CS3700 - INTRODUCTION TO DATABASE SYSTEMS

Assignment - 4 - Report

CH18B035 - A Gunavardhan Reddy

Query : Find roll numbers, sex and department names of all students named ‘Zander’ who are enrolled in any course thought by professor ‘Mingoz’ between year 2001 and 2005 along with course name.

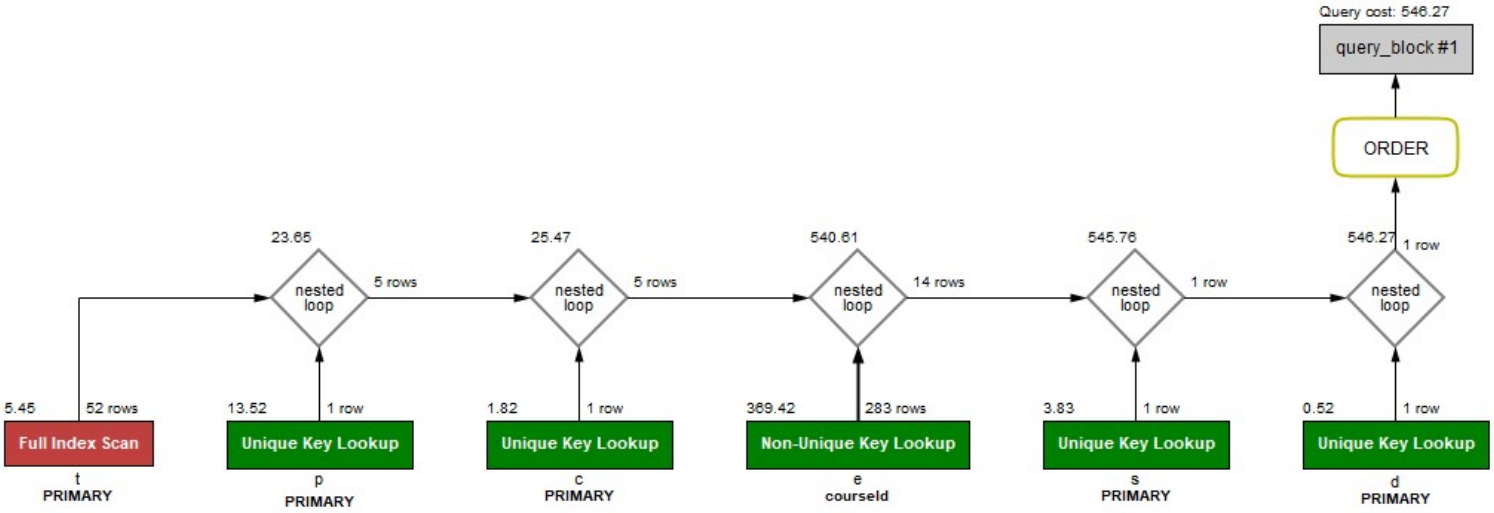
Explanation: Query is straight forward, finding the details of the students named ‘Zander’ enrolled in a course taught by ‘Mingoz’ in between years 2001 and 2005, but this needs combination many different tables.

Tables used: student, department, professor, course, enrollment, teaching

SQL: formulation of query :

```
select s.rollNo, s.name, s.sex, d.name, e.courseId, c.cname
from student as s join enrollment as e on e.rollNo = s.rollNo, department as d, professor as p, teaching
                                                    as t join course as c on c.courseId = t.courseId
where s.name = 'Zander' and p.name = 'Mingoz' and s.deptNo = d.deptId and t.courseId = e.courseId
                                                and e.sem = t.sem and e.year = t.year and p.empId = t.empId
                                                    and (e.year between 2001 and 2005) order by t.empId;
```

The output of explain command(before indexing):



Timing (as measured by the server):
Execution time: 0:00:0.00593090
Table lock wait time: 0:00:0.00017700

Errors:
Had Errors: NO
Warnings: 0

Rows Processed:
Rows affected: 0
Rows sent to client: 1
Rows examined: 3230

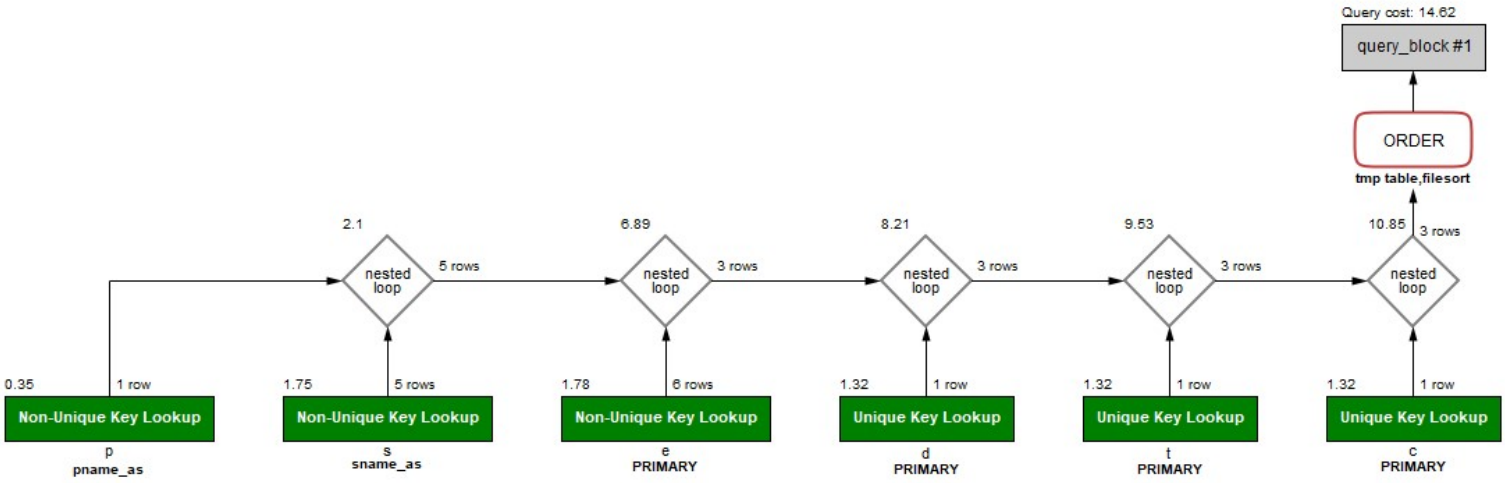
After creating two indices: on names of student and professor table

Why? The two fields which we selected for creating indices are not primary keys and a query on it requires entire table scan, but table scan on both tables requires lot of time and is very costly, hence we reduce the time in log times by indexing on the names(binary search takes place)

This improvement can be seen in results

```
create index sname_as on student(name asc);
create index pname_as on professor(name asc);
```

The output of explain command(after indexing):



Timing (as measured by the server):
Execution time: 0:00:0.00067620
Table lock wait time: 0:00:0.00013600

Errors:
Had Errors: NO
Warnings: 0

Rows Processed:
Rows affected: 0
Rows sent to client: 1
Rows examined: 85

Results	Before Indexing	After Indexing	Improvement
Total_Query_Cost	546.27	14.62	**97.32%**
Execution time	0.00593090	0.00067620	**88.59%**
Rows examined	3230	85	**97.36%**

As the main issue is table scan of names to find 'Zander' and 'Mingoz' when indexed it shows almost 90% enhancement in query evaluation. Also because of which rows examined are less as binary search skips half of the search space at a time. Hence saving the time and cost.