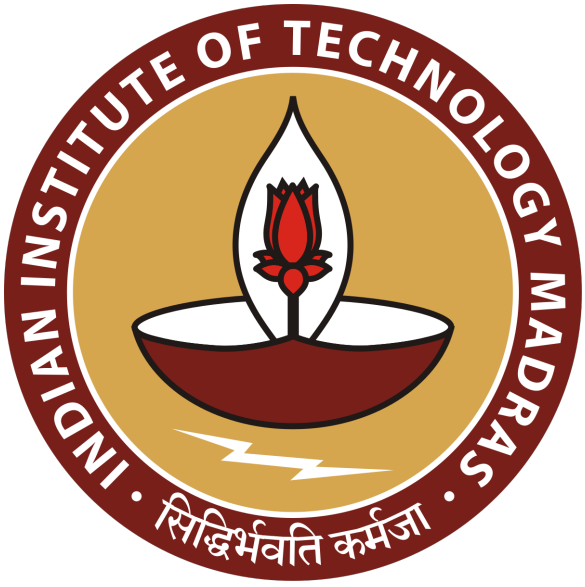
CS3700 - Introduction to Database Systems

Assignment 4B - Index Effect Study



Name - Jay Prajapati
Roll Number - ME21B143
Group No. - 11

Table of Contents

1 SQL Query Description

Give list of all classrooms with the course and professor name who taught in that classroom in even semester, which are offered by Physics or Psychology department which are taught by a professor who joined the institution on or after 1992.

```
SELECT c.cname AS course_name, p.name AS
   professor_name, t.classRoom, t.year
FROM teaching t, course c, professor p, department d
WHERE t.courseId = c.courseId
AND t.empId = p.empId
AND c.deptNo = d.deptId
AND (d.name = 'Physics' OR d.name = 'Psychology')
AND t.sem = "even"
AND p.startYear >= 1992;
```

2 Explain

Now updating the query by adding EXPLAIN command in the query to get the query plan generated by MySQL query serve, and inferening from the output on what index to create.

```
EXPLAIN SELECT c.cname AS course_name, p.name AS
   professor_name, t.classRoom, t.year
FROM teaching t, course c, professor p, department d
WHERE t.courseId = c.courseId
AND t.empId = p.empId
AND c.deptNo = d.deptId
AND (d.name = 'Physics' OR d.name = 'Psychology')
AND t.sem = "even"
AND p.startYear >= 1992;
```

```
1  EXPLAIN SELECT c.cname AS course_name, p.name AS professor_name, t.classRoom, t.year
2  FROM teaching t, course c, professor p, department d
3  WHERE t.courseId = c.courseId
4  AND t.empId = p.empId
5  AND c.deptNo = d.deptId
6  AND (d.name = 'Physics' OR d.name = 'Psychology')
7  AND t.sem = "even"
8  AND p.startYear >= 1992;
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	t	NULL	ALL	PRIMARY,courseId	NULL	NULL	NULL	52	10.00	Using where
1	SIMPLE	p	NULL	eq_ref	PRIMARY	PRIMARY	22	academic_insti.t.empId	1	33.33	Using where
1	SIMPLE	c	NULL	eq_ref	PRIMARY,deptNo	PRIMARY	34	academic_insti.t.courseId	1	100.00	Using where
1	SIMPLE	d	NULL	eq_ref	PRIMARY	PRIMARY	82	academic_insti.c.deptNo	1	19.00	Using where

Figure 1: Without Indexing

3 Index Creation

To optimize the performance of the query, we recommend creating the following indices:

```
CREATE INDEX idx_courseId ON teaching (courseId);
CREATE INDEX idx_empId ON teaching (empId);
CREATE INDEX idx_deptNo ON course (deptNo);
CREATE INDEX idx_name ON department (name);
CREATE INDEX idx_sem ON teaching (sem);
CREATE INDEX idx_startYear ON professor (startYear);
```

```
1  CREATE INDEX idx_courseId ON teaching (courseId);
2  CREATE INDEX idx_empId ON teaching (empId);
3  CREATE INDEX idx_deptNo ON course (deptNo);
4  CREATE INDEX idx_name ON department (name);
5  CREATE INDEX idx_sem ON teaching (sem);
6  CREATE INDEX idx_startYear ON professor (startYear);
7
8  EXPLAIN SELECT c.cname AS course_name, p.name AS professor_name, t.classRoom, t.year
9  FROM teaching t, course c, professor p, department d
10 WHERE t.courseId = c.courseId
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	d	NULL	range	PRIMARY,idx_name	idx_name	83	NULL	2	100.00	Using where; Using index
1	SIMPLE	c	NULL	ref	PRIMARY,idx_deptNo	idx_deptNo	83	academic_insti.d.deptId	10	100.00	NULL
1	SIMPLE	t	NULL	ref	PRIMARY,idx_courseId,idx_empId,idx_sem	idx_courseId	34	academic_insti.c.courseId	1	44.23	Using index condition
1	SIMPLE	p	NULL	eq_ref	PRIMARY,idx_startYear	PRIMARY	22	academic_insti.t.empId	1	88.46	Using where

Figure 2: With Indexing

4 Reason for Index Creation

- **idx_courseId:** This index is created on the `courseId` column in the `teaching` table. It is specifically designed to improve the efficiency of joins involving the `teaching` table. By creating an index on `courseId`, the database can quickly locate matching rows in the `teaching` table when joining with other tables based on the `courseId` column.
- **idx_empId:** Similarly, this index is created on the `empId` column in the `teaching` table to optimize join operations. By indexing `empId`, the database can efficiently perform joins with other tables based on the `empId` column, such as the `professor` table.
- **idx_deptNo:** This index is created on the `deptNo` column in the `course` table to enhance join performance. Indexing `deptNo` allows the database to quickly locate matching rows in the `course` table when joining with other tables based on the department number.
- **idx_name:** Created on the `name` column in the `department` table, this index aims to improve overall query performance. By indexing `name`, the database can efficiently filter or join based on department names, enhancing the speed of queries involving the `department` table.
- **idx_sem:** This index is created on the `sem` column in the `teaching` table to improve filtering and grouping efficiency. Indexing `sem` allows the database to quickly locate rows based on the semester, facilitating faster filtering and grouping operations.
- **idx_startYear:** Created on the `startYear` column in the `professor` table, this index enhances filtering efficiency based on the starting year of professors. By indexing `startYear`, the database can quickly identify rows based on the start year, leading to faster query execution when filtering based on professor start years.

5 Inferences

Total number of rows accessed reduced from 53 to 14:

- Access to department increased from 1 to 2 rows.
- Access to course increased from 1 to 10 rows.
- Access to professor remained unchanged at 1 row.
- Access to teaching reduced from 52 to 1 row.

6 Conclusion

Overall, the indexing has significantly improved query performance by reducing the number of rows accessed, particularly evident in the `teaching` table. While there's a slight increase in the number of rows accessed in the `department` and `course` tables, the overall query execution is much more efficient due to the substantial reduction in `teaching` table access.