MySQL Marathon

MySQL Marathon

- Build the employees database again from Module 4
- Idea: Practice your SQL

  - Open the sql script downloaded from the website

Select Statement

- Find out who is the employee with ID 201771

Solution

- Ms. Fumitaka Gammage:

```
SELECT
    first_name, last_name, gender
FROM
    employees
WHERE
    emp_no = 201771;
```

Select Statement

- Find out all data from the departments table

Solution

```
SELECT
    *
FROM
    departments;
```

```
'd009','Customer Service'
'd005','Development'
'd002','Finance'
'd003','Human Resources'
'd001','Marketing'
'd004','Production'
'd006','Quality Management'
'd008','Research'
'd007','Sales'
```

Select Statement

- Find all employee's last name whose first name is 'Elvis'

Solution

```
SELECT
    first_name, last_name
FROM
    employees
WHERE
    first_name = 'Elvis';
```

returns 246 rows

Select Statement

- Find the name of all employees with a salary of more than $100,000 currently
  - Currently: to_date in salaries is 9999-01-01

- Two solutions, first without join, then with join

Solution

```
SELECT
    first_name, last_name, salary, from_date, to_date
FROM
    employees,
    salaries
WHERE
    employees.emp_no = salaries.emp_no
        AND salaries.salary > 100000
        AND salaries.to_date = '9999-01-01';
```

- We do not need to preface the attributes with the table name if they are unambiguous
- Notice how we pile on conditions
- Since we are accessing two tables, we can also use a join

Solution

```
SELECT
    first_name, last_name, salary, from_date, to_date
FROM
    employees
        JOIN
    salaries ON employees.emp_no = salaries.emp_no
WHERE
    salaries.salary > 100000
        AND salaries.to_date = '9999-01-01';
```

Select Statement

- Use the in-clause to find all employees with first name Denis or Elvis

Solution

```
SELECT
    first_name, last_name, gender
FROM
    employees
WHERE
    first_name in ('Denis', 'Elvis');
```

Select Statement

- Find how many employees were hired in 1998
  - Remember that LIKE allows similarity queries
    - Wild cards:
    - % — any sequence of characters
    - _ (underscore) a single character
    - * (asterisk) anything
  - Remember count

Solution

```
SELECT
    COUNT(*)
FROM
    employees
WHERE
   hire_date LIKE ('1998%');
```

Select Statement

- Use 'BETWEEN' — 'AND' to find the first, last name, salary, and timeframe of the salalry of all employees that made at one point between 40,000 and 50,000 dollars in salary

Solution

```
SELECT
   first_name, last_name, salary, from_date, to_date
FROM
   employees, salaries
WHERE
   employees.emp_no = salaries.emp_no
    AND salary BETWEEN 40000 AND 50000;
```

```
SELECT
    first_name, last_name, salary, from_date, to_date
FROM
    employees
        JOIN
    salaries ON employees.emp_no = salaries.emp_no
WHERE
    employees.emp_no = salaries.emp_no
```

Select Statement

- Use BETWEEN — AND to find all employees with first name that were hired between March 1 1988 and June 1 1988 and whose first name is 'Gunilla'

Solution

```
SELECT
   first_name, last_name, hire_date
FROM
   employees
WHERE
   hire_date BETWEEN '1986-3-1' AND '1986-6-1'
    AND first_name = 'Gunilla';
```

Select Statement

- What is the average life-time salary of people hired in 1985 compared to people hired in 1989
  - Hint: you need to calculate two averages. First the average for a single employees

Solution

```
SELECT AVG(Averages.indAvg)
FROM   (
SELECT
    AVG(salary) AS indAvg
FROM
    employees,
    salaries
WHERE
    salaries.emp_no = employees.emp_no
        AND employees.hire_date BETWEEN '1989-1-1' AND '1990-1-1'
GROUP BY employees.emp_no) as Averages;
```

Solution

- Here we have an inner table that we need to give its own alias.
  - You notice that you forget it because the workbench gives you an error.

Select Statement

- WHERE versus HAVING
  - Example:
    - Extract all first names from the employees table that appear more than three times, but only for employees hired after January 1 1999.
  - The second condition is the selection of records, so that is a WHERE clause
  - The first condition (count) is a condition after the grouping, so it is a HAVING clause

Select Statement

```
SELECT
    first_name, COUNT(first_name)
FROM
    employees
WHERE
    hire_date > '1999-01-01'
GROUP BY first_name
HAVING COUNT(first_name) > 3
ORDER BY COUNT(first_name) DESC;
```

Select Statement

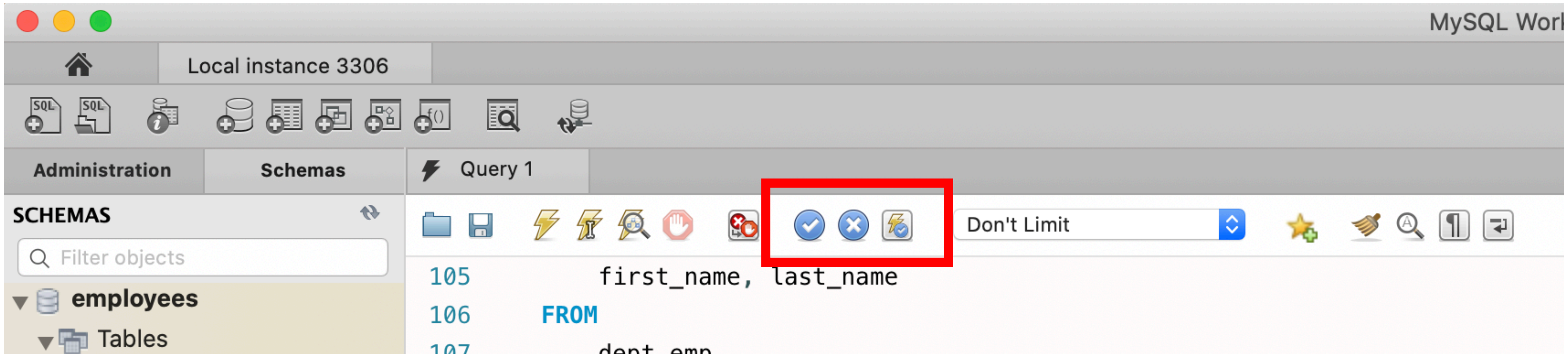
- Problem:
  - Select the number of employees who have had more than one contract after 2000
    - Use the dept_emp

Solution

```
SELECT
    first_name, last_name
FROM
    dept_emp,
    employees
WHERE
    dept_emp.emp_no = employees.emp_no
    AND employees.hire_date > '2000-01-01'
GROUP BY employees.emp_no
HAVING COUNT(dept_emp.from_date) > 1;
```

Update Statements

- Remember commits and rollbacks
- Go to the SQL workbench and disable automatic commit
  - Click on the right button to enable the commit and rollback buttons



```
        first_name, last_name
    FROM
        dept_emp
```

Update Statements

- Change the record with employee number 99901 to your data
- Before and after, check the values, then rollback and check the values for employee number 99901 again

Solution

```
UPDATE
    employees
SET
   first_name = 'Thomas',
    last_name = 'Schwarz',
    birth_date = '2000-12-15'
WHERE
    emp_no = 99901;

SELECT
    *
FROM
    employees
WHERE
    emp_no = 99901;
```

Update Statements

- Change the to_date to April 1, 2020 for all open-ended contracts (ending at 9999-01-01).

Solution

```
UPDATE
    salaries
SET
    to_date = '2020-04-01'
WHERE
    to_date = '9999-01-01';
```

Update Statements

- Insert into departments a new department with dept_no d010 and dept_name 'Business Analytics'

Solution

```
INSERT INTO departments
VALUES ('d010', 'Business Analytics');
```

Update Statements

- Change the name of the 'Business Analytics' department into 'Data Analysis'

Solution

```
UPDATE
   departments
SET
   dept_name = 'Data Analysis'
WHERE
   dept_name = 'Business Analytics';
```

```
SELECT
    *
FROM
    departments;
```

Aggregate Statements

- How many departments are there now in departments?

Solution

```
SELECT
    COUNT(*)
FROM
    departments;
```

Aggregate Statements

- Find the average salary of contracts after January 1, 1985
    - Round to the nearest cent

Solution

```
SELECT
    ROUND(AVG(salary),2)
FROM
    salaries
WHERE
    from_date > '1985-01-01';
```

JOIN Statements

- Find the first name, last name, gender, and department name together with their tenure of all department managers using a JOIN statement

Solution

```
SELECT
    first_name, last_name, gender, dept_name, from_date, to_date
FROM
    employees
        JOIN
    (dept_manager JOIN departments ON dept_manager.dept_no =
departments.dept_no)
    ON dept_manager.emp_no = employees.emp_no;
```

JOIN Statements

- Join the employees and the department manager table in order to find out whether any one with last name 'Markovitch'
- Order the results first by deptartment number, then by the employee number

Solution

```
SELECT
    e.first_name, e.last_name, m.dept_no, m.from_date
FROM
    employees e
        JOIN
    dept_manager m ON e.emp_no = m.emp_no
WHERE
    e.last_name = 'Markovitch'
ORDER BY m.dept_no DESC , e.emp_no;
```

JOIN Statements

- Remember that we introduced another department
  - It has no manager, so it is not represented in the manager table.
  - Create a table that shows all manager emp_no and their departments, but including the new department

Solution

```
SELECT
*
FROM
   departments d
    LEFT JOIN
    dept_manager m ON d.dept_no = m.dept_no
ORDER BY d.dept_no;
```

JOIN Statements

- Select the first name, last name, the hire date, and the job title of all employees whose first name is 'Margareta' and whose last name is 'Markovitch'

Solution

```
SELECT
   e.first_name, e.last_name, t.title, t.from_date, t.to_date
FROM
   employees e JOIN titles t ON e.emp_no = t.emp_no
WHERE
   e.first_name = 'Margareta' and e.last_name = 'Markovitch';
```