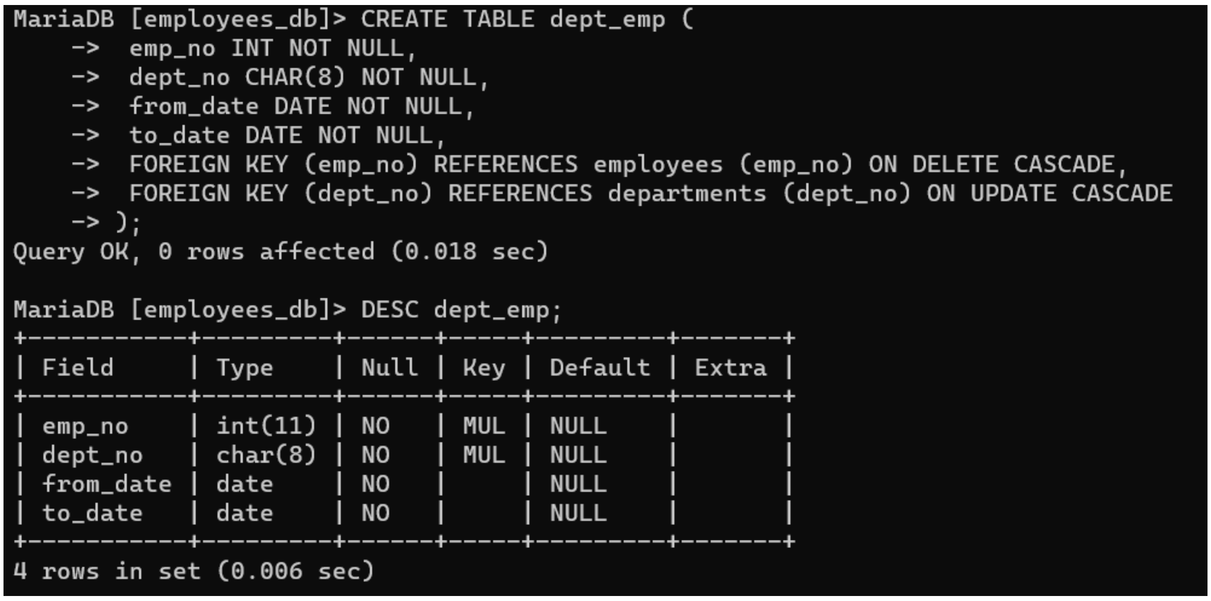
```
MariaDB [employees_db]> CREATE TABLE departments (
    ->  dept_no CHAR(8) NOT NULL,
    ->  dept_name VARCHAR(40) NOT NULL,
    -> UNIQUE KEY (dept_name),
    -> CONSTRAINT pk_dept PRIMARY KEY(dept_no)
    -> );
Query OK, 0 rows affected, 1 warning (0.013 sec)

MariaDB [employees_db]> DESC departments;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| dept_no   | char(8)     | NO   | PRI | NULL    |       |
| dept_name | varchar(40) | NO   | UNI | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
2 rows in set (0.004 sec)
```

- Dept_manager table:



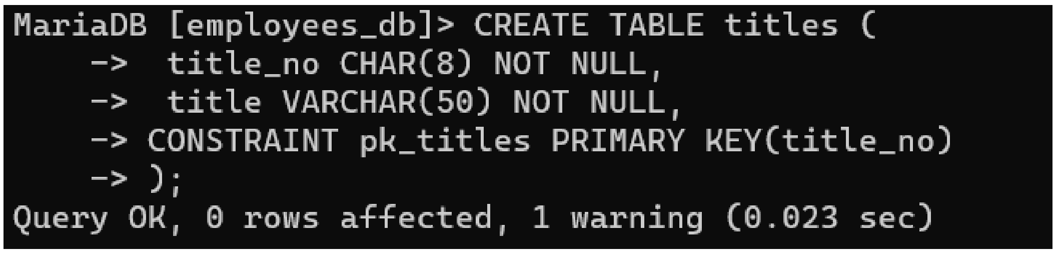
```
MariaDB [employees_db]> CREATE TABLE dept_manager (
    ->  emp_no INT NOT NULL,
    ->  dept_no CHAR(8) NOT NULL,
    ->  from_date DATE NOT NULL,
    ->  to_date DATE NOT NULL,
    -> FOREIGN KEY (emp_no) REFERENCES employees (emp_no) ON DELETE CASCADE,
    -> FOREIGN KEY (dept_no) REFERENCES departments (dept_no) ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.014 sec)

MariaDB [employees_db]> DESC dept_manager;
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| emp_no    | int(11) | NO   | MUL | NULL    |       |
| dept_no   | char(8) | NO   | MUL | NULL    |       |
| from_date | date    | NO   |     | NULL    |       |
| to_date   | date    | NO   |     | NULL    |       |
+-----------+---------+------+-----+---------+-------+
4 rows in set (0.004 sec)
```

- Dept_emp table:

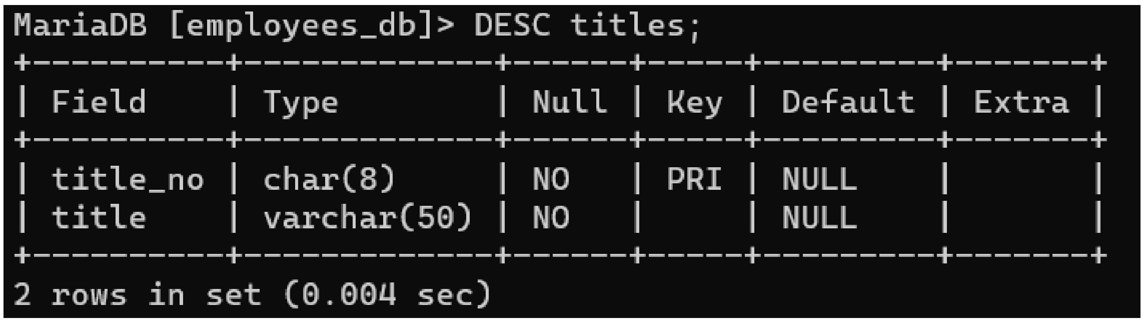


```
MariaDB [employees_db]> CREATE TABLE dept_emp (
    ->  emp_no INT NOT NULL,
    ->  dept_no CHAR(8) NOT NULL,
    ->  from_date DATE NOT NULL,
    ->  to_date DATE NOT NULL,
    ->  FOREIGN KEY (emp_no) REFERENCES employees (emp_no) ON DELETE CASCADE,
    ->  FOREIGN KEY (dept_no) REFERENCES departments (dept_no) ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.018 sec)

MariaDB [employees_db]> DESC dept_emp;
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| emp_no    | int(11) | NO   | MUL | NULL    |       |
| dept_no   | char(8) | NO   | MUL | NULL    |       |
| from_date | date    | NO   |     | NULL    |       |
| to_date   | date    | NO   |     | NULL    |       |
+-----------+---------+------+-----+---------+-------+
4 rows in set (0.006 sec)
```

- Titles table:

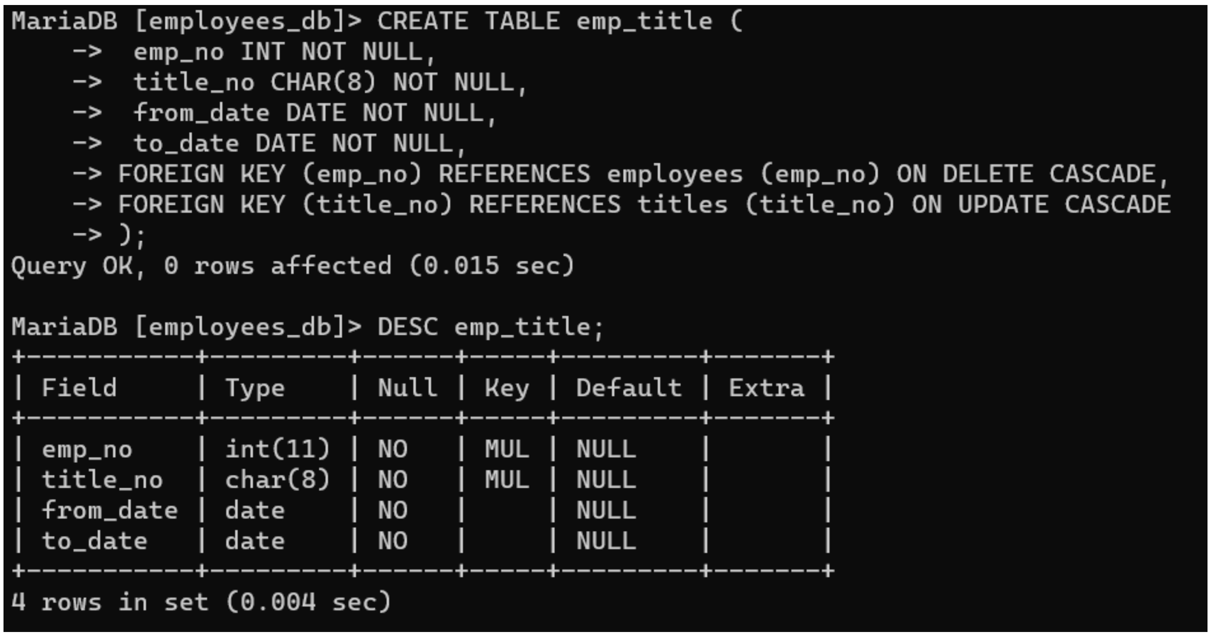


```
MariaDB [employees_db]> CREATE TABLE titles (
    ->  title_no CHAR(8) NOT NULL,
    ->  title VARCHAR(50) NOT NULL,
    -> CONSTRAINT pk_titles PRIMARY KEY(title_no)
    -> );
Query OK, 0 rows affected, 1 warning (0.023 sec)
```



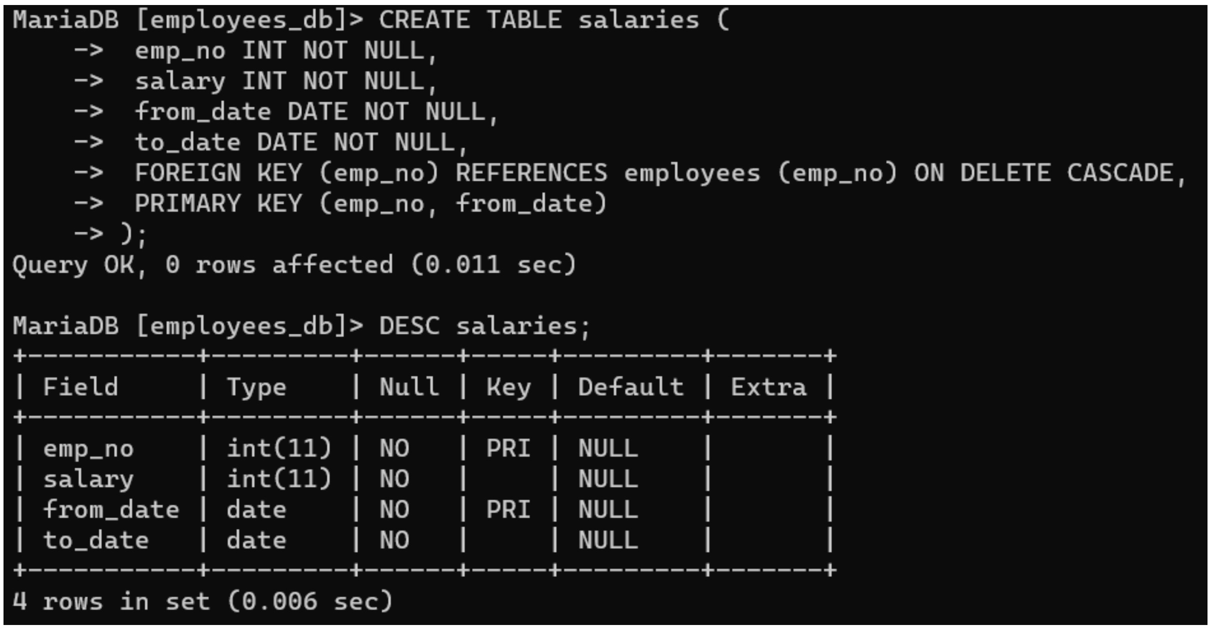
```
MariaDB [employees_db]> DESC titles;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| title_no | char(8)     | NO   | PRI | NULL    |       |
| title    | varchar(50) | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.004 sec)
```

- Emp_title table:



```
MariaDB [employees_db]> CREATE TABLE emp_title (
    ->  emp_no INT NOT NULL,
    ->  title_no CHAR(8) NOT NULL,
    ->  from_date DATE NOT NULL,
    ->  to_date DATE NOT NULL,
    -> FOREIGN KEY (emp_no) REFERENCES employees (emp_no) ON DELETE CASCADE,
    -> FOREIGN KEY (title_no) REFERENCES titles (title_no) ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.015 sec)

MariaDB [employees_db]> DESC emp_title;
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| emp_no    | int(11) | NO   | MUL | NULL    |       |
| title_no  | char(8) | NO   | MUL | NULL    |       |
| from_date | date    | NO   |     | NULL    |       |
| to_date   | date    | NO   |     | NULL    |       |
+-----------+---------+------+-----+---------+-------+
4 rows in set (0.004 sec)
```

- Salaries table:



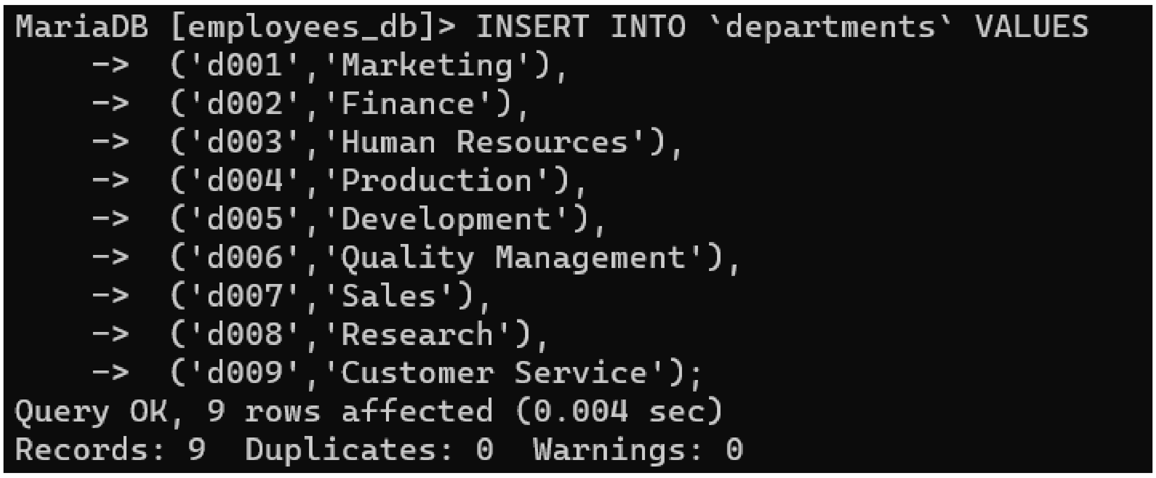
```
MariaDB [employees_db]> CREATE TABLE salaries (
    ->  emp_no INT NOT NULL,
    ->  salary INT NOT NULL,
    ->  from_date DATE NOT NULL,
    ->  to_date DATE NOT NULL,
    ->  FOREIGN KEY (emp_no) REFERENCES employees (emp_no) ON DELETE CASCADE,
    ->  PRIMARY KEY (emp_no, from_date)
    -> );
Query OK, 0 rows affected (0.011 sec)

MariaDB [employees_db]> DESC salaries;
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| emp_no    | int(11) | NO   | PRI | NULL    |       |
| salary    | int(11) | NO   |     | NULL    |       |
| from_date | date    | NO   | PRI | NULL    |       |
| to_date   | date    | NO   |     | NULL    |       |
+-----------+---------+------+-----+---------+-------+
4 rows in set (0.006 sec)
```

Inserting the data:

Let's not forget to restructure our insert statements to reflect the new structure. We have added transformed the titles table in 2 tables:

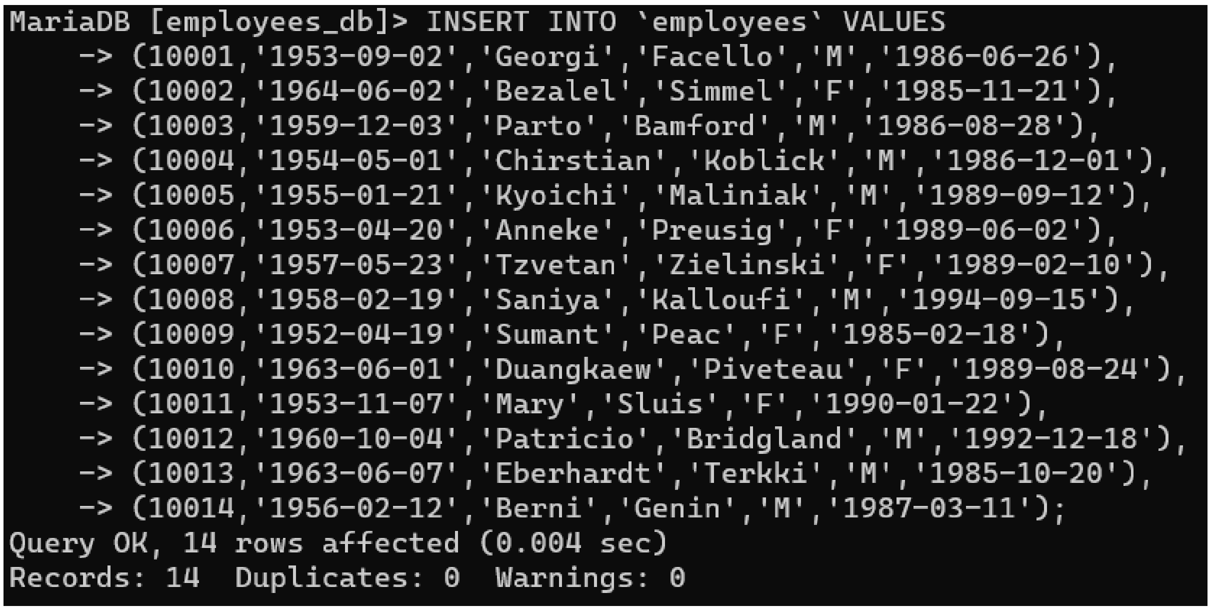
- Emp_title to record emp_no, title_no, from_date and to_date.
- Titles table to list all possible titles with a column id as primary key to reference.

1. Department values:



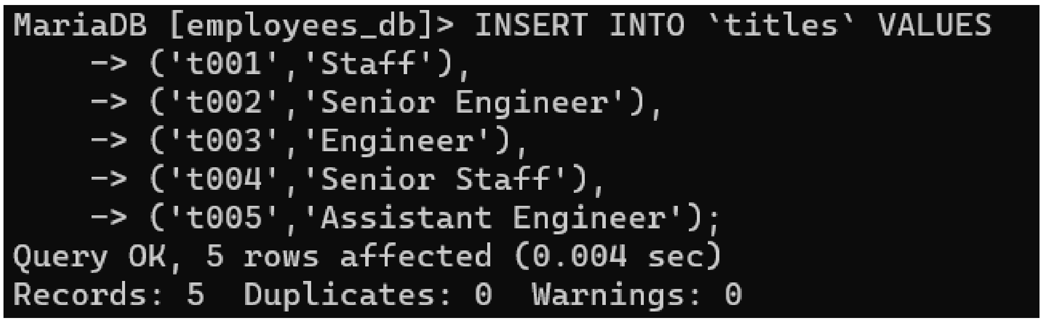
```
MariaDB [employees_db]> INSERT INTO `departments` VALUES
    -> ('d001','Marketing'),
    -> ('d002','Finance'),
    -> ('d003','Human Resources'),
    -> ('d004','Production'),
    -> ('d005','Development'),
    -> ('d006','Quality Management'),
    -> ('d007','Sales'),
    -> ('d008','Research'),
    -> ('d009','Customer Service');
Query OK, 9 rows affected (0.004 sec)
Records: 9  Duplicates: 0  Warnings: 0
```

2. Employees values:



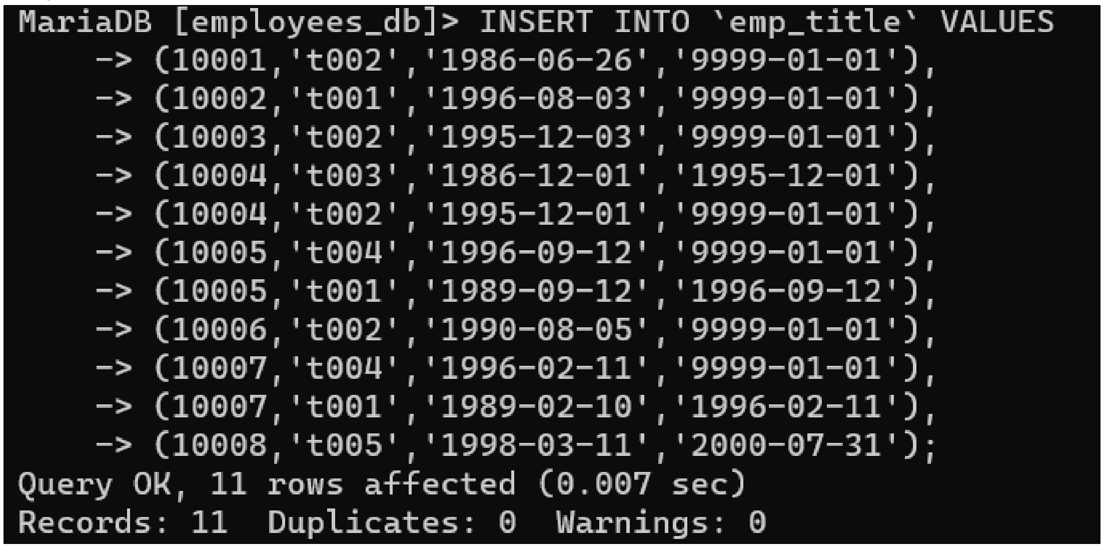
```
MariaDB [employees_db]> INSERT INTO `employees` VALUES
    -> (10001,'1953-09-02','Georgi','Facello','M','1986-06-26'),
    -> (10002,'1964-06-02','Bezalel','Simmel','F','1985-11-21'),
    -> (10003,'1959-12-03','Parto','Bamford','M','1986-08-28'),
    -> (10004,'1954-05-01','Chirstian','Koblick','M','1986-12-01'),
    -> (10005,'1955-01-21','Kyoichi','Maliniak','M','1989-09-12'),
    -> (10006,'1953-04-20','Anneke','Preusig','F','1989-06-02'),
    -> (10007,'1957-05-23','Tzvetan','Zielinski','F','1989-02-10'),
    -> (10008,'1958-02-19','Saniya','Kalloufi','M','1994-09-15'),
    -> (10009,'1952-04-19','Sumant','Peac','F','1985-02-18'),
    -> (10010,'1963-06-01','Duangkaew','Piveteau','F','1989-08-24'),
    -> (10011,'1953-11-07','Mary','Sluis','F','1990-01-22'),
    -> (10012,'1960-10-04','Patricio','Bridgland','M','1992-12-18'),
    -> (10013,'1963-06-07','Eberhardt','Terkki','M','1985-10-20'),
    -> (10014,'1956-02-12','Berni','Genin','M','1987-03-11');
Query OK, 14 rows affected (0.004 sec)
Records: 14  Duplicates: 0  Warnings: 0
```

3. Titles table:



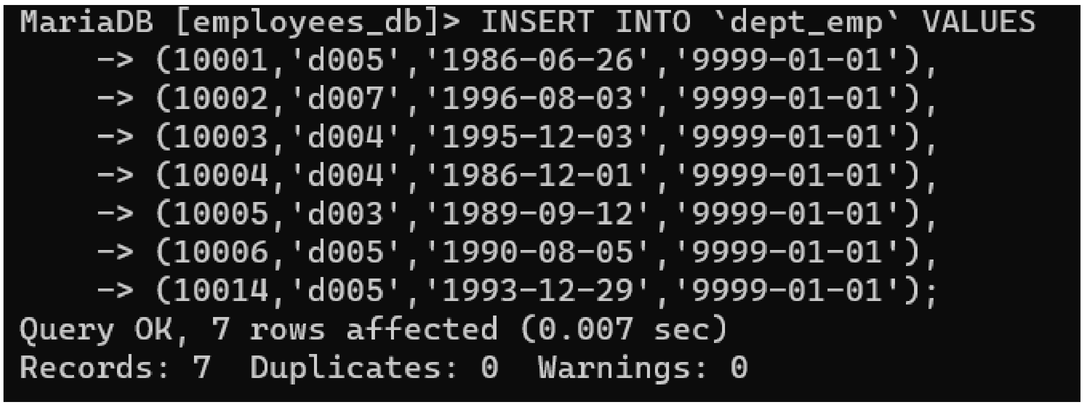
```
MariaDB [employees_db]> INSERT INTO `titles` VALUES
    -> ('t001','Staff'),
    -> ('t002','Senior Engineer'),
    -> ('t003','Engineer'),
    -> ('t004','Senior Staff'),
    -> ('t005','Assistant Engineer');
Query OK, 5 rows affected (0.004 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

4. Emp_title table:



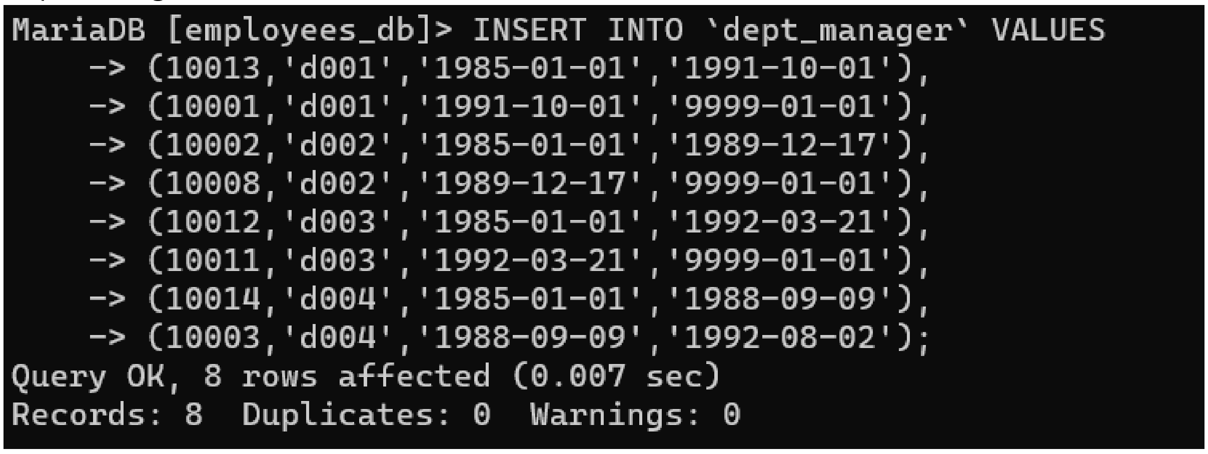
```
MariaDB [employees_db]> INSERT INTO `emp_title` VALUES
    -> (10001,'t002','1986-06-26','9999-01-01'),
    -> (10002,'t001','1996-08-03','9999-01-01'),
    -> (10003,'t002','1995-12-03','9999-01-01'),
    -> (10004,'t003','1986-12-01','1995-12-01'),
    -> (10004,'t002','1995-12-01','9999-01-01'),
    -> (10005,'t004','1996-09-12','9999-01-01'),
    -> (10005,'t001','1989-09-12','1996-09-12'),
    -> (10006,'t002','1990-08-05','9999-01-01'),
    -> (10007,'t004','1996-02-11','9999-01-01'),
    -> (10007,'t001','1989-02-10','1996-02-11'),
    -> (10008,'t005','1998-03-11','2000-07-31');
Query OK, 11 rows affected (0.007 sec)
Records: 11  Duplicates: 0  Warnings: 0
```

5. Dept_emp table:



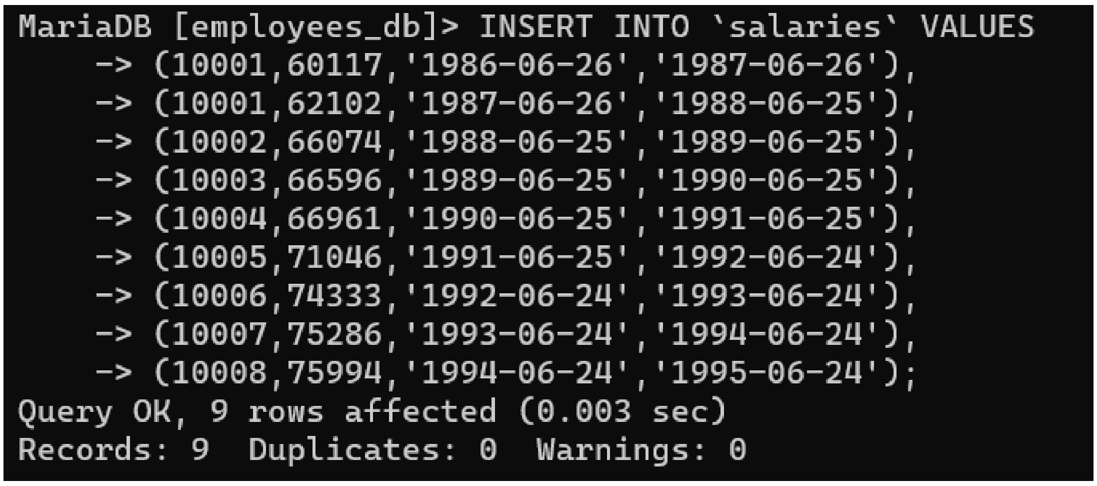
```
MariaDB [employees_db]> INSERT INTO `dept_emp` VALUES
    -> (10001,'d005','1986-06-26','9999-01-01'),
    -> (10002,'d007','1996-08-03','9999-01-01'),
    -> (10003,'d004','1995-12-03','9999-01-01'),
    -> (10004,'d004','1986-12-01','9999-01-01'),
    -> (10005,'d003','1989-09-12','9999-01-01'),
    -> (10006,'d005','1990-08-05','9999-01-01'),
    -> (10014,'d005','1993-12-29','9999-01-01');
Query OK, 7 rows affected (0.007 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

6. Dept_manager table:



```
MariaDB [employees_db]> INSERT INTO `dept_manager` VALUES
    -> (10013,'d001','1985-01-01','1991-10-01'),
    -> (10001,'d001','1991-10-01','9999-01-01'),
    -> (10002,'d002','1985-01-01','1989-12-17'),
    -> (10008,'d002','1989-12-17','9999-01-01'),
    -> (10012,'d003','1985-01-01','1992-03-21'),
    -> (10011,'d003','1992-03-21','9999-01-01'),
    -> (10014,'d004','1985-01-01','1988-09-09'),
    -> (10003,'d004','1988-09-09','1992-08-02');
Query OK, 8 rows affected (0.007 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

7. Salaries table:

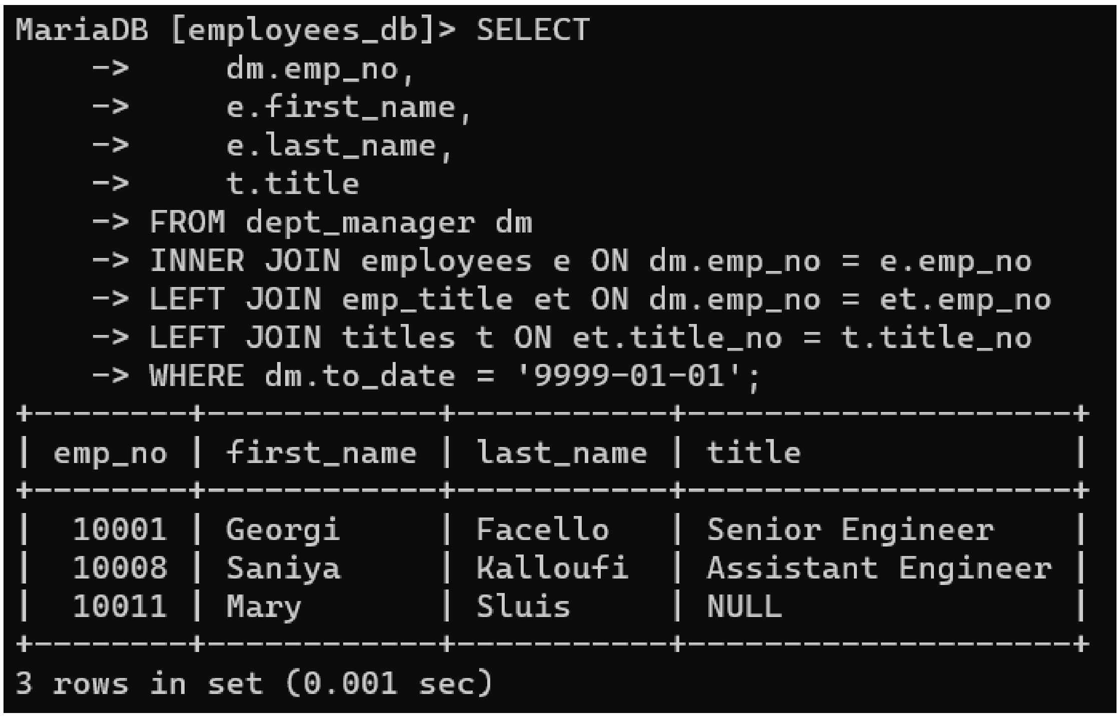


```
MariaDB [employees_db]> INSERT INTO `salaries` VALUES
    -> (10001,60117,'1986-06-26','1987-06-26'),
    -> (10001,62102,'1987-06-26','1988-06-25'),
    -> (10002,66074,'1988-06-25','1989-06-25'),
    -> (10003,66596,'1989-06-25','1990-06-25'),
    -> (10004,66961,'1990-06-25','1991-06-25'),
    -> (10005,71046,'1991-06-25','1992-06-24'),
    -> (10006,74333,'1992-06-24','1993-06-24'),
    -> (10007,75286,'1993-06-24','1994-06-24'),
    -> (10008,75994,'1994-06-24','1995-06-24');
Query OK, 9 rows affected (0.003 sec)
Records: 9  Duplicates: 0  Warnings: 0
```

Creating the queries:

1. Create a SQL statement to list all managers and their titles.

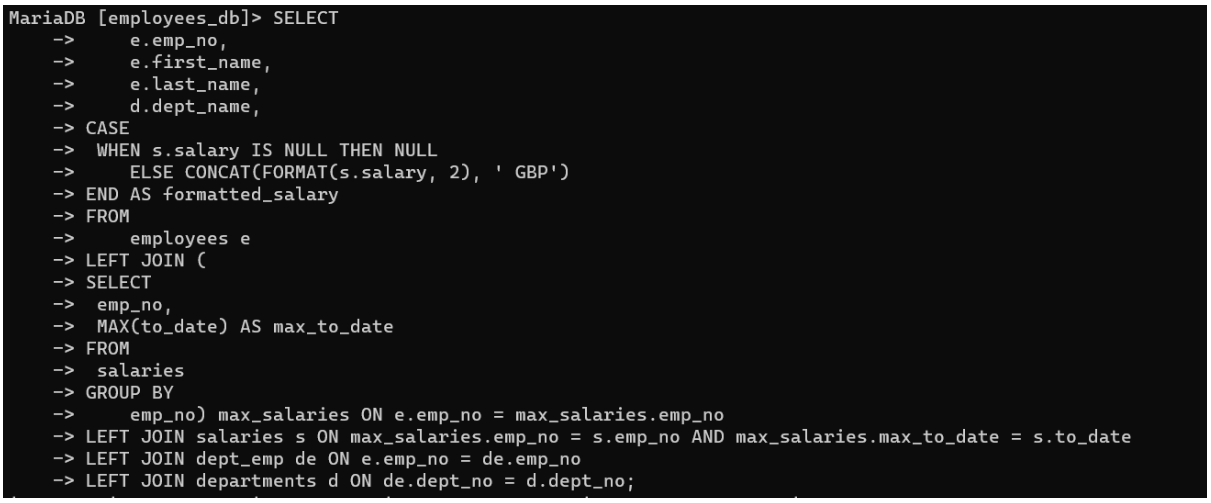
I decide to select first from the dept_manager table to make sure that we capture all employees registered as managers.
I add a where statement to select only current managers.
I use LEFT JOIN to make sure that I also include managers for whom I have missing title data. I then know they manage a department, even if I don't know their title.



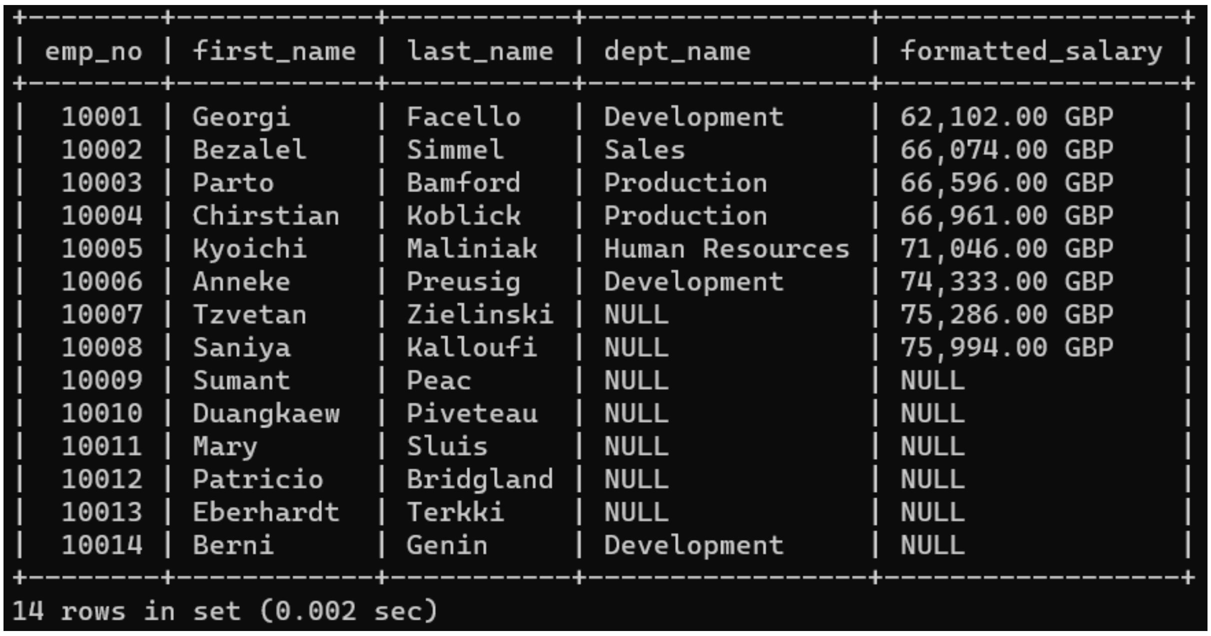
```
MariaDB [employees_db]> SELECT
    ->     dm.emp_no,
    ->     e.first_name,
    ->     e.last_name,
    ->     t.title
    -> FROM dept_manager dm
    -> INNER JOIN employees e ON dm.emp_no = e.emp_no
    -> LEFT JOIN emp_title et ON dm.emp_no = et.emp_no
    -> LEFT JOIN titles t ON et.title_no = t.title_no
    -> WHERE dm.to_date = '9999-01-01';
+--------+------------+-----------+--------------------+
| emp_no | first_name | last_name | title              |
+--------+------------+-----------+--------------------+
|  10001 | Georgi     | Facello   | Senior Engineer    |
|  10008 | Saniya     | Kalloufi  | Assistant Engineer |
|  10011 | Mary       | Sluis     | NULL               |
+--------+------------+-----------+--------------------+
3 rows in set (0.001 sec)
```

2. Create a SQL statement to show the salary of all employees and their department name.

Here I start from the employee table to make sure I include the employees for which I do not have a salary or a department. I run into duplicates so I select the most recent salary.I format the salary for easier reading



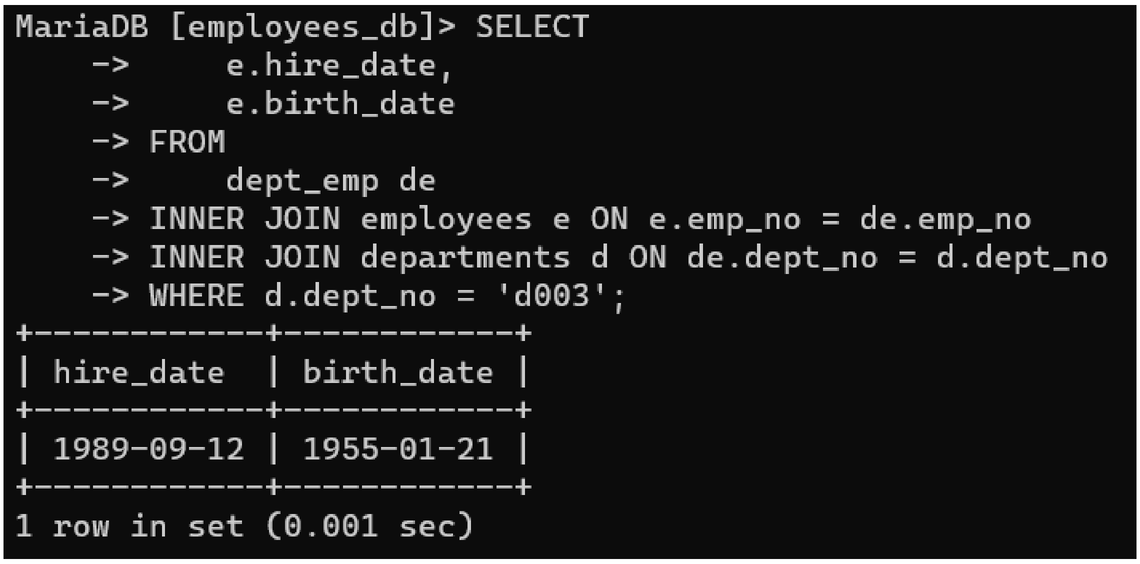
```
MariaDB [employees_db]> SELECT
    ->     e.emp_no,
    ->     e.first_name,
    ->     e.last_name,
    ->     d.dept_name,
    -> CASE
    ->  WHEN s.salary IS NULL THEN NULL
    ->     ELSE CONCAT(FORMAT(s.salary, 2), ' GBP')
    -> END AS formatted_salary
    -> FROM
    ->     employees e
    -> LEFT JOIN (
    -> SELECT
    ->  emp_no,
    ->  MAX(to_date) AS max_to_date
    -> FROM
    ->  salaries
    -> GROUP BY
    ->     emp_no) max_salaries ON e.emp_no = max_salaries.emp_no
    -> LEFT JOIN salaries s ON max_salaries.emp_no = s.emp_no AND max_salaries.max_to_date = s.to_date
    -> LEFT JOIN dept_emp de ON e.emp_no = de.emp_no
    -> LEFT JOIN departments d ON de.dept_no = d.dept_no;
```



```
+--------+------------+-----------+-----------------+------------------+
| emp_no | first_name | last_name | dept_name       | formatted_salary |
+--------+------------+-----------+-----------------+------------------+
|  10001 | Georgi     | Facello   | Development     | 62,102.00 GBP    |
|  10002 | Bezalel    | Simmel    | Sales           | 66,074.00 GBP    |
|  10003 | Parto      | Bamford   | Production      | 66,596.00 GBP    |
|  10004 | Chirstian  | Koblick   | Production      | 66,961.00 GBP    |
|  10005 | Kyoichi    | Maliniak  | Human Resources | 71,046.00 GBP    |
|  10006 | Anneke     | Preusig   | Development     | 74,333.00 GBP    |
|  10007 | Tzvetan    | Zielinski | NULL            | 75,286.00 GBP    |
|  10008 | Saniya     | Kalloufi  | NULL            | 75,994.00 GBP    |
|  10009 | Sumant     | Peac      | NULL            | NULL             |
|  10010 | Duangkaew  | Piveteau  | NULL            | NULL             |
|  10011 | Mary       | Sluis     | NULL            | NULL             |
|  10012 | Patricio   | Bridgland | NULL            | NULL             |
|  10013 | Eberhardt  | Terkki    | NULL            | NULL             |
|  10014 | Berni      | Genin     | Development     | NULL             |
+--------+------------+-----------+-----------------+------------------+
14 rows in set (0.002 sec)
```

3. Create a SQL statement to show the hire date and birth date who belongs to HR department

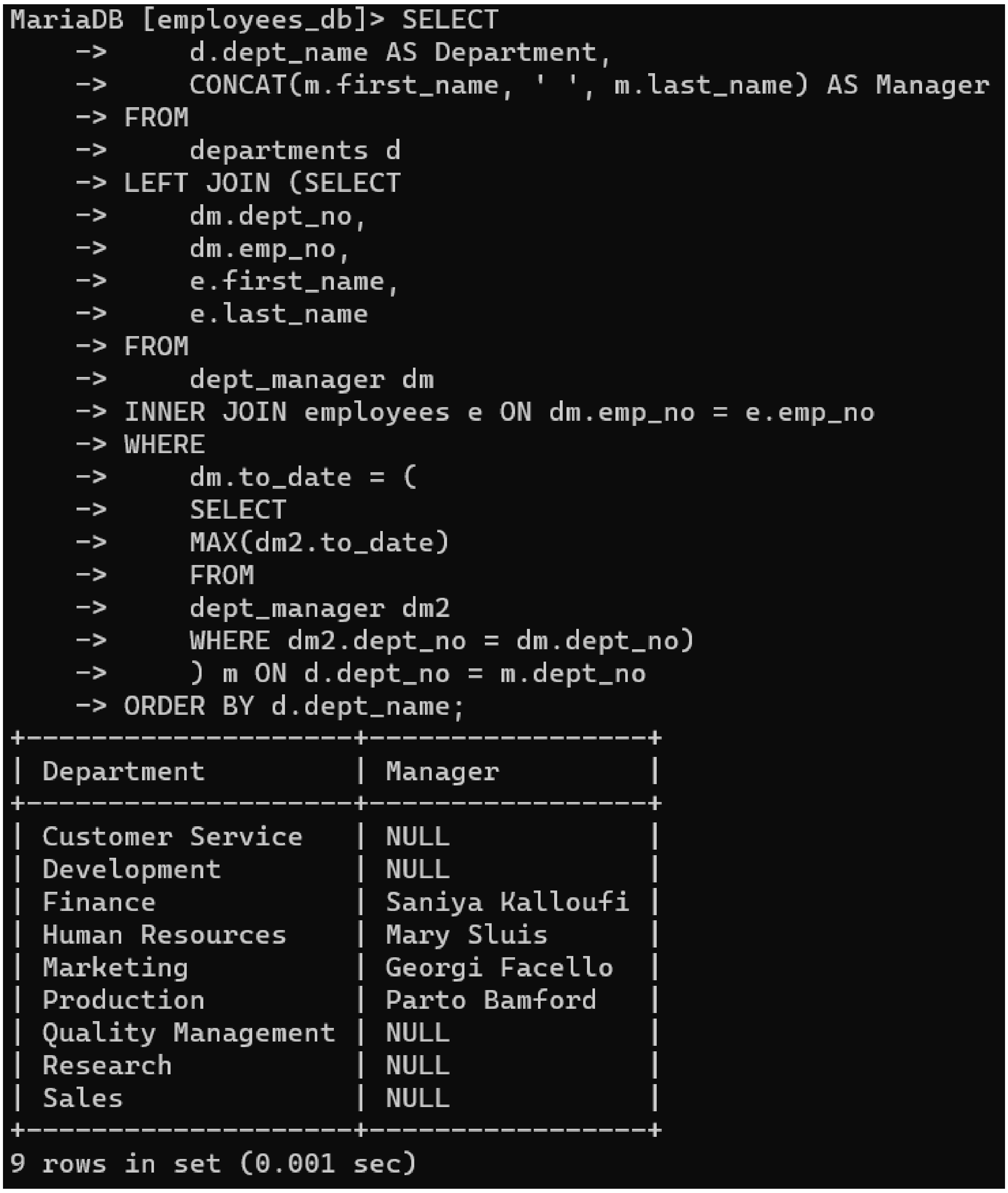
In this statement, I start from the dept_emp table to make sure that the employee is registered in a department. I then join to both department and employee tables with a where clause to select only those in the HR department.



```
MariaDB [employees_db]> SELECT
    ->     e.hire_date,
    ->     e.birth_date
    -> FROM
    ->     dept_emp de
    -> INNER JOIN employees e ON e.emp_no = de.emp_no
    -> INNER JOIN departments d ON de.dept_no = d.dept_no
    -> WHERE d.dept_no = 'd003';
+------------+------------+
| hire_date  | birth_date |
+------------+------------+
| 1989-09-12 | 1955-01-21 |
+------------+------------+
1 row in set (0.001 sec)
```

4. Create a SQL statement to show all departments and their department's managers.

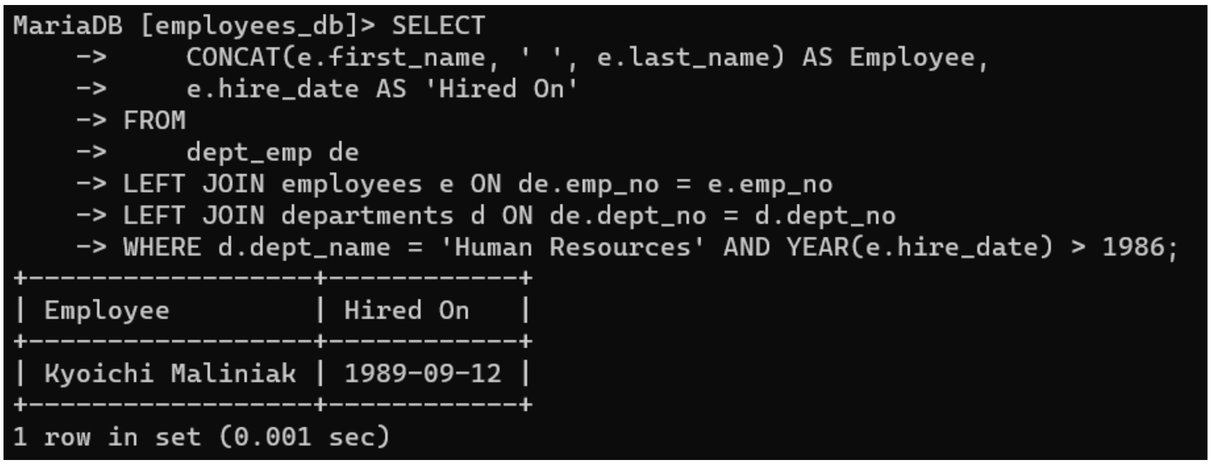
Here we start by concatenating the names and ordering departments for ease of reading. We use a left join to make sure we have all the departments, even if no manager. We then make sure we select the most recent/current manager for each department



```
MariaDB [employees_db]> SELECT
    ->     d.dept_name AS Department,
    ->     CONCAT(m.first_name, ' ', m.last_name) AS Manager
    -> FROM
    ->     departments d
    -> LEFT JOIN (SELECT
    ->     dm.dept_no,
    ->     dm.emp_no,
    ->     e.first_name,
    ->     e.last_name
    -> FROM
    ->     dept_manager dm
    -> INNER JOIN employees e ON dm.emp_no = e.emp_no
    -> WHERE
    ->     dm.to_date = (
    ->     SELECT
    ->     MAX(dm2.to_date)
    ->     FROM
    ->     dept_manager dm2
    ->     WHERE dm2.dept_no = dm.dept_no)
    ->     ) m ON d.dept_no = m.dept_no
    -> ORDER BY d.dept_name;
+--------------------+-----------------+
| Department         | Manager         |
+--------------------+-----------------+
| Customer Service   | NULL            |
| Development        | NULL            |
| Finance            | Saniya Kalloufi |
| Human Resources    | Mary Sluis      |
| Marketing          | Georgi Facello  |
| Production         | Parto Bamford   |
| Quality Management | NULL            |
| Research           | NULL            |
| Sales              | NULL            |
+--------------------+-----------------+
9 rows in set (0.001 sec)
```

5. Create a SQL statement to show a list of HR's employees who were hired after 1986

Here we start from the dept_emp table to make sure we have all registered HR employees. We then use left join to ensure we cover potential missing data in related tables. We change the column headers for ease of reading.

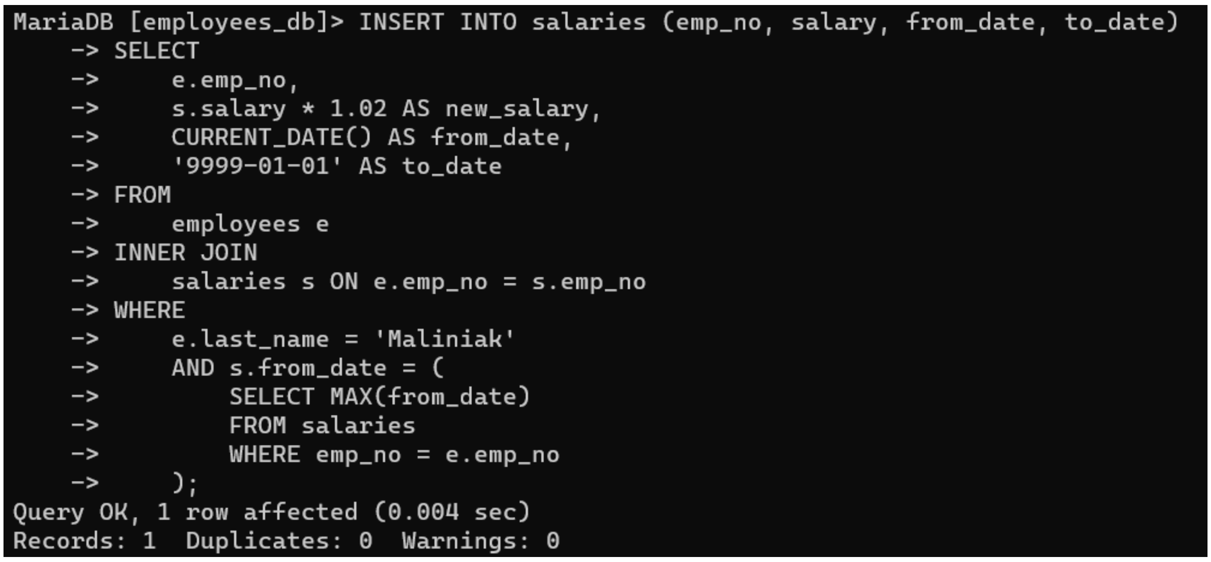


```
MariaDB [employees_db]> SELECT
    ->     CONCAT(e.first_name, ' ', e.last_name) AS Employee,
    ->     e.hire_date AS 'Hired On'
    -> FROM
    ->     dept_emp de
    -> LEFT JOIN employees e ON de.emp_no = e.emp_no
    -> LEFT JOIN departments d ON de.dept_no = d.dept_no
    -> WHERE d.dept_name = 'Human Resources' AND YEAR(e.hire_date) > 1986;
+------------------+------------+
| Employee         | Hired On   |
+------------------+------------+
| Kyoichi Maliniak | 1989-09-12 |
+------------------+------------+
1 row in set (0.001 sec)
```

6. Create a SQL statement to increase any employee's salary up to 2%. Assume the employee has just phoned in with his/her last name.

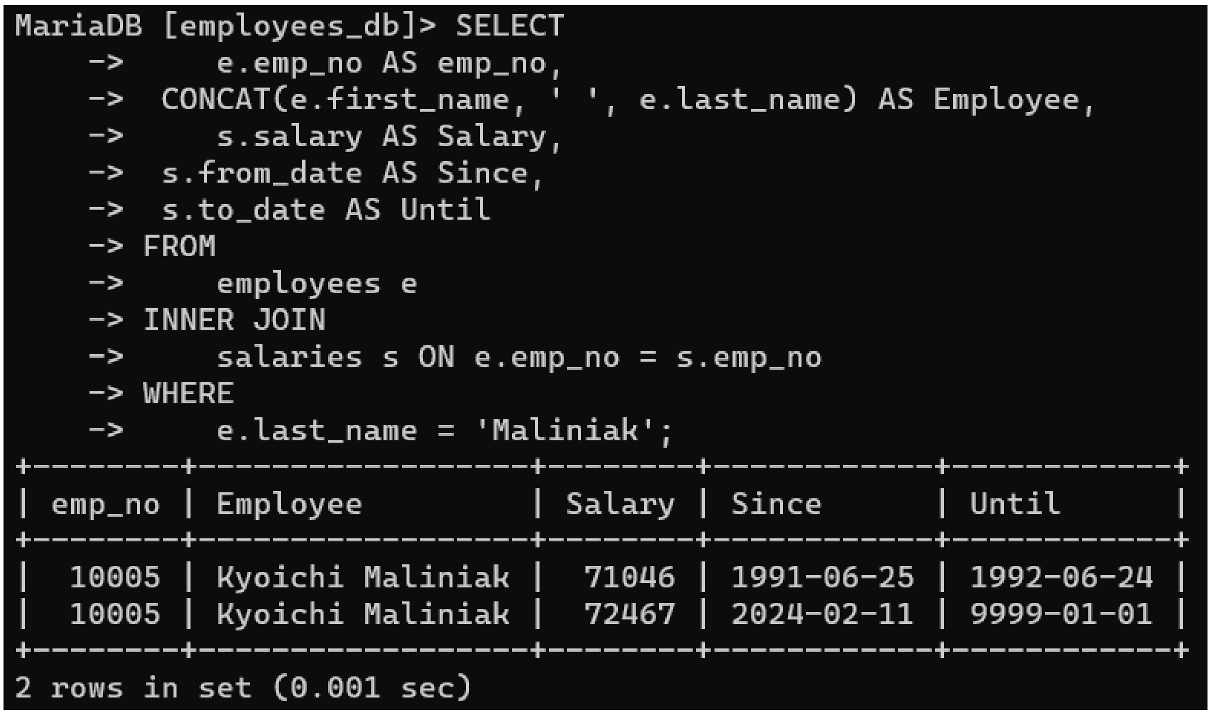
In this scenario:

- First I calculate the new salary. I must make sure I add 2% to the employee's most recent salary.
- Then I create a new record in the salaries table.
- Then I update the old salary. We decide to go with the employee called Maliniak:



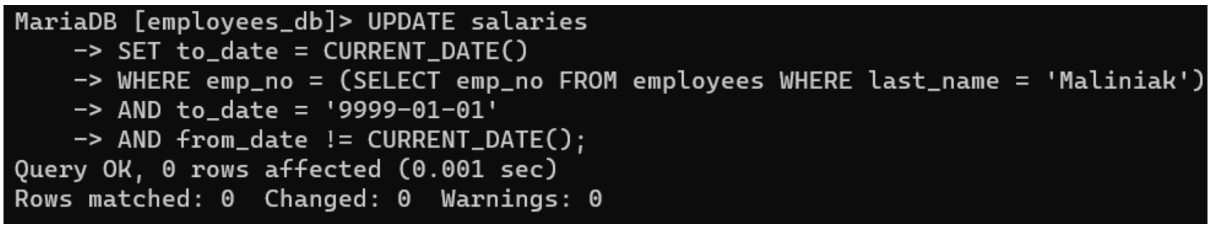
```
MariaDB [employees_db]> INSERT INTO salaries (emp_no, salary, from_date, to_date)
    -> SELECT
    ->     e.emp_no,
    ->     s.salary * 1.02 AS new_salary,
    ->     CURRENT_DATE() AS from_date,
    ->     '9999-01-01' AS to_date
    -> FROM
    ->     employees e
    -> INNER JOIN
    ->     salaries s ON e.emp_no = s.emp_no
    -> WHERE
    ->     e.last_name = 'Maliniak'
    ->     AND s.from_date = (
    ->         SELECT MAX(from_date)
    ->         FROM salaries
    ->         WHERE emp_no = e.emp_no
    ->     );
Query OK, 1 row affected (0.004 sec)
Records: 1  Duplicates: 0  Warnings: 0
```

Now a quick check to make sure our new salary is added:



```
MariaDB [employees_db]> SELECT
    ->     e.emp_no AS emp_no,
    ->  CONCAT(e.first_name, ' ', e.last_name) AS Employee,
    ->     s.salary AS Salary,
    ->  s.from_date AS Since,
    ->  s.to_date AS Until
    -> FROM
    ->     employees e
    -> INNER JOIN
    ->     salaries s ON e.emp_no = s.emp_no
    -> WHERE
    ->     e.last_name = 'Maliniak';
+--------+------------------+--------+------------+------------+
| emp_no | Employee         | Salary | Since      | Until      |
+--------+------------------+--------+------------+------------+
|  10005 | Kyoichi Maliniak |  71046 | 1991-06-25 | 1992-06-24 |
|  10005 | Kyoichi Maliniak |  72467 | 2024-02-11 | 9999-01-01 |
+--------+------------------+--------+------------+------------+
2 rows in set (0.001 sec)
```

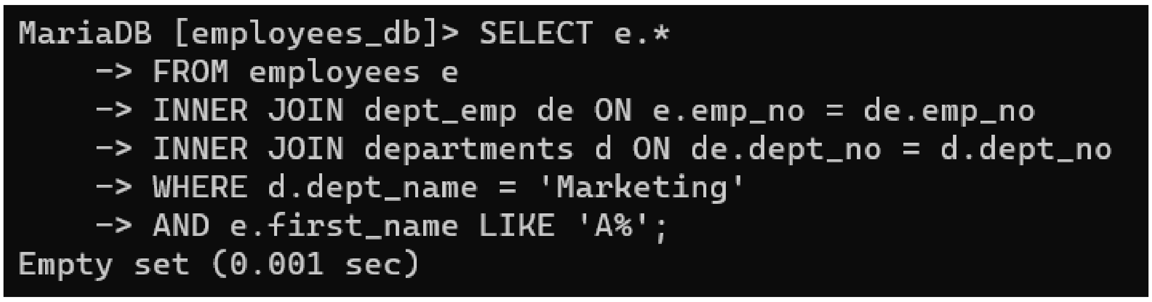
And now, we must make sure to update the old salary. In our example the employee did not have his current salary and only had one salary entered so we didn't encounter problem. But if an employee has a current salary (to_date ‘9999-01-01) then we must change it to current date for consistency.



```
MariaDB [employees_db]> UPDATE salaries
    -> SET to_date = CURRENT_DATE()
    -> WHERE emp_no = (SELECT emp_no FROM employees WHERE last_name = 'Maliniak')
    -> AND to_date = '9999-01-01'
    -> AND from_date != CURRENT_DATE();
Query OK, 0 rows affected (0.001 sec)
Rows matched: 0  Changed: 0  Warnings: 0
```

7. Create a SQL statement to delete employee’s record who belongs to marketing department and name start with A

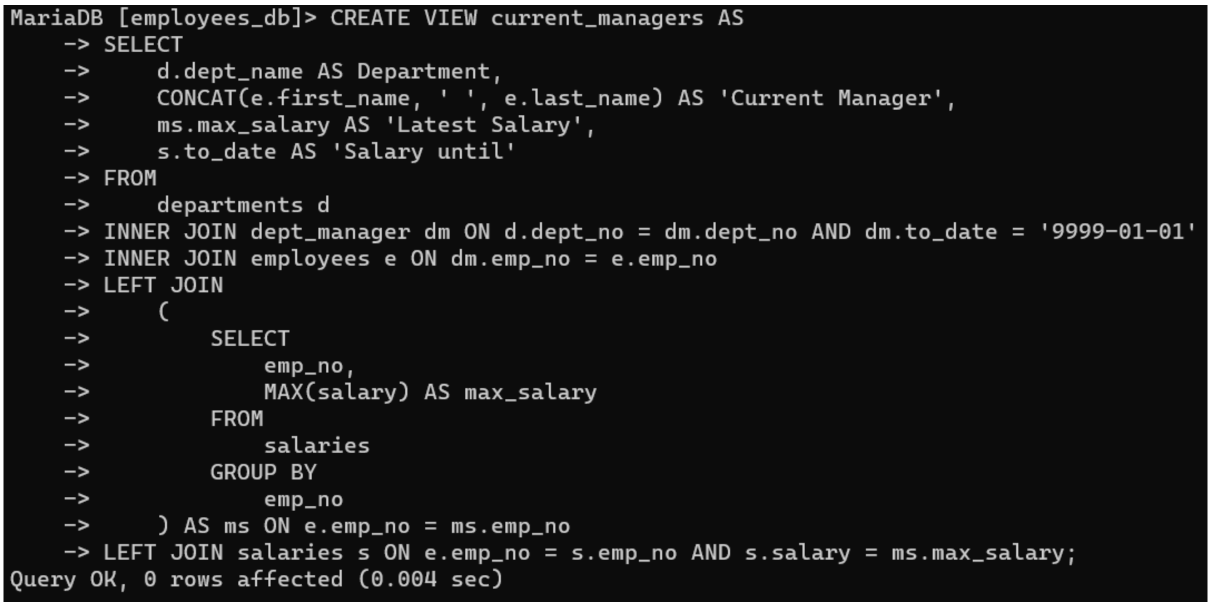
Instead of deleting straight away, I create the query to select the same sample of employees. I can then check that the query works. Then I only need to replace SELECT by DELETE and it will perform the deletion on exactly the same selection.



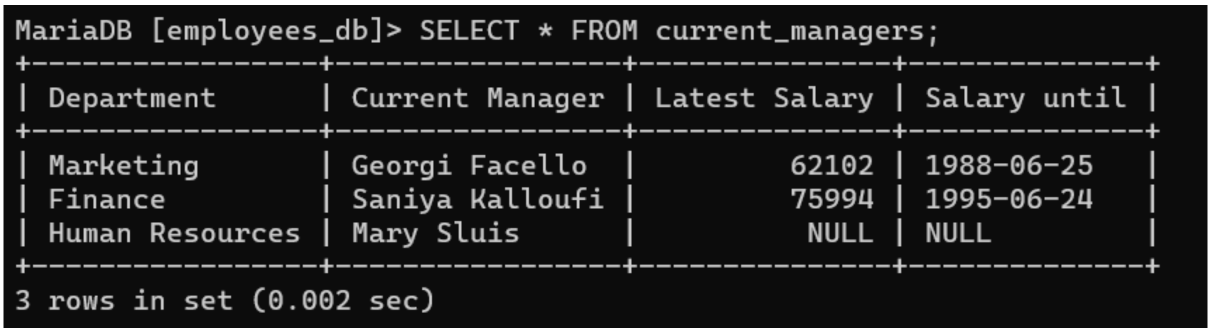
```
MariaDB [employees_db]> SELECT e.*
    -> FROM employees e
    -> INNER JOIN dept_emp de ON e.emp_no = de.emp_no
    -> INNER JOIN departments d ON de.dept_no = d.dept_no
    -> WHERE d.dept_name = 'Marketing'
    -> AND e.first_name LIKE 'A%';
Empty set (0.001 sec)
```

8. Create a database view to list the full names of all departments' managers, and their salaries

After querying the data, it appears that we do not have a list where we have all current department managers with all their current salaries. I have decided to provide a view of all the current managers, their department, and the relevant salary information (amount, and to date to understand if this is their current or an old salary).



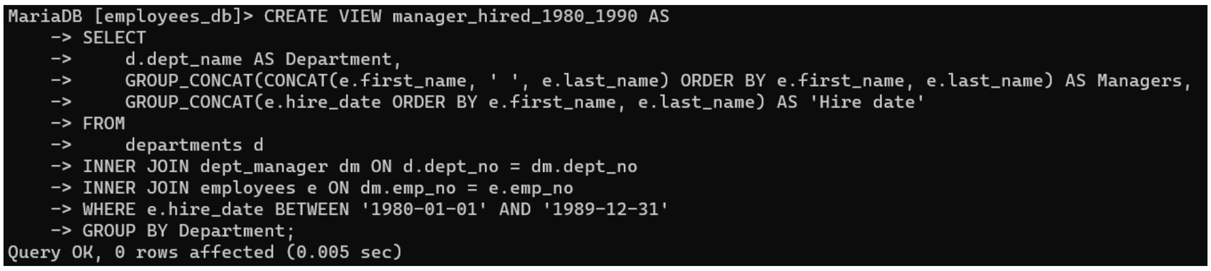
```
MariaDB [employees_db]> CREATE VIEW current_managers AS
    -> SELECT
    ->     d.dept_name AS Department,
    ->     CONCAT(e.first_name, ' ', e.last_name) AS 'Current Manager',
    ->     ms.max_salary AS 'Latest Salary',
    ->     s.to_date AS 'Salary until'
    -> FROM
    ->     departments d
    -> INNER JOIN dept_manager dm ON d.dept_no = dm.dept_no AND dm.to_date = '9999-01-01'
    -> INNER JOIN employees e ON dm.emp_no = e.emp_no
    -> LEFT JOIN
    ->     (
    ->         SELECT
    ->             emp_no,
    ->             MAX(salary) AS max_salary
    ->         FROM
    ->             salaries
    ->         GROUP BY
    ->             emp_no
    ->     ) AS ms ON e.emp_no = ms.emp_no
    -> LEFT JOIN salaries s ON e.emp_no = s.emp_no AND s.salary = ms.max_salary;
Query OK, 0 rows affected (0.004 sec)
```



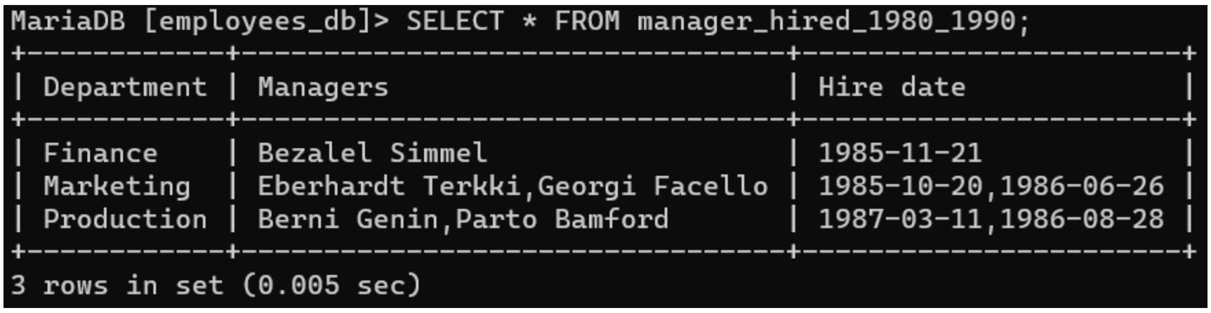
```
MariaDB [employees_db]> SELECT * FROM current_managers;
+-----------------+-----------------+---------------+--------------+
| Department      | Current Manager | Latest Salary | Salary until |
+-----------------+-----------------+---------------+--------------+
| Marketing       | Georgi Facello  |         62102 | 1988-06-25   |
| Finance         | Saniya Kalloufi |         75994 | 1995-06-24   |
| Human Resources | Mary Sluis      |          NULL | NULL         |
+-----------------+-----------------+---------------+--------------+
3 rows in set (0.002 sec)
```

9. Create a database view to list all departments and their department's managers, who were hired between 1980 and 1990

I decide to go for all the manager names, grouped by department with the hire date between 1980-01-01 and 1989-12-31



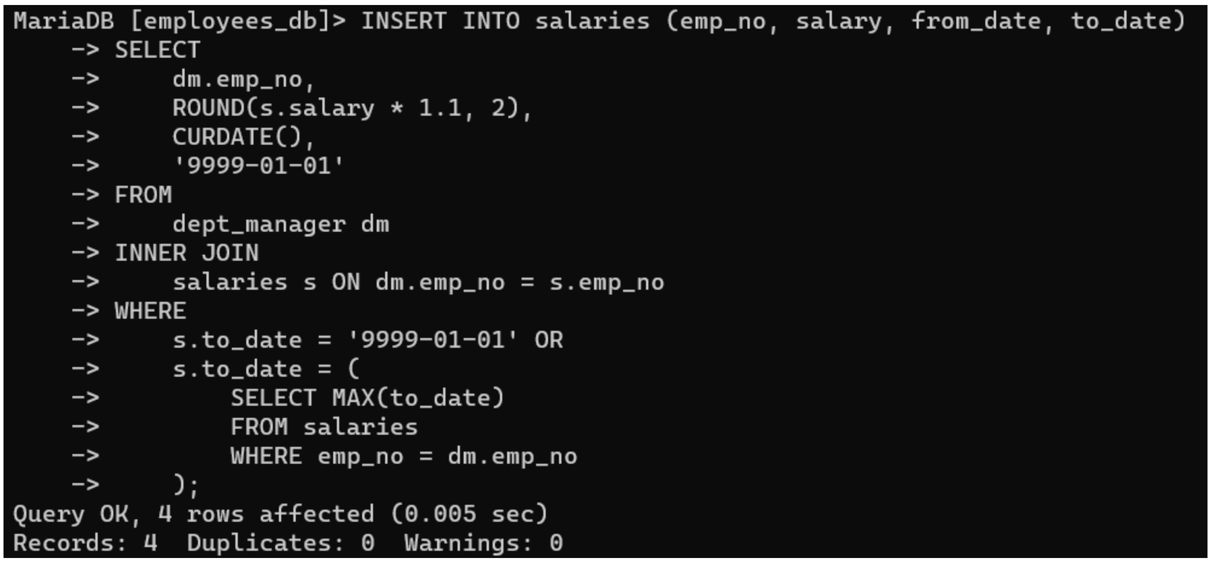
```
MariaDB [employees_db]> CREATE VIEW manager_hired_1980_1990 AS
    -> SELECT
    ->     d.dept_name AS Department,
    ->     GROUP_CONCAT(CONCAT(e.first_name, ' ', e.last_name) ORDER BY e.first_name, e.last_name) AS Managers,
    ->     GROUP_CONCAT(e.hire_date ORDER BY e.first_name, e.last_name) AS 'Hire date'
    -> FROM
    ->     departments d
    -> INNER JOIN dept_manager dm ON d.dept_no = dm.dept_no
    -> INNER JOIN employees e ON dm.emp_no = e.emp_no
    -> WHERE e.hire_date BETWEEN '1980-01-01' AND '1989-12-31'
    -> GROUP BY Department;
Query OK, 0 rows affected (0.005 sec)
```



```
MariaDB [employees_db]> SELECT * FROM manager_hired_1980_1990;
+------------+---------------------------------+-----------------------+
| Department | Managers                        | Hire date             |
+------------+---------------------------------+-----------------------+
| Finance    | Bezalel Simmel                  | 1985-11-21            |
| Marketing  | Eberhardt Terkki,Georgi Facello | 1985-10-20,1986-06-26 |
| Production | Berni Genin,Parto Bamford       | 1987-03-11,1986-08-28 |
+------------+---------------------------------+-----------------------+
3 rows in set (0.005 sec)
```

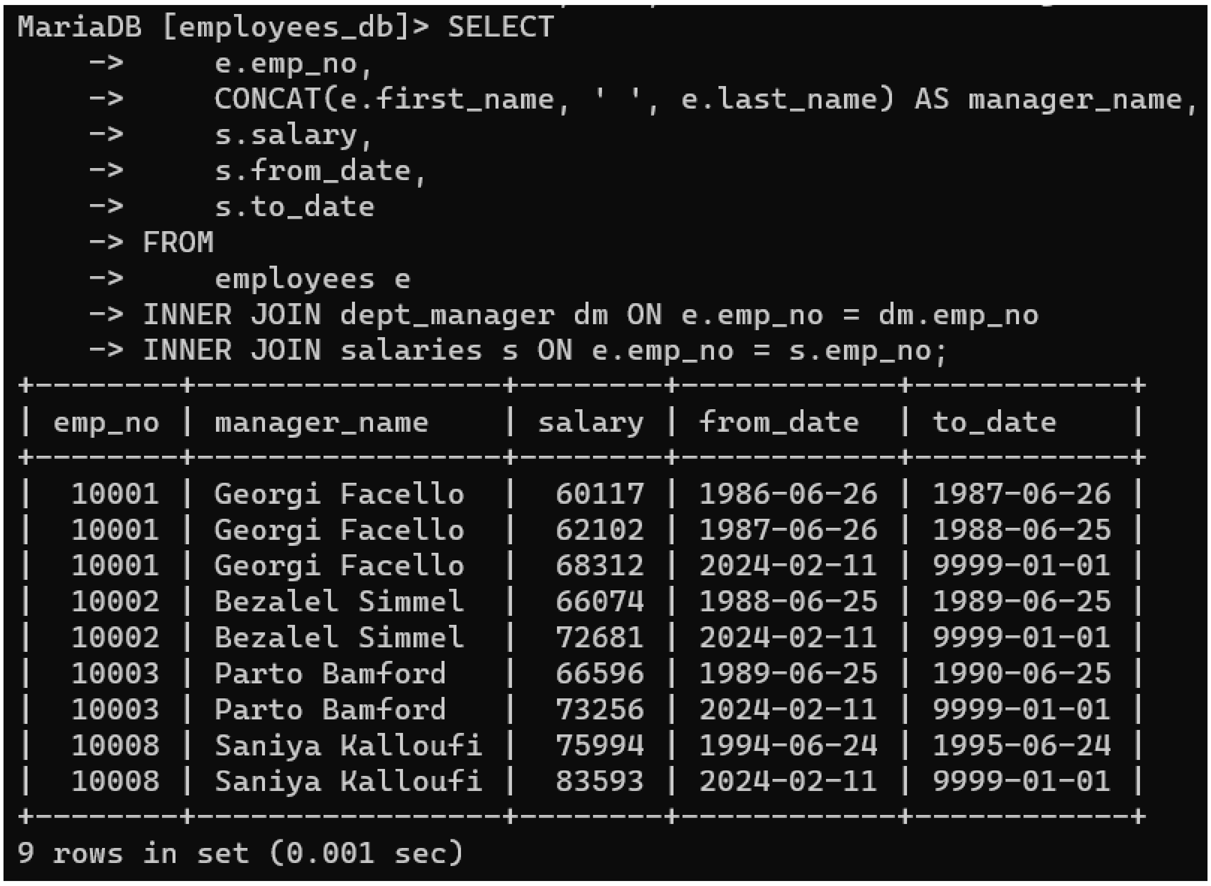
10. Create a SQL statement to increase salaries of all department's managers up to 10% who are working since 1990

Since we do not have the current salary of the current manager, I have decided to select the current manager and their latest known salary to increase it by 10% if they don't have a current one. I then create new records in the salaries table to reflect the increase.



```
MariaDB [employees_db]> INSERT INTO salaries (emp_no, salary, from_date, to_date)
    -> SELECT
    ->     dm.emp_no,
    ->     ROUND(s.salary * 1.1, 2),
    ->     CURDATE(),
    ->     '9999-01-01'
    -> FROM
    ->     dept_manager dm
    -> INNER JOIN
    ->     salaries s ON dm.emp_no = s.emp_no
    -> WHERE
    ->     s.to_date = '9999-01-01' OR
    ->     s.to_date = (
    ->         SELECT MAX(to_date)
    ->         FROM salaries
    ->         WHERE emp_no = dm.emp_no
    ->     );
Query OK, 4 rows affected (0.005 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

Quick query to see the changes:



```
MariaDB [employees_db]> SELECT
    ->     e.emp_no,
    ->     CONCAT(e.first_name, ' ', e.last_name) AS manager_name,
    ->     s.salary,
    ->     s.from_date,
    ->     s.to_date
    -> FROM
    ->     employees e
    -> INNER JOIN dept_manager dm ON e.emp_no = dm.emp_no
    -> INNER JOIN salaries s ON e.emp_no = s.emp_no;
+--------+-----------------+--------+------------+------------+
| emp_no | manager_name    | salary | from_date  | to_date    |
+--------+-----------------+--------+------------+------------+
|  10001 | Georgi Facello  |  60117 | 1986-06-26 | 1987-06-26 |
|  10001 | Georgi Facello  |  62102 | 1987-06-26 | 1988-06-25 |
|  10001 | Georgi Facello  |  68312 | 2024-02-11 | 9999-01-01 |
|  10002 | Bezalel Simmel  |  66074 | 1988-06-25 | 1989-06-25 |
|  10002 | Bezalel Simmel  |  72681 | 2024-02-11 | 9999-01-01 |
|  10003 | Parto Bamford   |  66596 | 1989-06-25 | 1990-06-25 |
|  10003 | Parto Bamford   |  73256 | 2024-02-11 | 9999-01-01 |
|  10008 | Saniya Kalloufi |  75994 | 1994-06-24 | 1995-06-24 |
|  10008 | Saniya Kalloufi |  83593 | 2024-02-11 | 9999-01-01 |
+--------+-----------------+--------+------------+------------+
9 rows in set (0.001 sec)
```