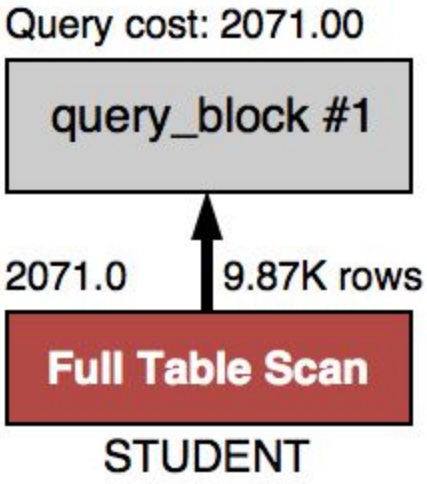
CSCI 4370 Project 4 Report for MySQL Queries
Team: Bits Please

Author of queries 1 - 3: Edvin Dizdarevic

Query 1 Before: List the name of the student with id equal to v1 (id).

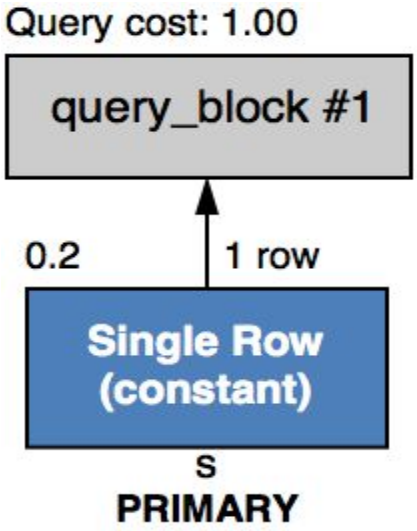
```
SELECT sname
FROM STUDENT
WHERE id = '545899';
```



Time to run: 0.004 seconds

Query 1 Optimized: List the name of the student with id equal to v1 (id).

```
SELECT s.sname
FROM STUDENT AS s
WHERE s.id = '545899';
```

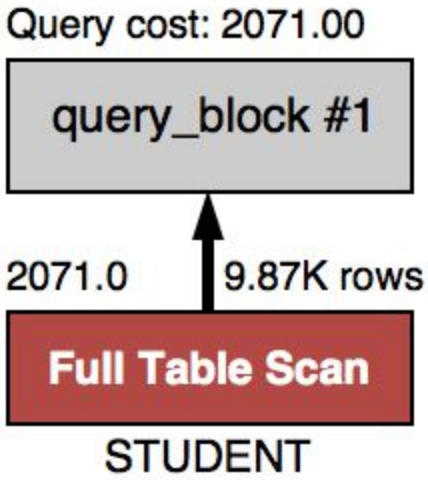


Time to run: 0.00044 seconds

Observations from before to after optimization: The query cost jumps from 2071 in the unoptimized to only 1 in the optimized query, making it run significantly faster and traversed only a single row rather than 9.87k rows. Also, we go from a FULL TABLE SCAN to a constant SINGLE ROW SCAN.

Query 2 Before: List the names of students with id in the range of v2 (id) to v3 (inclusive).

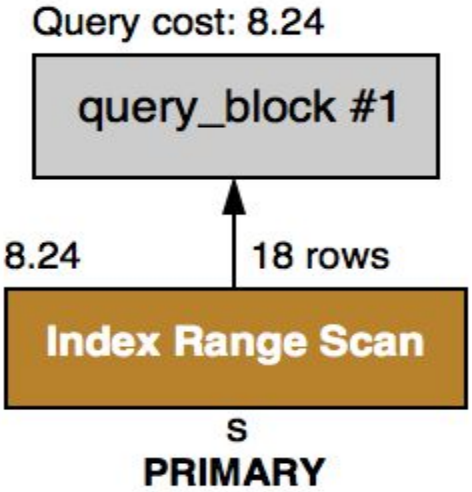
```
SELECT sname
FROM STUDENT
WHERE id BETWEEN '15192' AND '17138';
```



Time to run: 0.0043 seconds

Query 2 Optimized: List the names of students with id in the range of v2 (id) to v3 (inclusive).

```
SELECT s.sname
FROM STUDENT AS s
WHERE s.id BETWEEN '15192' AND '17138';
```

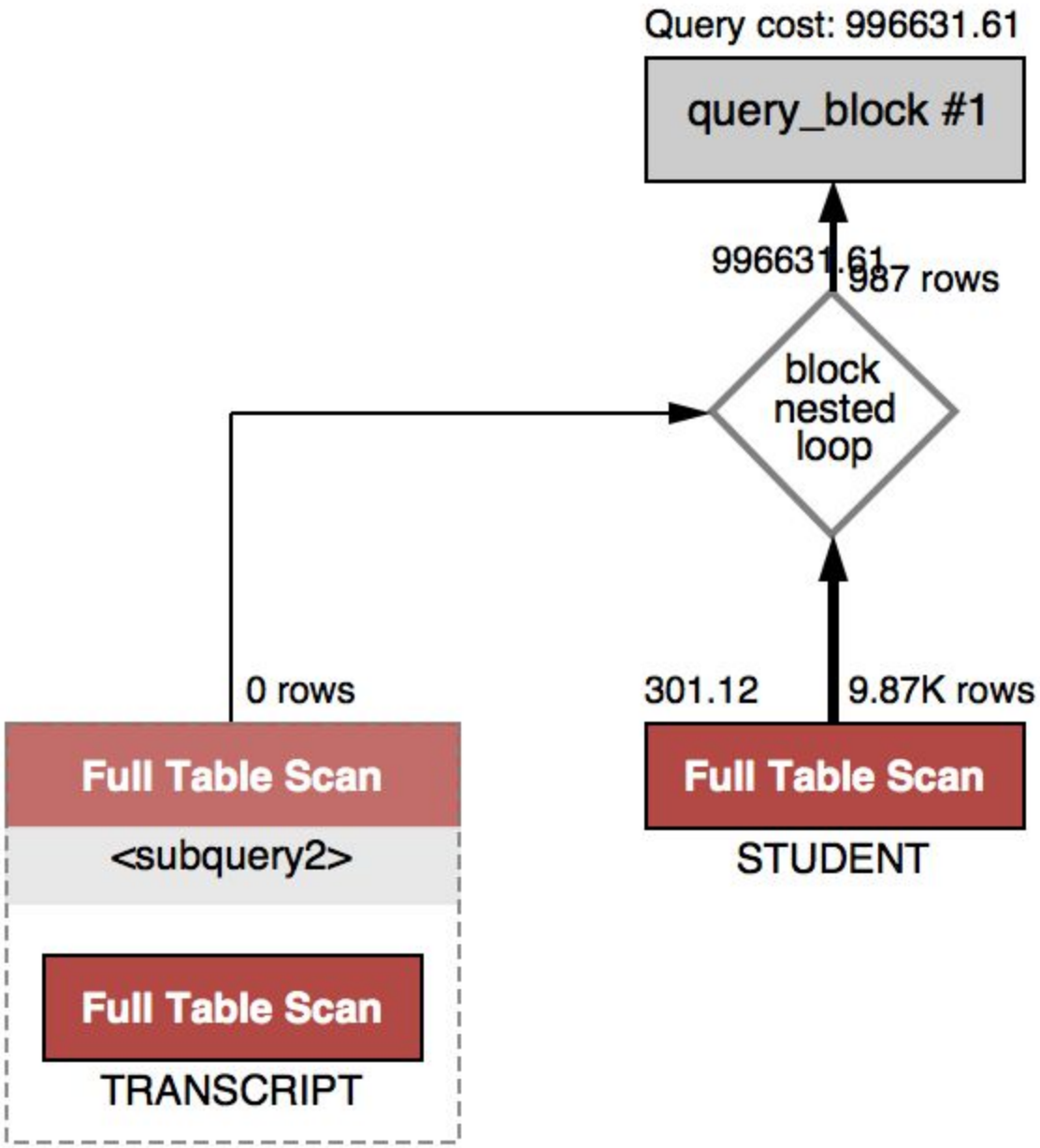


Time to run: 0.00034 seconds

Observations from before to after optimization: The query cost jumps from 2071 in the unoptimized to 8.24 in the optimized query, making it run significantly faster and traversed only a 18 rows rather than 9.87k rows. Also, we go from a FULL TABLE SCAN to an INDEX RANGE SCAN.

Query 3 Before: List the names of students who have taken course v4 (crsCode).

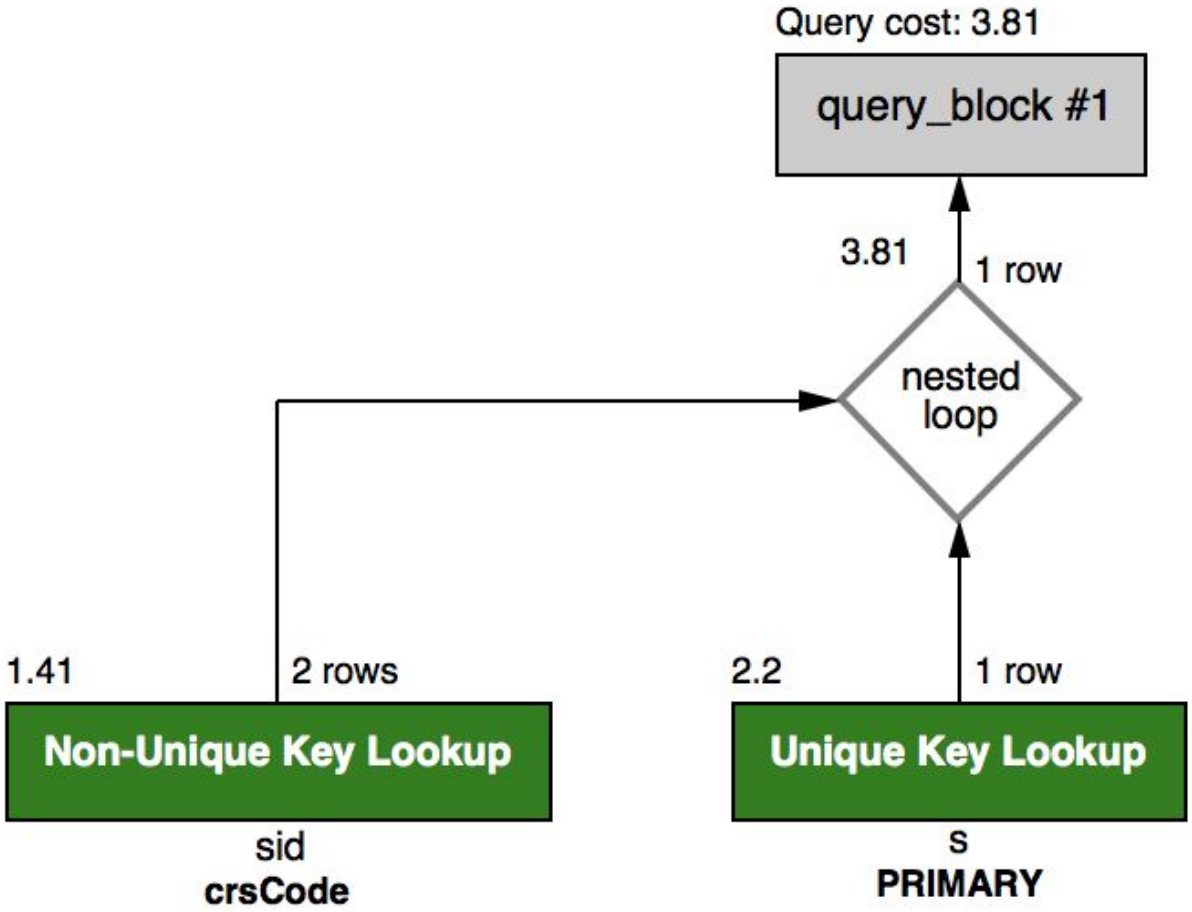
```
SELECT sname
FROM STUDENT
WHERE id = ANY (SELECT studId
        FROM TRANSCRIPT
        WHERE crsCode = 'crsCode902901');
```



Time to run: 0.0072 seconds

Query 3 Optimized: List the names of students who have taken course v4 (crsCode).

```
SELECT s.sname
FROM STUDENT AS s
WHERE s.id = ANY (SELECT sid.studId
        FROM TRANSCRIPT AS sid
        WHERE sid.crsCode = 'crsCode902901');
```



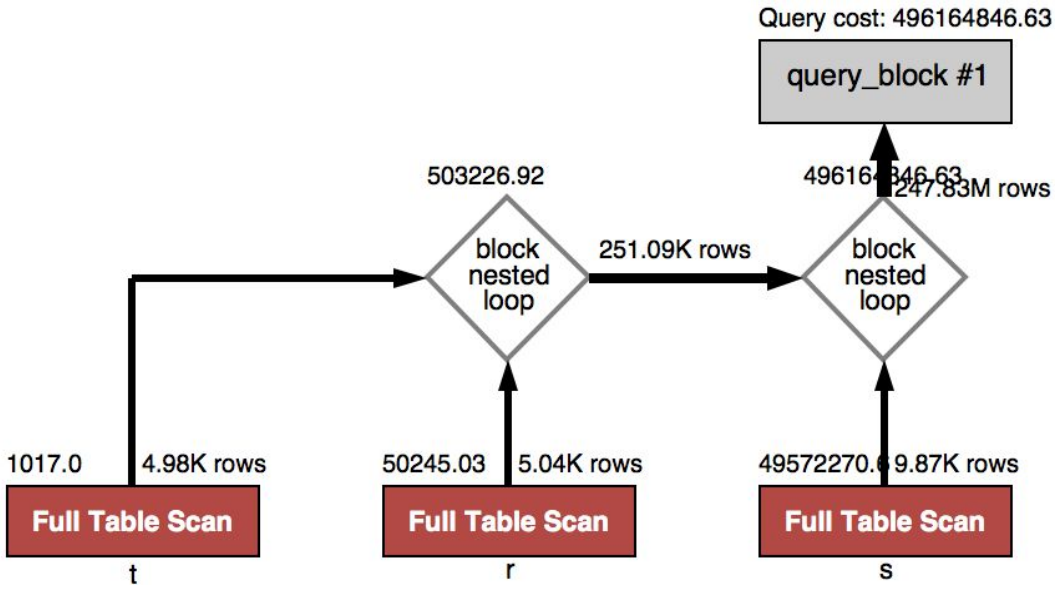
Time to run: 0.00048 seconds

Observations from before to after optimization: The query cost jumps from 996631.61 in the unoptimized to only 3.81 in the optimized query, making it run significantly faster and traversed only a 4 rows total rather than 10.857k rows total. Also, we go from a FULL TABLE SCAN to a UNIQUE and NON-UNIQUE KEY LOOKUPS.

Author of queries 4 - 6: Jonathan Waring

Query 4 Before: List the names of students who have taken a course taught by professor v5 (491584)

```
SELECT s.sname
FROM STUDENT AS s, TRANSCRIPT AS r, TEACHING AS t
WHERE t.profId=491584 AND r.crsCode=t.crsCode AND r.studId=s.id;
```

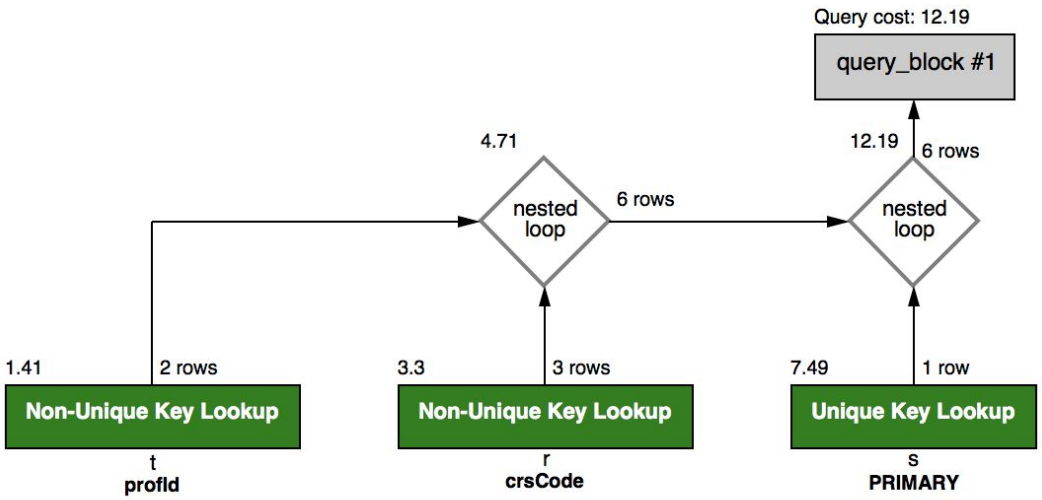


Time to run: 0.031 seconds

Observations: Cross product and no indexing leads to full table scans searching over a huge amount of rows. Query cost far exceeds cost of optimized query.

Query 4 Optimized: List the names of students who have taken a course taught by professor v5 (name)

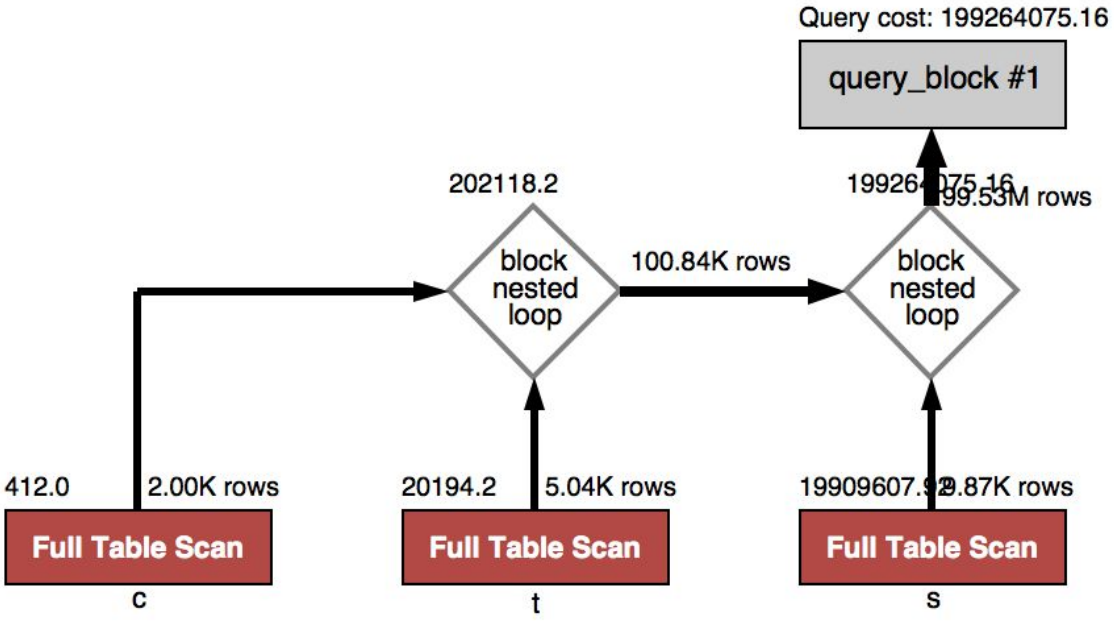
```
SELECT s.sname
FROM STUDENT AS s INNER JOIN TRANSCRIPT AS r ON r.studId=s.id INNER JOIN TEACHING AS t
ON r.crsCode=t.crsCode
WHERE t.profId=491584;
```



Time to run: 0.000 seconds
Observations: Indexing and inner joins over cross products, leads to less rows being searched and a much lower query cost.

Query 5 Before: List the names of students who have taken a course from department v6 (deptId), but not v7

```
SELECT s.sname
FROM STUDENT AS s, TRANSCRIPT AS t, COURSE AS c
WHERE t.crsCode=c.crsCode AND c.deptId='deptId664077' AND t.studId=s.id AND c.deptId!='deptId424969';
```

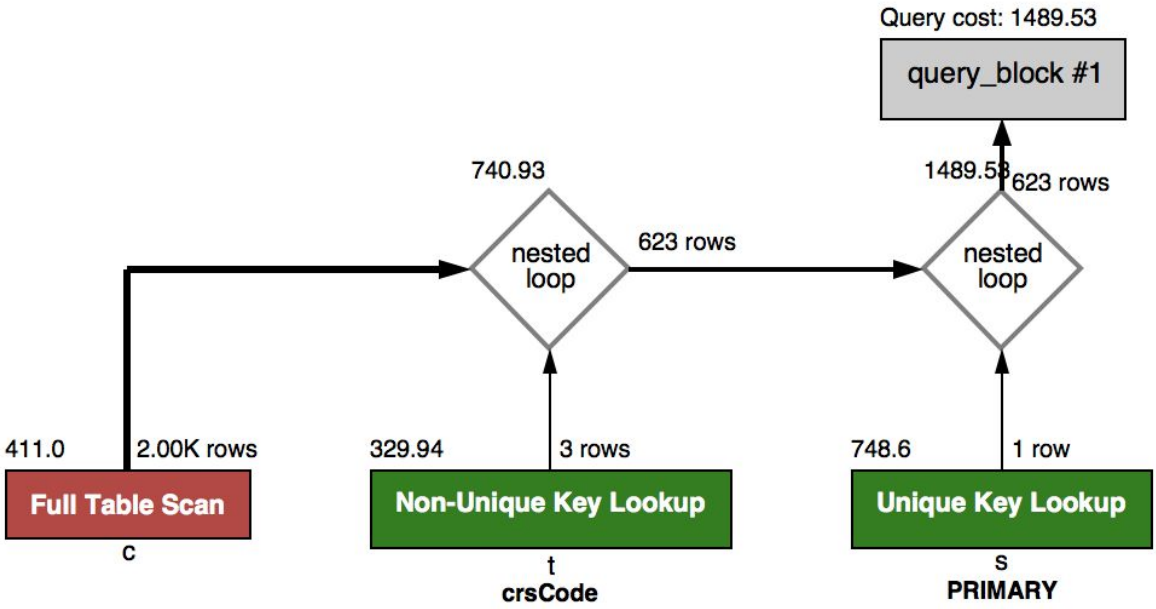


Time to run: 0.016 seconds

Observations: Cross product and no indexing leads to full table scans searching over a huge amount of rows. Query cost far exceeds cost of optimized query.

Query 5 Optimized: List the names of students who have taken a course from department v6 (deptId), but not v7

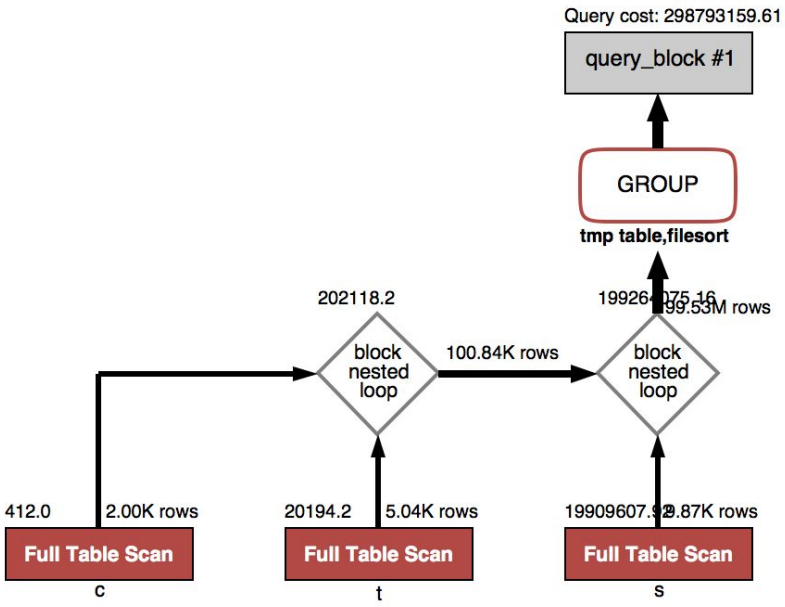
```
SELECT s.sname
FROM STUDENT AS s INNER JOIN TRANSCRIPT AS t ON s.id=t.studId INNER JOIN COURSE AS c ON t.crsCode=c.crsCode
WHERE c.deptId='deptId664077' AND c.deptId!='deptId424969';
```



Time to run: 0.000 seconds
Observations: Indexing and inner joins over cross products, leads to less rows being searched and a much lower query cost.

Query 6 Before: List the names of students who have taken all courses offered by department v8 (deptId)

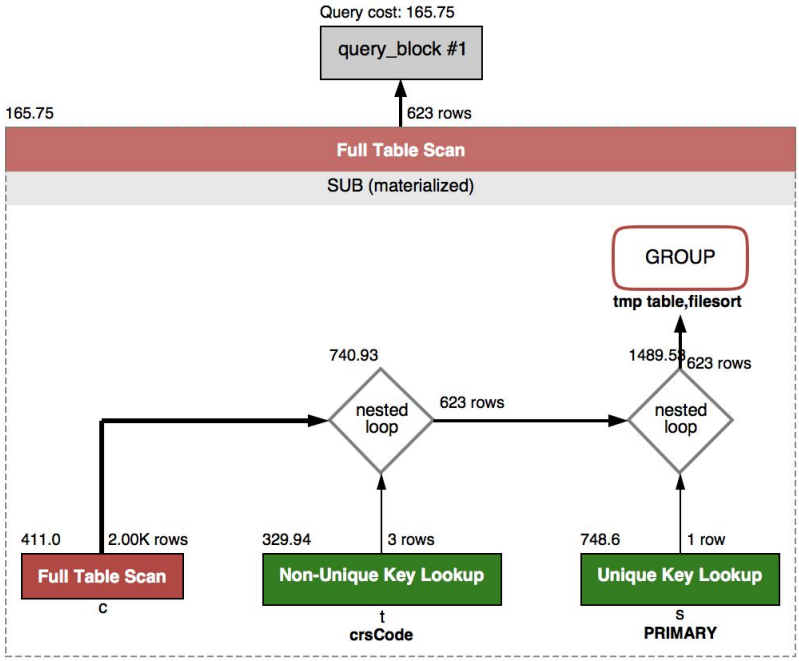
```
SELECT s.sname
FROM STUDENT AS s, COURSE AS c, TRANSCRIPT AS t
WHERE t.crsCode=c.crsCode AND t.studId=s.id AND c.deptId='deptId424969'
GROUP BY s.sname
HAVING COUNT(*) = (SELECT COUNT(*) FROM COURSE WHERE COURSE.deptId='deptId424969');
```



Time to run: 2.172 seconds
Observations: Cross product and no indexing leads to full table scans searching over a huge amount of rows. Query cost far exceeds cost of optimized query.

Query 6 Optimized: List the names of students who have taken all courses offered by department v8 (deptId)

```
SELECT s.sname
FROM STUDENT AS s
WHERE(SELECT COUNT(c.crsCode) AS cnt
                FROM STUDENT AS s INNER JOIN TRANSCRIPT AS t ON t.studId=s.id INNER JOIN
                COURSE AS c ON t.crsCode=c.crsCode
                WHERE c.deptId = 'deptId424969'
                GROUP BY s.sname
                HAVING cnt = (SELECT COUNT(crsCode) FROM COURSE WHERE deptId =
                'deptId424969')
        )
```



Time to run: 0.016 seconds

Observations: Indexing and inner joins over cross products, leads to less rows being searched and a much lower query cost. Searching over a materialized subquery also improved speed.