CO527 – Lab02

E/17/407

Wijesooriya H.D

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

```
SELECT first_name

FROM employees

ORDER BY first_name ASC ;
```

- Time taken = 1.281s

(02) 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.

```
CREATE INDEX fname_index ON

Company.employees(first_name) ;
```

- Time taken = 0.094s
- Performance improvement gain = $\frac{(1.281-0.094)}{1.281}$x100%
  = 92.66%

(03) 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.

Technique : BTREE

(04) 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.

```
CREATE UNIQUE INDEX emp_uix ON
Company.employees(emp_no,first_name,last_name);
```

```
SELECT emp_no,first_name,last_name
FROM employees;
```

- Time taken = 0.078s
- Performance is improved.

(05)

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';

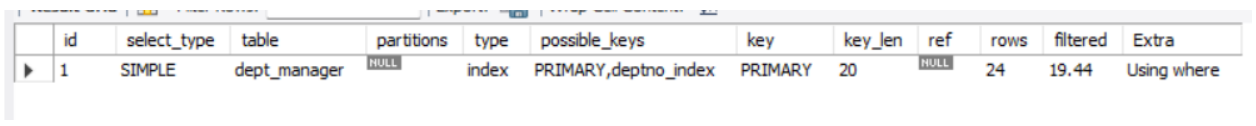
I. Choose one single simple index (i.e index on one attribute) that is most likely to speed up all 3 queries giving reasons for your selection.

CREATE INDEX deptno_index ON

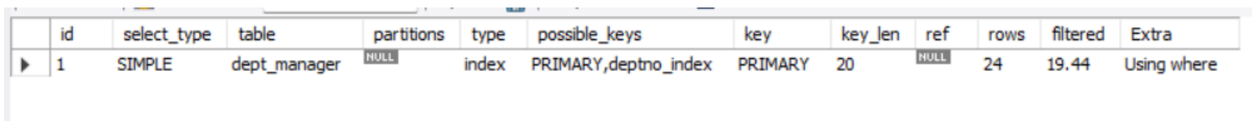
Company.dept_manager(dept_no);

- Reasons for selecting dept_no as an index: all three queries use dept_no in the where clause and to speed up the process we need to choose attributes related to the where clause.

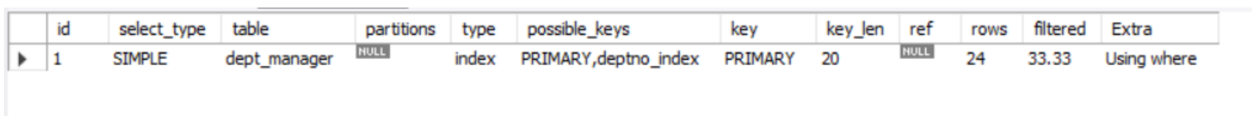
II. For each of the 3 queries, check if MySQL storage engine used that index. If not, give a short explanation why not.



id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	dept_manager	NULL	index	PRIMARY,deptno_index	PRIMARY	20	NULL	24	19.44	Using where



id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	dept_manager	NULL	index	PRIMARY,deptno_index	PRIMARY	20	NULL	24	19.44	Using where



id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	dept_manager	NULL	index	PRIMARY,deptno_index	PRIMARY	20	NULL	24	33.33	Using where

MySQL engine did not use the index that I created.It used the primary key of the dept_manager table. This is because the primary key contains the field that I used to create the index.

* field that I used to create the index = dept_no , primary key =(dept_no+emp_no)

(06) 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.

Q2: We should create an index on the last_name field of the 'employees' table. Because there we get the last names and their counts from the table 'employees'.

Q3: Here we should create an index on title, and to_date fileds of 'titles' table as they speed up the query. Because in the where clause we consider about the title and to_date of an employee. (title of an employee is the main target of that query)

Q4: Here we can create multiple indexes on different tables. They are given below.

- An index on title and to_date fileds of 'titles' table
- An index on sex filed of 'employees' table

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

Q5: We should create multiple indexes on different tables. They are shown below.

- An index on salary and to_date fileds of 'salary' table
- An index on to_date filed of 'dept_emp' table
- An index on title filed of 'titles' table

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

Q6: In here also we need to create multiple indexes.

- An index on hire_date and birth_date fileds of 'employees' table
- An index on to_date filed of 'dept_emp' table

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

Q7: Here we need to create an index on dept_name filed of the 'departments' table. Because the main aim of that query was to find the employees who work at the human resources department. (the most important part of the where clause of that query was dept_name as it speeds up it)

Q8: In that query we need to have multiple indexes.

- An index on salary field of the 'salaries' table
- An index on dept_name field of the 'departments' table

Because in that query we were supposed to find the names of all employees in the database who earn more than every employee in the Finance department and there in the where clause I have included above mentioned fields. So above indexes will speed up the query.

Q9: In this question we need to create an index on salary and to_date fields of 'salaries' table. Because in the where clause we compare the salary and to_date of an employee with another employee's salary and to_date.

Q10: Here we need to create multiple indexes in order to speed up the query execution time.

- An index on salary field of the table 'salaries'
- An index on title field of the table 'titles'

Because in that query I 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.

(07) 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?

Query execution time will increase (having indexes on such relations will slow down the process**).**