

Aluno: Caique Salvador Noboa

RA: 1904949

a)

- Query:

```
SELECT title, ROUND(AVG(salary), 2)
FROM employees.salaries
NATURAL JOIN employees.titles
GROUP BY title;
```

- Foi executado 5 vezes, e o tempo médio de execução foi: 5.371 segundos.

- Comando Explain:

```
'1', 'SIMPLE', 'titles', NULL, 'ALL', 'PRIMARY', NULL, NULL, NULL, '442605', '100.00', 'Using
temporary'

'1', 'SIMPLE', 'salaries', NULL, 'eq_ref', 'PRIMARY', 'PRIMARY', '7', 'employees.titles.emp_no,
employees.titles.from_date', '1', '10.00', 'Using where'
```

Na segunda linha, é possível ver que será usado o comando WHERE nos campos from_date e emp_no. Então criaremos um index dessas duas colunas na tabela salaries.

- Criando index:

```
CREATE INDEX no_and_from_date ON employees.salaries(emp_no, from_date);
```

- Foi executado 5 vezes, e o tempo médio de execução foi: 4.393 segundos. Um ganho de 18%. Acredito que o ganho não foi maior pois o maior custo dessa operação é somar todos os salarios, e fazer a média.

b)

- Query:

```
SELECT first_name, last_name, COUNT(dept_no)
FROM employees.dept_emp
NATURAL JOIN employees.employees
GROUP BY emp_no
HAVING COUNT(dept_no)>1;
```

- Foi executado 5 vezes, e o tempo médio de execução foi: 0.208 segundos.

- Comando Explain:

```
'1', 'SIMPLE', 'employees', NULL, 'ALL', 'PRIMARY', NULL, NULL, NULL, '299335', '100.00', 'Using
temporary'
```

```
'1', 'SIMPLE', 'dept_emp', NULL, 'ref', 'PRIMARY,dept_no', 'PRIMARY', '4',
'employees.employees.emp_no', '1', '100.00', 'Using index'
```

Podemos ver que nesse caso já existe um index, por ser uma foreign key.

c)

- Query:

```
SELECT emp_no, salary, to_date
FROM employees.employees
NATURAL JOIN employees.salaries
WHERE emp_no=10001
ORDER BY to_date DESC;
```

- Foi executado 5 vezes, e o tempo médio de execução foi: 0.000 segundos.

- Comando Explain:

```
'1', 'SIMPLE', 'employees', NULL, 'const', 'PRIMARY', 'PRIMARY', '4', 'const', '1', '100.00', 'Using
index; Using filesort'
```

```
'1', 'SIMPLE', 'salaries', NULL, 'ref', 'PRIMARY', 'PRIMARY', '4', 'const', '17', '100.00', NULL
```

- Create Index:

```
create index no_and_to_date on employees.salaries(emp_no, to_date);
```

- Foi executado 5 vezes, e o tempo médio de execução foi: 0.000 segundos. Ganho de 0%

d)

- Query:

```
SELECT emp_no, salary, MAX(to_date)
FROM employees.employees
NATURAL JOIN employees.salaries
GROUP BY emp_no;
```

- Foi executado 5 vezes, e o tempo médio de execução foi: 0.125 segundos.

- Comando Explain:

```
'1', 'SIMPLE', 'employees', NULL, 'index', 'PRIMARY', 'PRIMARY', '4', NULL, '299335', '100.00',
'Using index'
'1', 'SIMPLE', 'salaries', NULL, 'ref', 'PRIMARY', 'PRIMARY', '4', 'employees.employees.emp_no',
'9', '100.00', NULL
```

Create Index:

```
create index emp_no_index on employees.employees(emp_no);
```

- Foi executado 5 vezes, e o tempo médio de execução foi: 0.068 segundos. Um ganho de 45,6%