E/16/156

H.A.D.T.T. Jayathilaka

Department of Computer Engineering

University of Peradeniya

CO527 Advanced Database Systems

Lab Task :

Refer to the same Company database you created for the previous lab(Lab 01 - Review on SQL) and answer the following questions.

1. **Assuming no indexes are used, record the query execution time for retrieving all the employees by first name in ascending order.**

```
5      -- Question 1
6 •    SELECT*
7      FROM employees
8      ORDER BY first_name ASC ;
9 •    SHOW PROFILES;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Query_ID	Duration	Query
1	0.37752850	SELECT* FROM employees ORDER BY first_name ASC LIMIT 0, 1000
2	0.04331490	SET SESSION profiling = 1
3	0.39781180	SELECT* FROM employees ORDER BY first_name ASC LIMIT 0, 1000

Execution time = 0.3978 sec

2. **Create an index called fname_index on the first_name of the employee table. Retrieve all the employees by first name and record the query execution time. Observe the performance improvement gained when accessing with index.**

```
11      -- Question 2
12 •    CREATE INDEX fname_index ON
13      company.employees(first_name) ;
14
15 •    SELECT*
16      FROM employees
17      ORDER BY first_name ASC ;
18
19 •    SHOW PROFILES;
20
21      -- Question 3
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Query_ID	Duration	Query
6	0.00018660	SELECT DATABASE()
7	0.00014400	SET SESSION profiling = 1
8	0.33089390	SELECT* FROM employees ORDER BY first_name ASC LIMIT 0, 1000
9	2.67470090	CREATE INDEX fname_index ON company.employees(first_name)
10	0.35734420	SELECT* FROM employees ORDER BY first_name ASC LIMIT 0, 1000

Execution time = 0.3573 sec

We can clearly see a considerable difference between execution times without indexing and with indexing. So, using indexing we can increase the performance.

3. **Which indexing technique has been used when creating the above index? Hint: You can use SHOW INDEX FROM [mytable]; to see details of your indexes.**

```
16      -- Question 3
17 •    SHOW INDEX FROM employees;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	C
employees	0	PRIMARY	1	emp_no	A	299523	NULL	NULL		BTREE	
employees	1	fname_index	1	first_name	A	2475	NULL	NULL	YES	BTREE	

Index type is BTREE.

4. **Create a unique index on emp_no, first_name and last_name of employees table. Retrieve all the employees by emp_no, first_name and last_name. Observe if there is any performance improvement with respect to question1. If not, explain any possible reason.**

```
24      -- Question 4
25 •    CREATE UNIQUE INDEX emp_uix ON
26      Company.employees(emp_no,first_name,last_name);
27
28 •    SELECT emp_no,first_name,last_name
29      FROM employees;
30
31 •    SHOW PROFILES;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Query_ID	Duration	Query
9	2.67470090	CREATE INDEX fname_index ON company.employees(first_name)
10	0.35734420	SELECT* FROM employees ORDER BY first_name ASC LIMIT 0, 1000
11	4.44648870	CREATE UNIQUE INDEX emp_uix ON Company.employees(emp_no,first_nam...
12	0.04789940	SELECT emp_no,first_name,last_name FROM employees LIMIT 0, 1000
13	0.00181720	SELECT emp_no,first_name,last_name FROM employees LIMIT 0, 1000
14	0.66126670	SELECT* FROM employees ORDER BY first_name ASC LIMIT 0, 1000
15	0.03509370	SELECT emp_no,first_name,last_name FROM employees LIMIT 0, 1000

There is a considerable time difference in the duration.

5. **Take the following 3 queries.**
   a. **select distinct emp_no from dept_manager where from_date>= '1985-01-01' and dept_no>= 'd005';**
   b. **select distinct emp_no from dept_manager where from_date>= '1996-01-03' and dept_no>= 'd005';**
   c. **select distinct emp_no from dept_manager where from_date>= '1985-01-01' and dept_no<= 'd009';**
      i. **Choose one single simple index(i.e index on one attribute) that is most likely to speed up all 3 queries giving reasons for your selection.**

The index to speed up all 3 queries: dept_no

The attributes related to the where clause can speed up the process. If we compare all three queries, we can see that all three queries have used dept_no in the where clause. So the index which need to speed up all 3 queries is dept_no.

      ii. **For each of the 3 queries, check if MySQL storage engine used that index. If not, give a short explanation why not. You can prefix your select queries with EXPLAIN EXTENDED or with EXPLAIN to display a query execution plan. (Note that in MySQL InnoDB engine uses a clustered index usually on the primary key of the table, by default. We only care about the index you create.)**

```
37 • explain select distinct emp_no from dept_manager where from_date>= "1985-01-01" and dept_no>= "d005";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	dept_manager	range	PRIMARY,deptno_index	PRIMARY	4	NULL	14	Using where; ...

```
39 • explain select distinct emp_no from dept_manager where from_date>= "1996-01-03" and dept_no>= "d005";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	dept_manager	range	PRIMARY,deptno_index	PRIMARY	4	NULL	14	Using where; ...

```
41 • explain select distinct emp_no from dept_manager where from_date>= "1985-01-01" and dept_no<= "d009";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	dept_manager	range	PRIMARY,deptno_index	PRIMARY	4	NULL	25	Using where; ...

MySQL engine has used the primary key of the dept_manager table. That is because the primary key contains the dept_no which has been used as the index.

6. **Consider the queries you wrote for questions 2 - 10 in Lab 01 assignment. Give with short explanations, which attributes on which relations should be used for creating indexes that could speed up your queries.**

Question2:

Index: “last_name” field of the ‘employees’ table

Because that query included the last names and their counts from the table ‘employees’.

Question3:

Index: “title”, “to_date” fields of ‘titles’ table

Because the where clause included about the title and to_date of an employee since title of an employee is the main target of that query.

Question4:

Index: “title”, “to_date” fields of ‘titles’ table

“sex” field of ‘employees’ table

Because that query was used to find the female- department managers who have worked as senior engineers.

Question5:

Index: “salary”, “to_date” fields of ‘salaries’ table

“to_date” field of ‘dept_emp’ table

“title” field of ‘titles’ table

Because that query was used to find the departments and titles of employees who have a salary greater than 115000 and the above index will speed up the query.

Question6:

Index: “hire_date”, “birth_date” fileds of ‘employees’ table

“to_date” field of ‘dept_emp’ table

Because the where clause of that query included about the hire_date, bith_date and to_date of an employee.

Question7:

Index: “dept_name” field of the ‘departments’ table

Because that query used to find the employees who work at the human resources department. Since the most important part of the where clause of that query was dept_name and it speeds up the process.

Question8:

Index: “salary” field of ‘salaries’ table

“dept_name” filed of the ‘departments’ table

Because that query used to find the names of all employees in the database who earn more than every employee in the Finance department. The where clause of the above query included above mentioned fields. So above indexes will speed up the query.

Question9:

Index: “salary”, “to_date” fields of ‘salaries’ table

Because the where clause of the above query compare the salary and to_date of an employee with another employee’s salary and to_date.

Question10:

Index: “salary” field of ‘salaries’ table

“title” field of the table ‘titles’

Because that query computed the difference between the average salary of a Senior Engineer and the average salary of all employees. So having above indexes will speed up the process.

7. **Assume that most of the queries on a relation are insert/update/delete. What will happen to the query execution time if that relation has an index created?**

The query execution time will increase because indexes on such relations will slow down the process**.**

Queries

```
USE company;

SET SESSION profiling = 1;


-- Question 1

SELECT*

FROM employees

ORDER BY first_name ASC ;

SHOW PROFILES;


-- Question 2

CREATE INDEX fname_index ON

company.employees(first_name) ;


SELECT*

FROM employees

ORDER BY first_name ASC ;


SHOW PROFILES;


-- Question 3

SHOW INDEX FROM employees;


-- Question 4
```

```
CREATE UNIQUE INDEX emp_uix ON

Company.employees(emp_no,first_name,last_name);


SELECT emp_no,first_name,last_name

FROM employees;


SHOW PROFILES;

-- Question 5

CREATE INDEX deptno_index ON

Company.dept_manager(dept_no);


explain select distinct emp_no from dept_manager where from_date>= "1985-01-01" and
dept_no>= "d005";


explain select distinct emp_no from dept_manager where from_date>= "1996-01-03" and
dept_no>= "d005";


explain select distinct emp_no from dept_manager where from_date>= "1985-01-01" and
dept_no<= "d009";
```