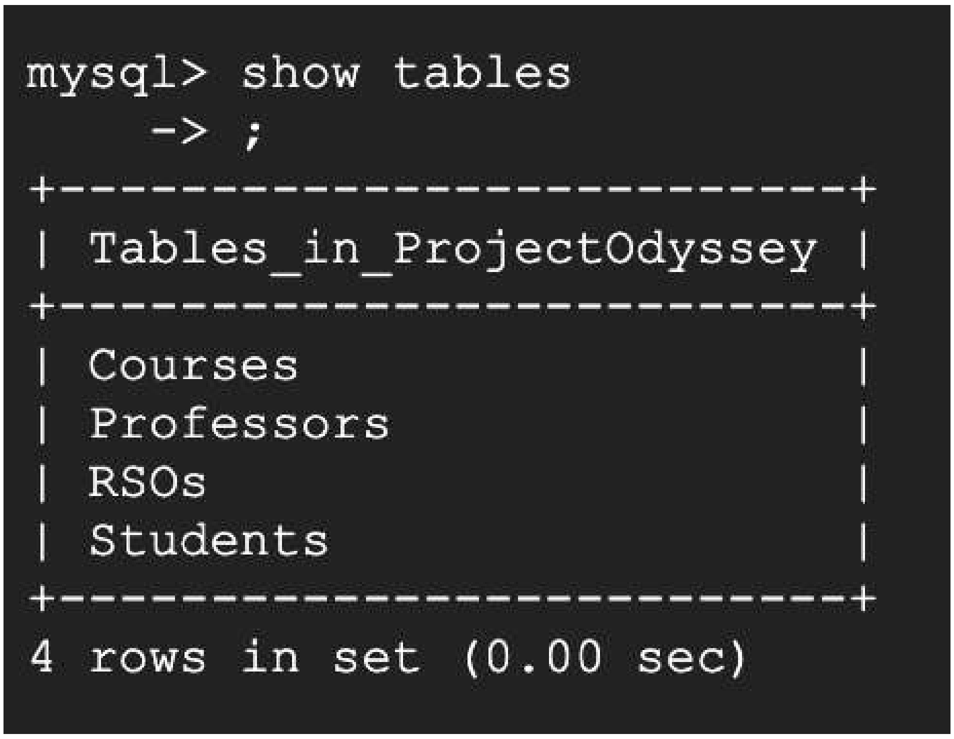
DDL Commands

```
CREATE TABLE Students (
        email VARCHAR (255),
        FirstName VARCHAR (255),
        LastName VARCHAR (255),
        Department VARCHAR (255),
        Year INT,
        CoursesTaken VARCHAR(255),
        Skills VARCHAR (255),
        RSOs VARCHAR (255),
        Interests VARCHAR (255)
);

CREATE TABLE Courses (
        CRN VARCHAR (255) [primary key],
        Course Name VARCHAR (255),
        department VARCHAR (255),
        Instructor Email VARCHAR (255) [FK to Professors.emal],
        Course Area VARCHAR (255)
);


CREATE TABLE Skills (
        email VARCHAR (255) [primary key]
        Programming Lang VARCHAR (255)
        tools VARCHAR (255)
        libraries VARCHAR (255)
);

CREATE TABLE RSOs (
     RSO Name VARCHAR (255) [primary key]
     RSO Area VARCHAR (255) [FK to Research Group.Research Area]
);

CREATE TABLE Research Group (
                Research Group Website VARCHAR (255) [primary key]
                Professor Name VARCHAR (255)
                Department VARCHAR (255) [FK to Course.department]
                Research Area VARCHAR (255)
                Research Topic VARCHAR (255)
);
```

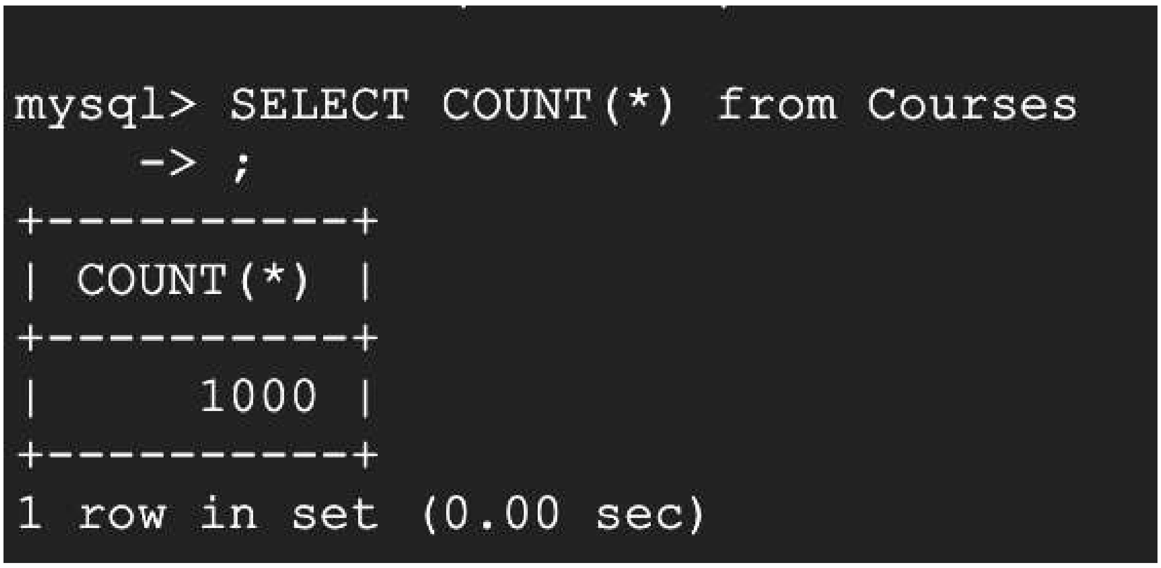
```
CREATE TABLE Professors (
        email VARCHAR (255) [primary key],
        FirstName VARCHAR (255),
        LastName VARCHAR (255),
        ResearchGroupWebsite VARCHAR(255) [FK to
ResearchGroup.ResearchGroupWebsite]
        Department VARCHAR (255),
        RecentCourseTaught VARCHAR (255),
        ResearchArea VARCHAR (255),
        ResearchTopic VARCHAR (255)
);

CREATE TABLE(
        Description TEXT,
);
```

<u>Database Tables on GCP</u>

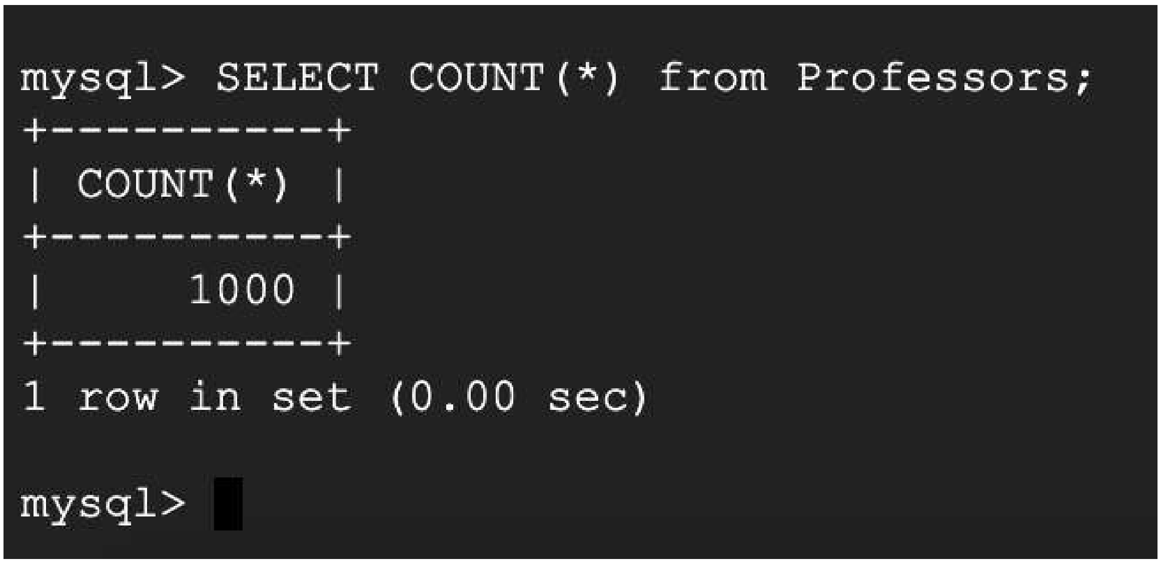


```
mysql> show tables
    -> ;
+--------------------------+
| Tables_in_ProjectOdyssey |
+--------------------------+
| Courses                  |
| Professors               |
| RSOs                     |
| Students                 |
+--------------------------+
4 rows in set (0.00 sec)
```

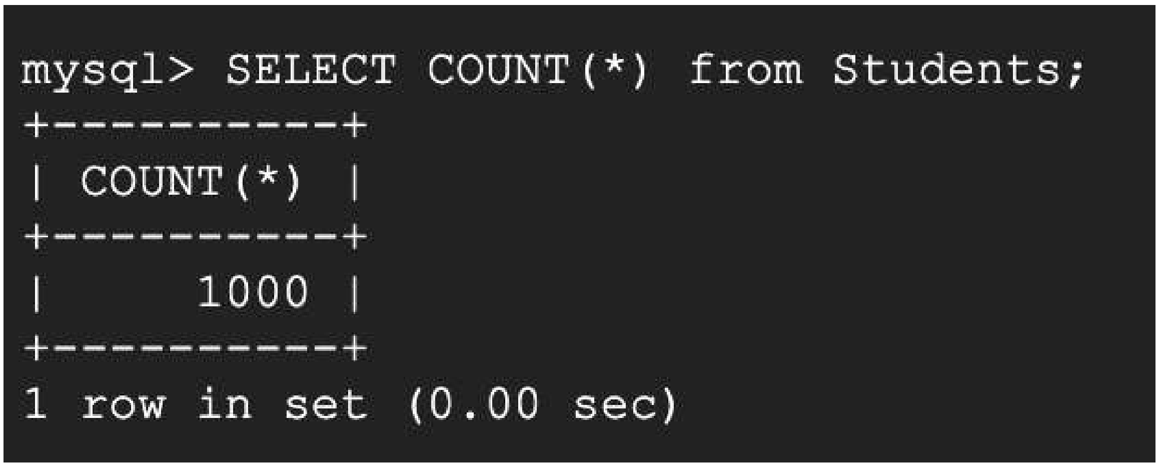
<u>Tables with atleast 1000 rows</u>



```
mysql> SELECT COUNT(*) from Courses
    -> ;
+----------+
| COUNT(*) |
+----------+
|     1000 |
+----------+
1 row in set (0.00 sec)
```

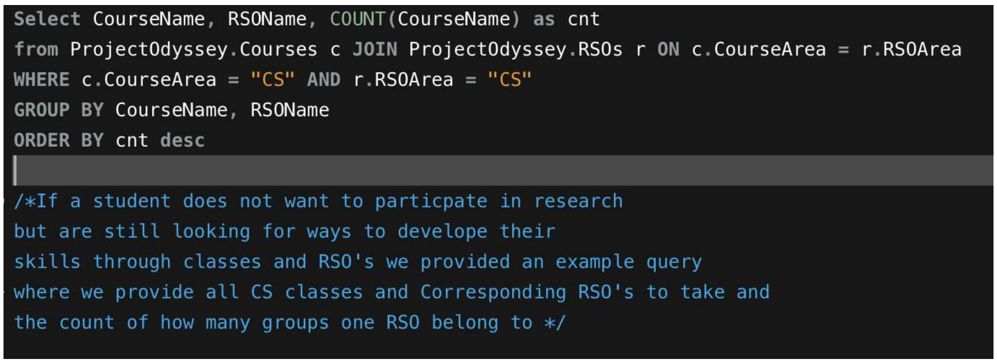


```
mysql> SELECT COUNT(*) from Professors;
+----------+
| COUNT(*) |
+----------+
|     1000 |
+----------+
1 row in set (0.00 sec)

mysql> 
```

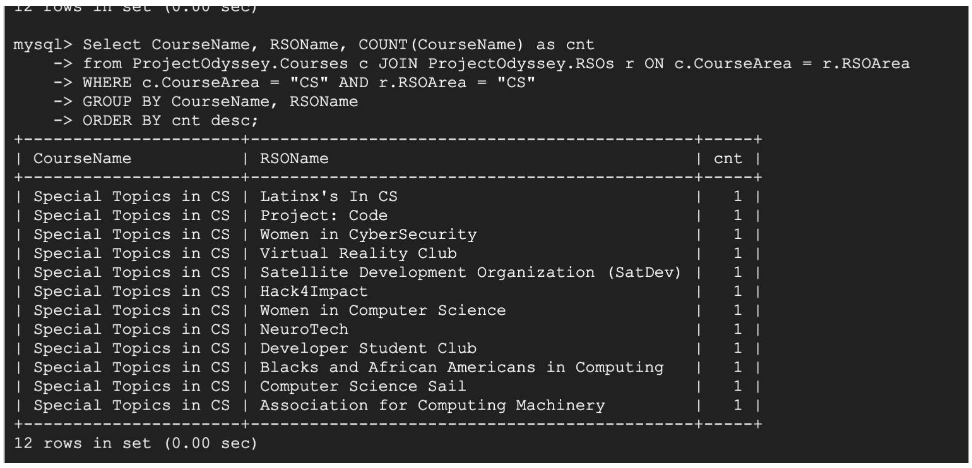


```
mysql> SELECT COUNT(*) from Students;
+----------+
| COUNT(*) |
+----------+
|     1000 |
+----------+
1 row in set (0.00 sec)
```

Query 1

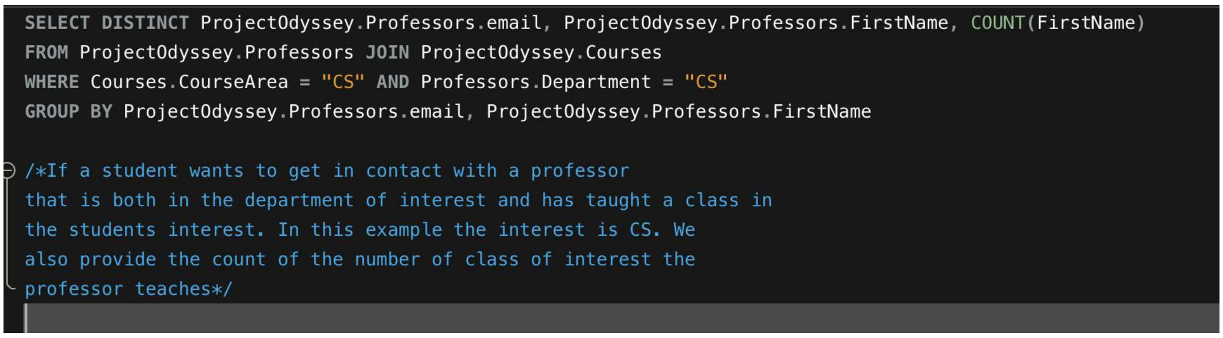


```
Select CourseName, RSOName, COUNT(CourseName) as cnt
from ProjectOdyssey.Courses c JOIN ProjectOdyssey.RSOs r ON c.CourseArea = r.RSOArea
WHERE c.CourseArea = "CS" AND r.RSOArea = "CS"
GROUP BY CourseName, RSOName
ORDER BY cnt desc

/*If a student does not want to particpate in research
but are still looking for ways to develope their
skills through classes and RSO's we provided an example query
where we provide all CS classes and Corresponding RSO's to take and
the count of how many groups one RSO belong to */
```

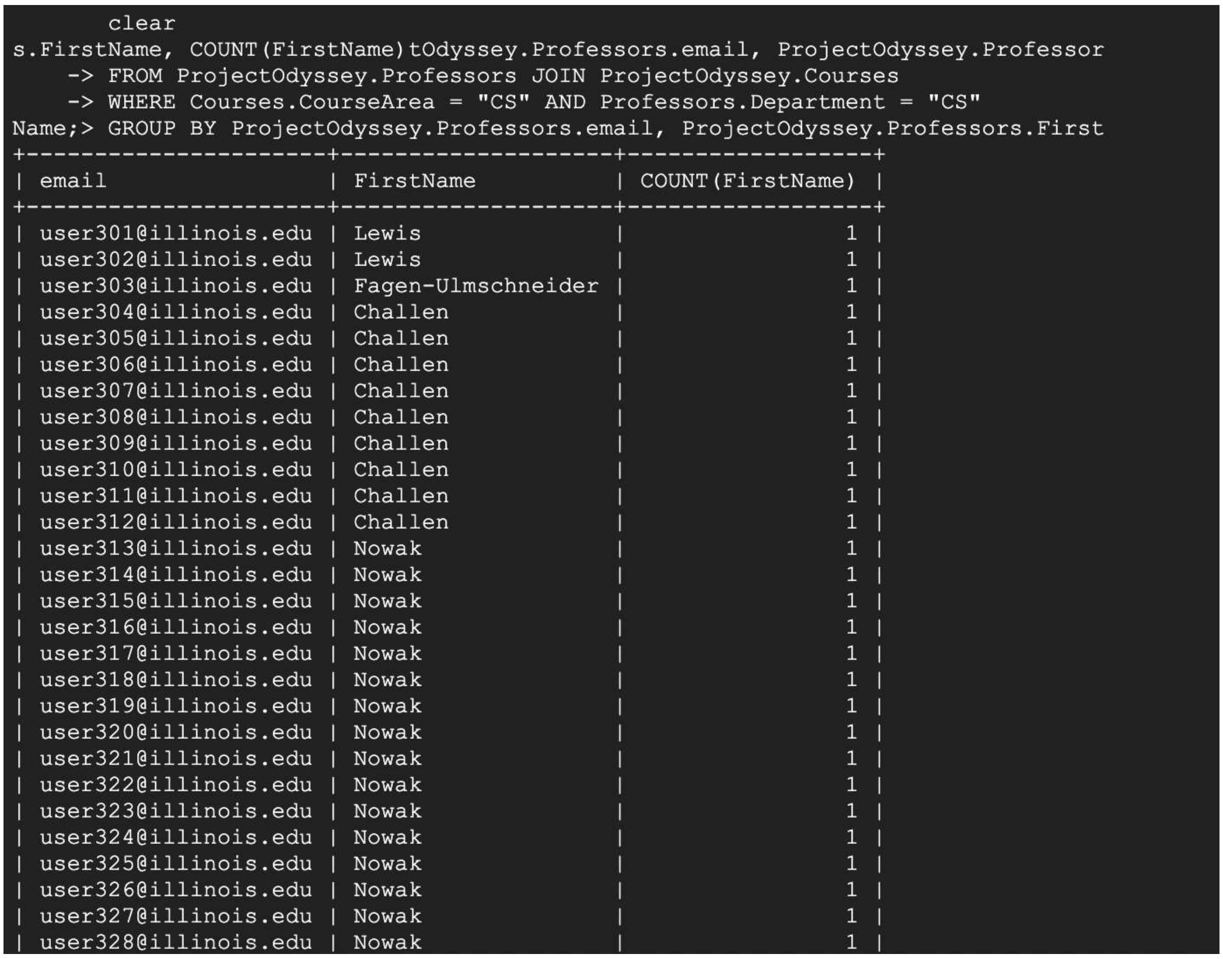


```
12 rows in set (0.00 sec)

mysql> Select CourseName, RSOName, COUNT(CourseName) as cnt
    -> from ProjectOdyssey.Courses c JOIN ProjectOdyssey.RSOs r ON c.CourseArea = r.RSOArea
    -> WHERE c.CourseArea = "CS" AND r.RSOArea = "CS"
    -> GROUP BY CourseName, RSOName
    -> ORDER BY cnt desc;
+----------------------+---------------------------------------------+-----+
| CourseName           | RSOName                                     | cnt |
+----------------------+---------------------------------------------+-----+
| Special Topics in CS | Latinx's In CS                              |   1 |
| Special Topics in CS | Project: Code                               |   1 |
| Special Topics in CS | Women in CyberSecurity                      |   1 |
| Special Topics in CS | Virtual Reality Club                        |   1 |
| Special Topics in CS | Satellite Development Organization (SatDev) |   1 |
| Special Topics in CS | Hack4Impact                                 |   1 |
| Special Topics in CS | Women in Computer Science                   |   1 |
| Special Topics in CS | NeuroTech                                   |   1 |
| Special Topics in CS | Developer Student Club                      |   1 |
| Special Topics in CS | Blacks and African Americans in Computing   |   1 |
| Special Topics in CS | Computer Science Sail                       |   1 |
| Special Topics in CS | Association for Computing Machinery         |   1 |
+----------------------+---------------------------------------------+-----+
12 rows in set (0.00 sec)
```

Query 2

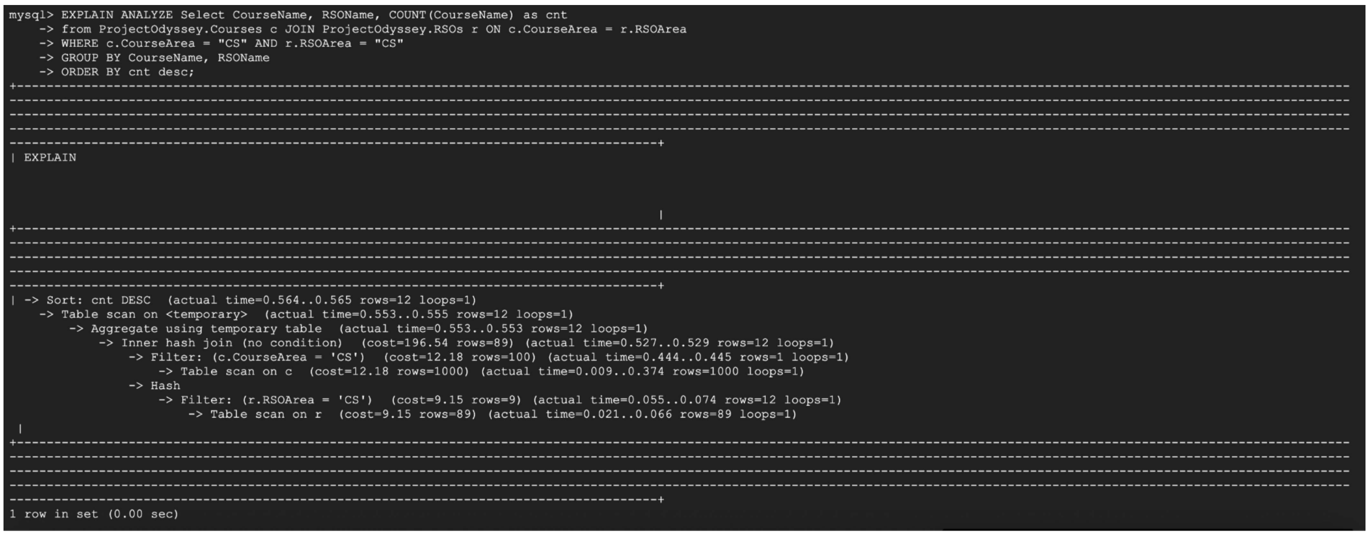


```
SELECT DISTINCT ProjectOdyssey.Professors.email, ProjectOdyssey.Professors.FirstName, COUNT(FirstName)
FROM ProjectOdyssey.Professors JOIN ProjectOdyssey.Courses
WHERE Courses.CourseArea = "CS" AND Professors.Department = "CS"
GROUP BY ProjectOdyssey.Professors.email, ProjectOdyssey.Professors.FirstName

/*If a student wants to get in contact with a professor
that is both in the department of interest and has taught a class in
the students interest. In this example the interest is CS. We
also provide the count of the number of class of interest the
professor teaches*/
```

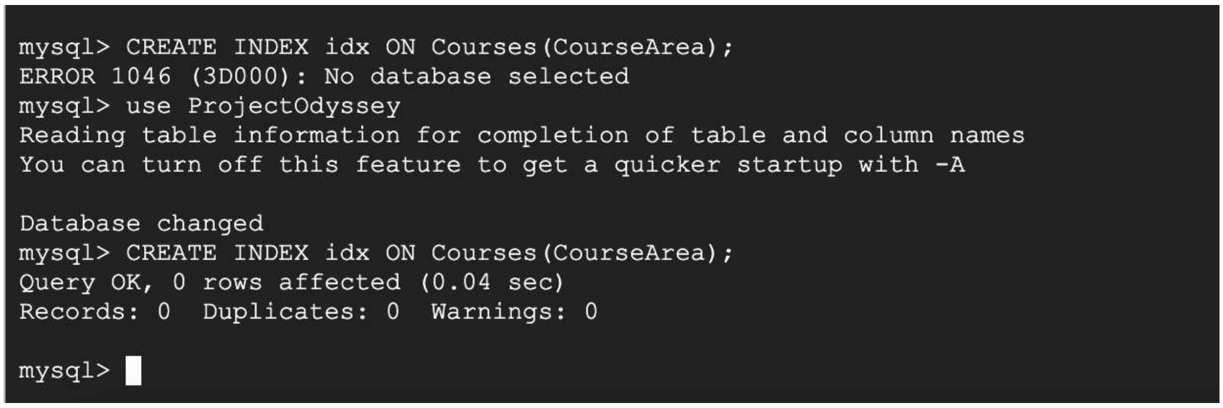


```
       clear
s.FirstName, COUNT(FirstName)tOdyssey.Professors.email, ProjectOdyssey.Professor
    -> FROM ProjectOdyssey.Professors JOIN ProjectOdyssey.Courses
    -> WHERE Courses.CourseArea = "CS" AND Professors.Department = "CS"
Name;> GROUP BY ProjectOdyssey.Professors.email, ProjectOdyssey.Professors.First
+----------------------+--------------------+------------------+
| email                | FirstName          | COUNT(FirstName) |
+----------------------+--------------------+------------------+
| user301@illinois.edu | Lewis              |                1 |
| user302@illinois.edu | Lewis              |                1 |
| user303@illinois.edu | Fagen-Ulmschneider |                1 |
| user304@illinois.edu | Challen            |                1 |
| user305@illinois.edu | Challen            |                1 |
| user306@illinois.edu | Challen            |                1 |
| user307@illinois.edu | Challen            |                1 |
| user308@illinois.edu | Challen            |                1 |
| user309@illinois.edu | Challen            |                1 |
| user310@illinois.edu | Challen            |                1 |
| user311@illinois.edu | Challen            |                1 |
| user312@illinois.edu | Challen            |                1 |
| user313@illinois.edu | Nowak              |                1 |
| user314@illinois.edu | Nowak              |                1 |
| user315@illinois.edu | Nowak              |                1 |
| user316@illinois.edu | Nowak              |                1 |
| user317@illinois.edu | Nowak              |                1 |
| user318@illinois.edu | Nowak              |                1 |
| user319@illinois.edu | Nowak              |                1 |
| user320@illinois.edu | Nowak              |                1 |
| user321@illinois.edu | Nowak              |                1 |
| user322@illinois.edu | Nowak              |                1 |
| user323@illinois.edu | Nowak              |                1 |
| user324@illinois.edu | Nowak              |                1 |
| user325@illinois.edu | Nowak              |                1 |
| user326@illinois.edu | Nowak              |                1 |
| user327@illinois.edu | Nowak              |                1 |
| user328@illinois.edu | Nowak              |                1 |
```

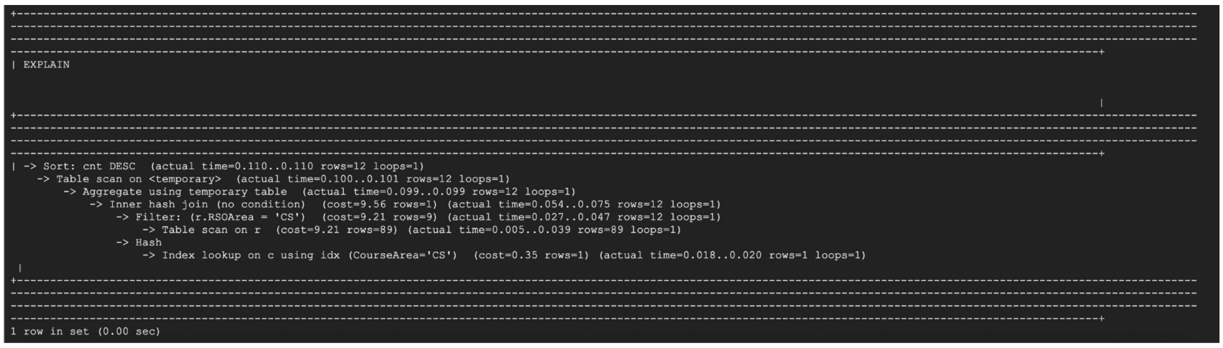
Index Analysis Query 1



```
mysql> EXPLAIN ANALYZE Select CourseName, RSOName, COUNT(CourseName) as cnt
    -> from ProjectOdyssey.Courses c JOIN ProjectOdyssey.RSOs r ON c.CourseArea = r.RSOArea
    -> WHERE c.CourseArea = "CS" AND r.RSOArea = "CS"
    -> GROUP BY CourseName, RSOName
    -> ORDER BY cnt desc;
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------+
| EXPLAIN



                                                                                      |
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------+
| -> Sort: cnt DESC  (actual time=0.564..0.565 rows=12 loops=1)
    -> Table scan on <temporary>  (actual time=0.553..0.555 rows=12 loops=1)
        -> Aggregate using temporary table  (actual time=0.553..0.553 rows=12 loops=1)
            -> Inner hash join (no condition)  (cost=196.54 rows=89) (actual time=0.527..0.529 rows=12 loops=1)
                -> Filter: (c.CourseArea = 'CS')  (cost=12.18 rows=100) (actual time=0.444..0.445 rows=1 loops=1)
                    -> Table scan on c  (cost=12.18 rows=1000) (actual time=0.009..0.374 rows=1000 loops=1)
                -> Hash
                    -> Filter: (r.RSOArea = 'CS')  (cost=9.15 rows=9) (actual time=0.055..0.074 rows=12 loops=1)
                        -> Table scan on r  (cost=9.15 rows=89) (actual time=0.021..0.066 rows=89 loops=1)
 |
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```



```
mysql> CREATE INDEX idx ON Courses(CourseArea);
ERROR 1046 (3D000): No database selected
mysql> use ProjectOdyssey
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> CREATE INDEX idx ON Courses(CourseArea);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> 
```

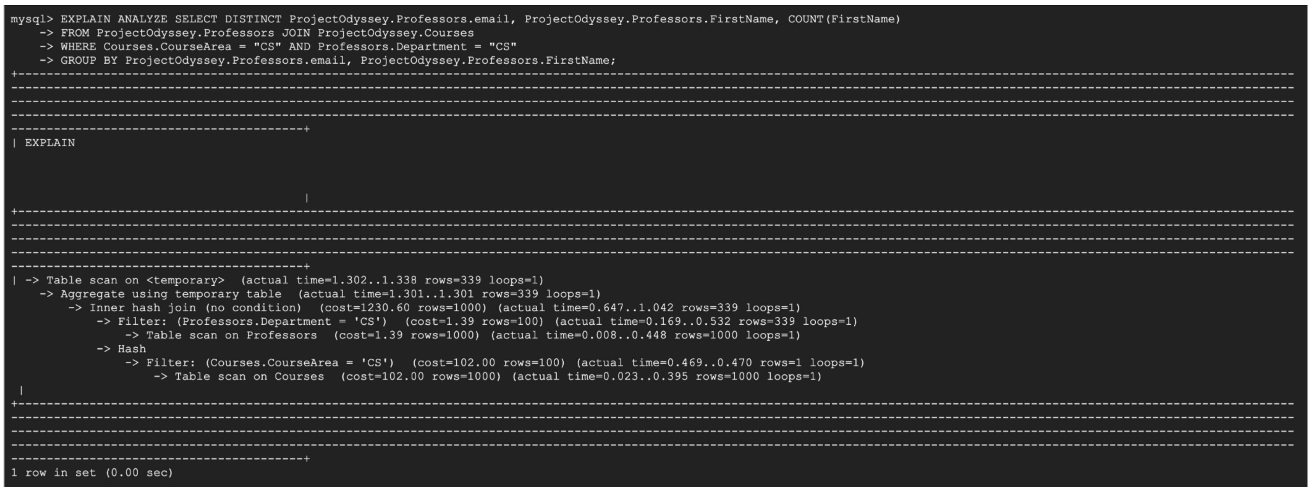


```
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------+
| EXPLAIN


                                                                                                                                          |
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------+
| -> Sort: cnt DESC  (actual time=0.110..0.110 rows=12 loops=1)
    -> Table scan on <temporary>  (actual time=0.100..0.101 rows=12 loops=1)
        -> Aggregate using temporary table  (actual time=0.099..0.099 rows=12 loops=1)
            -> Inner hash join (no condition)  (cost=9.56 rows=1) (actual time=0.054..0.075 rows=12 loops=1)
                -> Filter: (r.RSOArea = 'CS')  (cost=9.21 rows=9) (actual time=0.027..0.047 rows=12 loops=1)
                    -> Table scan on r  (cost=9.21 rows=89) (actual time=0.005..0.039 rows=89 loops=1)
                -> Hash
                    -> Index lookup on c using idx (CourseArea='CS')  (cost=0.35 rows=1) (actual time=0.018..0.020 rows=1 loops=1)
 |
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

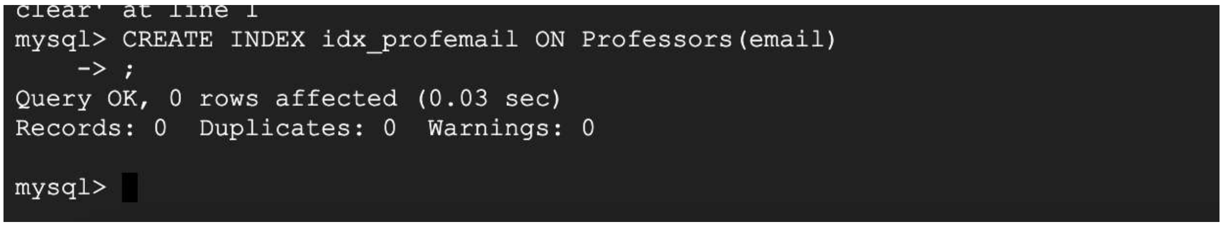
Findings and Explanation

Improvement: We find that the timing improved with the given index above. Instead of taking 0.564…0.565 seconds, the query now takes 0.110 seconds. We think there was an improvement because we think that the indexes on columns that are joined speed up the process of combining data from the joined tables.

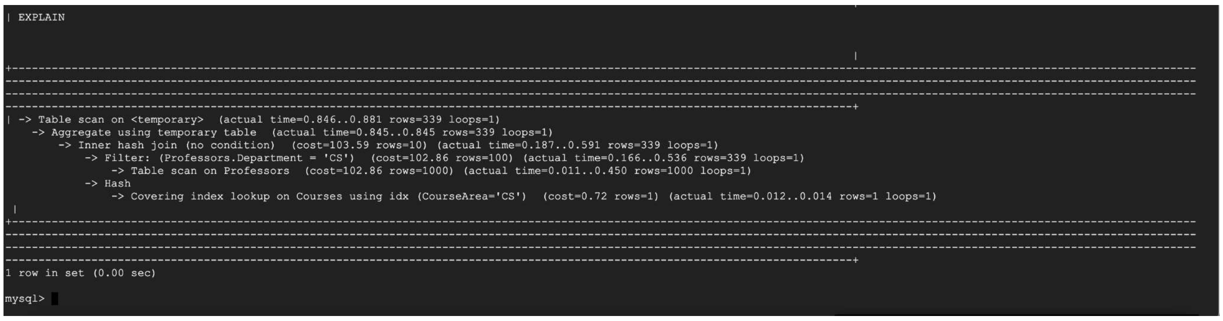
Index Analysis Query 2



```
mysql> EXPLAIN ANALYZE SELECT DISTINCT ProjectOdyssey.Professors.email, ProjectOdyssey.Professors.FirstName, COUNT(FirstName)
    -> FROM ProjectOdyssey.Professors JOIN ProjectOdyssey.Courses
    -> WHERE Courses.CourseArea = "CS" AND Professors.Department = "CS"
    -> GROUP BY ProjectOdyssey.Professors.email, ProjectOdyssey.Professors.FirstName;
+-------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------+
| EXPLAIN



                                         |
+-------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------+
| -> Table scan on <temporary>  (actual time=1.302..1.338 rows=339 loops=1)
    -> Aggregate using temporary table  (actual time=1.301..1.301 rows=339 loops=1)
        -> Inner hash join (no condition)  (cost=1230.60 rows=1000) (actual time=0.647..1.042 rows=339 loops=1)
            -> Filter: (Professors.Department = 'CS')  (cost=1.39 rows=100) (actual time=0.169..0.532 rows=339 loops=1)
                -> Table scan on Professors  (cost=1.39 rows=1000) (actual time=0.008..0.448 rows=1000 loops=1)
            -> Hash
                -> Filter: (Courses.CourseArea = 'CS')  (cost=102.00 rows=100) (actual time=0.469..0.470 rows=1 loops=1)
                    -> Table scan on Courses  (cost=102.00 rows=1000) (actual time=0.023..0.395 rows=1000 loops=1)
 |
+-------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------+
1 row in set (0.00 sec)
```



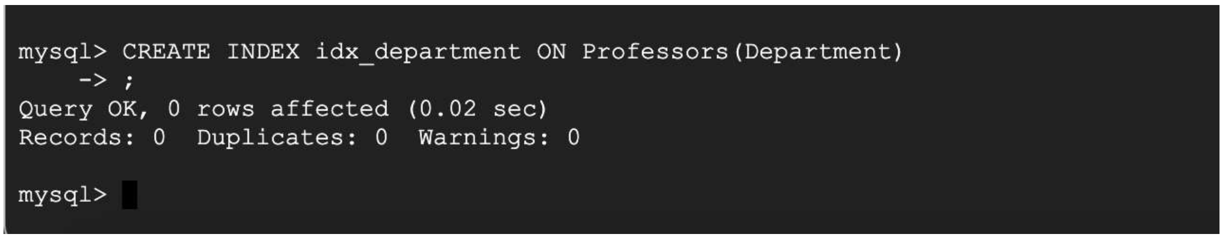
```
clear' at line 1
mysql> CREATE INDEX idx_profemail ON Professors(email)
    -> ;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> 
```



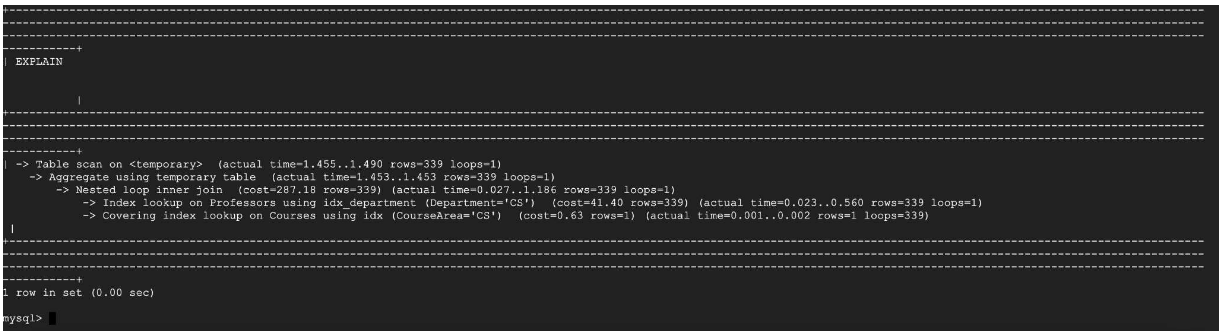
```
| EXPLAIN


                                                                                                                                                   |
+-------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------+
| -> Table scan on <temporary>  (actual time=0.846..0.881 rows=339 loops=1)
    -> Aggregate using temporary table  (actual time=0.845..0.845 rows=339 loops=1)
        -> Inner hash join (no condition)  (cost=103.59 rows=10) (actual time=0.187..0.591 rows=339 loops=1)
            -> Filter: (Professors.Department = 'CS')  (cost=102.86 rows=100) (actual time=0.166..0.536 rows=339 loops=1)
                -> Table scan on Professors  (cost=102.86 rows=1000) (actual time=0.011..0.450 rows=1000 loops=1)
            -> Hash
                -> Covering index lookup on Courses using idx (CourseArea='CS')  (cost=0.72 rows=1) (actual time=0.012..0.014 rows=1 loops=1)
 |
+-------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)

mysql> 
```

Findings and Explanation
Improvement: We find that the timing improved with the given index above. Instead of taking 1.302…1.338 seconds, the query now takes 0.846…0.881 seconds. We think there was an improvement because we think that the indices allow the database to locate the rows that match the criteria of the query.

2nd index for query 2



```
mysql> CREATE INDEX idx_department ON Professors(Department)
    -> ;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> 
```



```
+---------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------
----------+
| EXPLAIN


         |
+---------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------
----------+
| -> Table scan on <temporary>  (actual time=1.455..1.490 rows=339 loops=1)
    -> Aggregate using temporary table  (actual time=1.453..1.453 rows=339 loops=1)
        -> Nested loop inner join  (cost=287.18 rows=339) (actual time=0.027..1.186 rows=339 loops=1)
            -> Index lookup on Professors using idx_department (Department='CS')  (cost=41.40 rows=339) (actual time=0.023..0.560 rows=339 loops=1)
            -> Covering index lookup on Courses using idx (CourseArea='CS')  (cost=0.63 rows=1) (actual time=0.001..0.002 rows=1 loops=339)
 |
+---------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------
----------+
1 row in set (0.00 sec)

mysql> 
```

Findings and Explanation
No Improvement: We find that the timing improved with the given index above. Instead of taking 1.302…1.338 seconds, the query now takes 1.455…1.490 seconds. We think there was no improvement because we think that creating too many indexes on a table leads to increasingly worse performance. This is because each index would take up storage space, and the database needs to track these during the modification process.