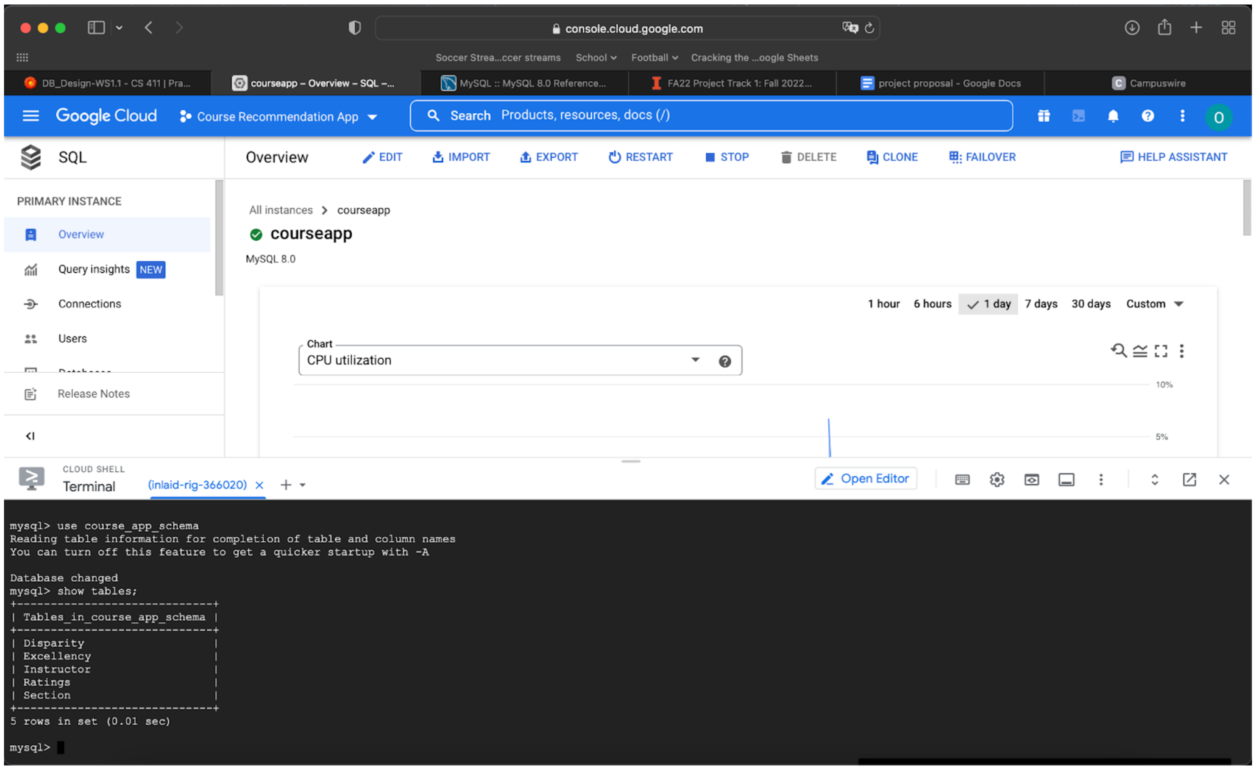
Connection to GCP:



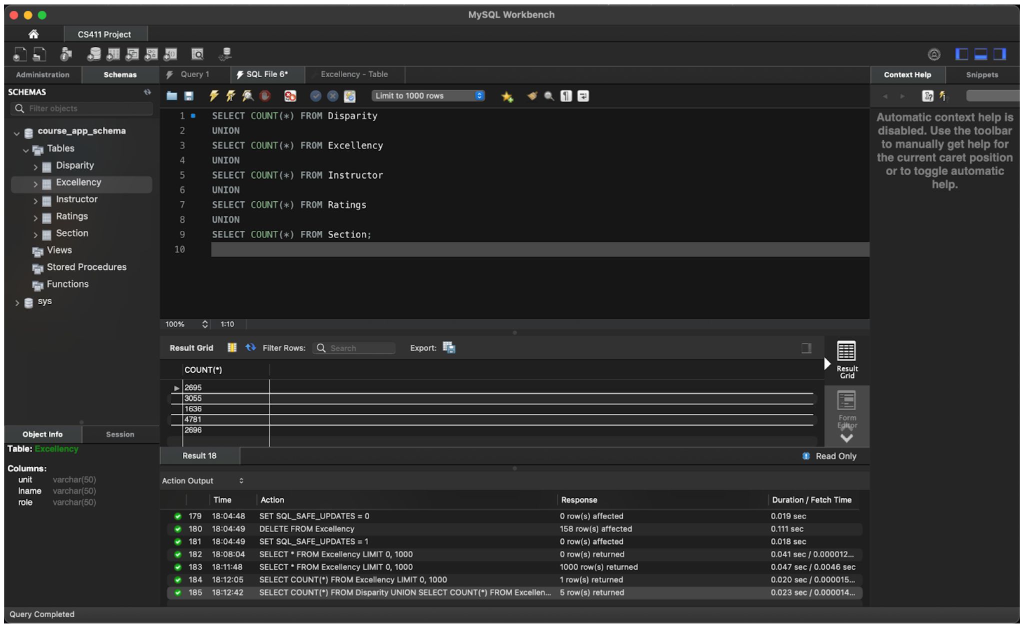
```
mysql> use course_app_schema
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------------------+
| Tables_in_course_app_schema |
+-----------------------------+
| Disparity                   |
| Excellency                  |
| Instructor                  |
| Ratings                     |
| Section                     |
+-----------------------------+
5 rows in set (0.01 sec)

mysql> 
```

Database DDLs:

```
CREATE TABLE `Disparity` (
  `CRN` varchar(10) NOT NULL,
  `A` int DEFAULT NULL,
  `B` int DEFAULT NULL,
  `C` int DEFAULT NULL,
  `D` int DEFAULT NULL,
  `F` int DEFAULT NULL,
  PRIMARY KEY (`CRN`)
);

CREATE TABLE `Excellency` (
  `unit` varchar(100) DEFAULT NULL,
  `lname` varchar(50) DEFAULT NULL,
  `role` varchar(50) DEFAULT NULL
);
```

```
CREATE TABLE `Instructor` (
  `name` varchar(50) NOT NULL,
  `average_rating` double DEFAULT NULL,
  `excellency_count` double DEFAULT NULL,
  PRIMARY KEY (`name`)
);

CREATE TABLE `Ratings` (
  `student_username` varchar(50) NOT NULL,
  `professor_name` varchar(50) DEFAULT NULL,
  `student_rating` double DEFAULT NULL,
  `student_difficulty` double DEFAULT NULL,
  PRIMARY KEY (`student_username`)
);

CREATE TABLE `Section` (
  `Subject` text,
  `Course` int DEFAULT NULL,
  `CRN` int DEFAULT NULL,
  `Course_Title` text,
  `Course_Section` text,
  `Average_Grade` double DEFAULT NULL,
  `Primary_Instructor` text
);
```

Database Counts:



```
SELECT COUNT(*) FROM Disparity
UNION
SELECT COUNT(*) FROM Excellency
UNION
SELECT COUNT(*) FROM Instructor
UNION
SELECT COUNT(*) FROM Ratings
UNION
SELECT COUNT(*) FROM Section;
```

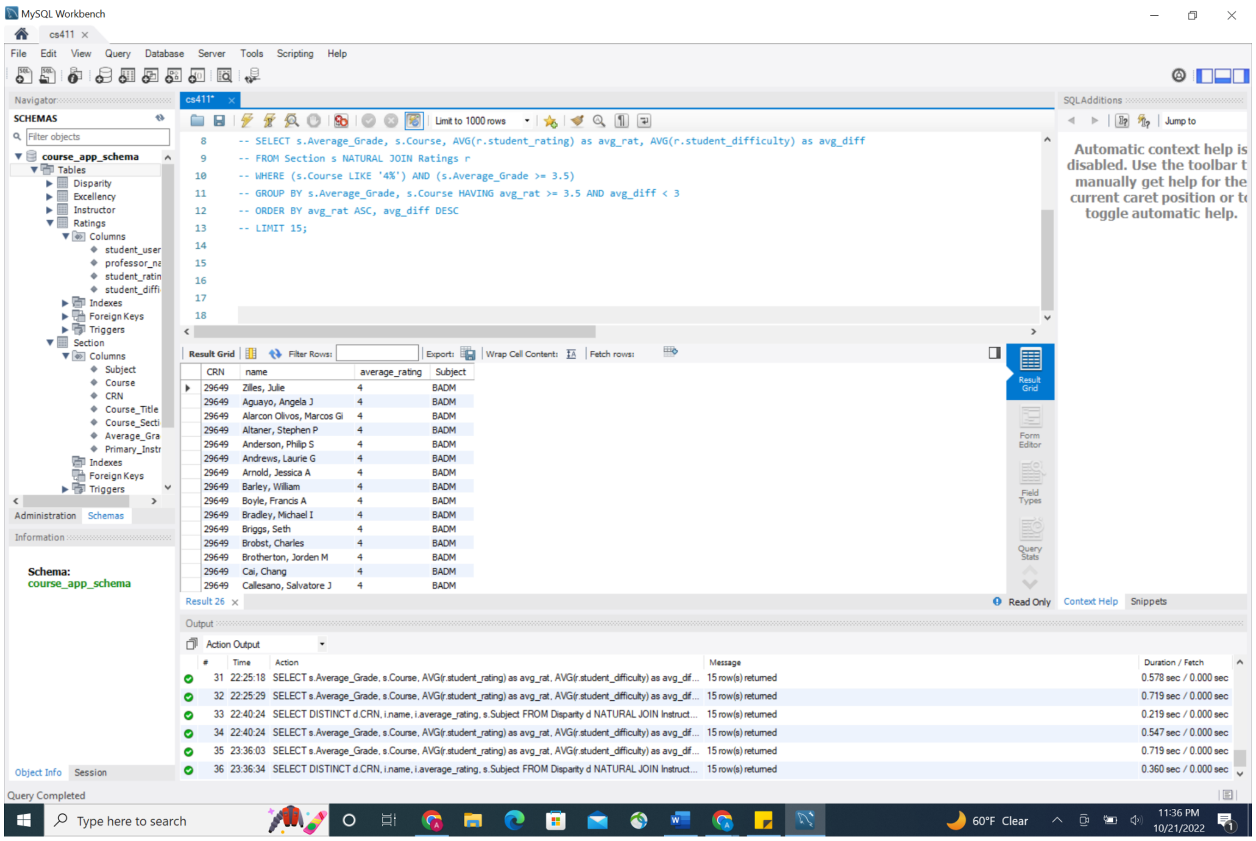
```
COUNT(*)
2695
3055
1636
4781
2696
```

Action Output

```
     Time      Action                                                             Response              Duration / Fetch Time
179  18:04:48  SET SQL_SAFE_UPDATES = 0                                           0 row(s) affected     0.019 sec
180  18:04:49  DELETE FROM Excellency                                             158 row(s) affected   0.111 sec
181  18:04:49  SET SQL_SAFE_UPDATES = 1                                           0 row(s) affected     0.018 sec
182  18:08:04  SELECT * FROM Excellency LIMIT 0, 1000                             0 row(s) returned     0.041 sec / 0.000012...
183  18:11:48  SELECT * FROM Excellency LIMIT 0, 1000                             1000 row(s) returned  0.047 sec / 0.0046 sec
184  18:12:05  SELECT COUNT(*) FROM Excellency LIMIT 0, 1000                      1 row(s) returned     0.020 sec / 0.000015...
185  18:12:42  SELECT COUNT(*) FROM Disparity UNION SELECT COUNT(*) FROM Excellen... 5 row(s) returned  0.023 sec / 0.000014...
```

ADVANCED QUERY #1:

SELECT DISTINCT d.CRN, i.name, i.average_rating, s.Subject
FROM Disparity d NATURAL JOIN Instructor i NATURAL JOIN Section s
WHERE (d.A >= (d.A + d.B + d.C + d.D + d.F) * .9) AND (i.average_rating >= 4) AND ('BADM' IN (SELECT s.Subject FROM Section s))
GROUP BY d.CRN, i.name, i.average_rating, s.Subject
ORDER BY i.average_rating DESC, d.CRN
LIMIT 15;



```
-- SELECT s.Average_Grade, s.Course, AVG(r.student_rating) as avg_rat, AVG(r.student_difficulty) as avg_diff
-- FROM Section s NATURAL JOIN Ratings r
-- WHERE (s.Course LIKE '4%') AND (s.Average_Grade >= 3.5)
-- GROUP BY s.Average_Grade, s.Course HAVING avg_rat >= 3.5 AND avg_diff < 3
-- ORDER BY avg_rat ASC, avg_diff DESC
-- LIMIT 15;
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

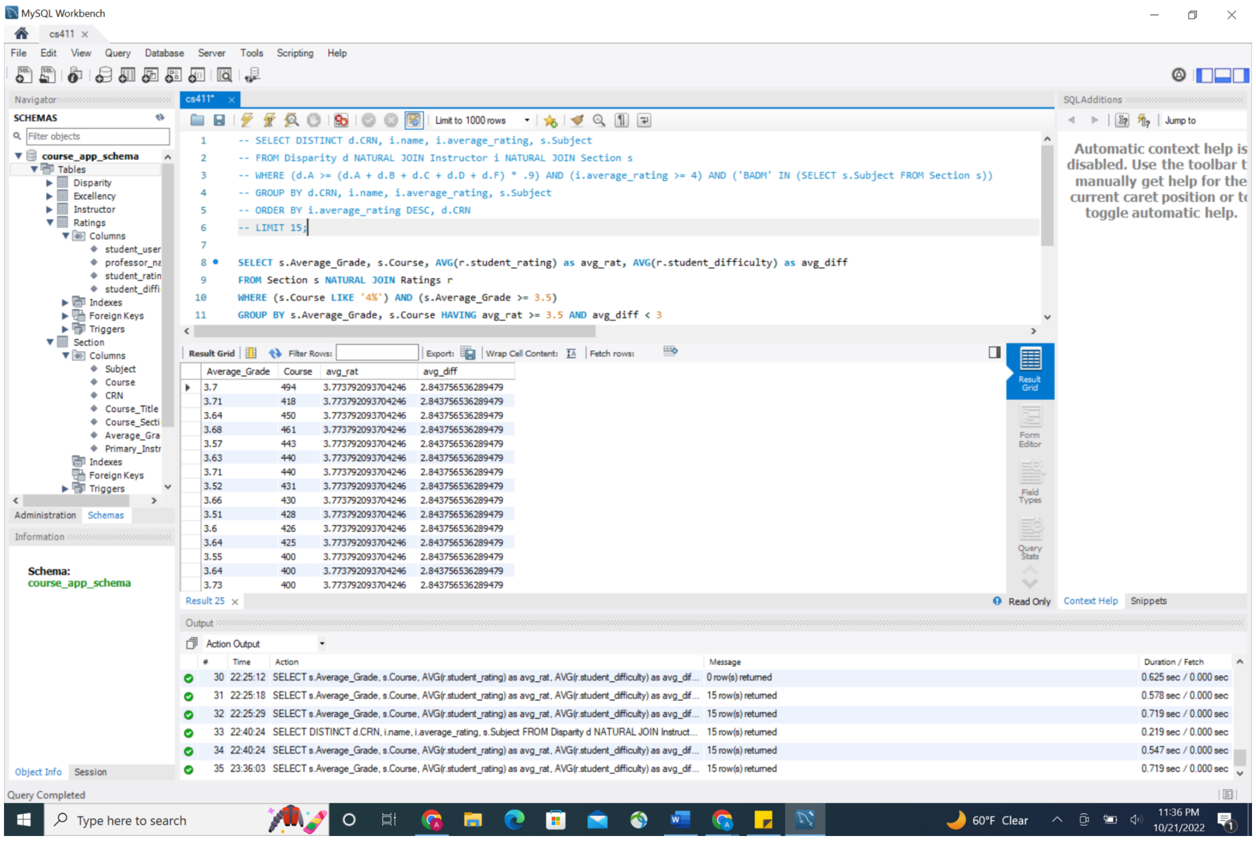
CRN	name	average_rating	Subject
29649	Zilles, Julie	4	BADM
29649	Aguayo, Angela J	4	BADM
29649	Alarcon Olivos, Marcos Gi	4	BADM
29649	Altaner, Stephen P	4	BADM
29649	Anderson, Philip S	4	BADM
29649	Andrews, Laurie G	4	BADM
29649	Arnold, Jessica A	4	BADM
29649	Barley, William	4	BADM
29649	Boyle, Francis A	4	BADM
29649	Bradley, Michael I	4	BADM
29649	Briggs, Seth	4	BADM
29649	Brobst, Charles	4	BADM
29649	Brotherton, Jorden M	4	BADM
29649	Cai, Chang	4	BADM
29649	Callesano, Salvatore J	4	BADM

Result 26

```
#   Time      Action                                                                                   Message              Duration / Fetch
31  22:25:18  SELECT s.Average_Grade, s.Course, AVG(r.student_rating) as avg_rat, AVG(r.student_difficulty) as avg_dif...  15 row(s) returned  0.578 sec / 0.000 sec
32  22:25:29  SELECT s.Average_Grade, s.Course, AVG(r.student_rating) as avg_rat, AVG(r.student_difficulty) as avg_dif...  15 row(s) returned  0.719 sec / 0.000 sec
33  22:40:24  SELECT DISTINCT d.CRN, i.name, i.average_rating, s.Subject FROM Disparity d NATURAL JOIN Instruct...        15 row(s) returned  0.219 sec / 0.000 sec
34  22:40:24  SELECT s.Average_Grade, s.Course, AVG(r.student_rating) as avg_rat, AVG(r.student_difficulty) as avg_dif...  15 row(s) returned  0.547 sec / 0.000 sec
35  23:36:03  SELECT s.Average_Grade, s.Course, AVG(r.student_rating) as avg_rat, AVG(r.student_difficulty) as avg_dif...  15 row(s) returned  0.719 sec / 0.000 sec
36  23:36:34  SELECT DISTINCT d.CRN, i.name, i.average_rating, s.Subject FROM Disparity d NATURAL JOIN Instruct...        15 row(s) returned  0.360 sec / 0.000 sec
```

ADVANCED QUERY #2:

SELECT s.Average_Grade, s.Course, AVG(r.student_rating) as avg_rat, AVG(r.student_difficulty) as avg_diff
FROM Section s NATURAL JOIN Ratings r
WHERE (s.Course LIKE '4%') AND (s.Average_Grade >= 3.5)
GROUP BY s.Average_Grade, s.Course HAVING avg_rat >= 3.5 AND avg_diff < 3
ORDER BY avg_rat ASC, avg_diff DESC
LIMIT 15;



```sql
-- SELECT DISTINCT d.CRN, i.name, i.average_rating, s.Subject
-- FROM Disparity d NATURAL JOIN Instructor i NATURAL JOIN Section s
-- WHERE (d.A >= (d.A + d.B + d.C + d.D + d.F) * .9) AND (i.average_rating >= 4) AND ('BADM' IN (SELECT s.Subject FROM Section s))
-- GROUP BY d.CRN, i.name, i.average_rating, s.Subject
-- ORDER BY i.average_rating DESC, d.CRN
-- LIMIT 15;

SELECT s.Average_Grade, s.Course, AVG(r.student_rating) as avg_rat, AVG(r.student_difficulty) as avg_diff
FROM Section s NATURAL JOIN Ratings r
WHERE (s.Course LIKE '4%') AND (s.Average_Grade >= 3.5)
GROUP BY s.Average_Grade, s.Course HAVING avg_rat >= 3.5 AND avg_diff < 3
```

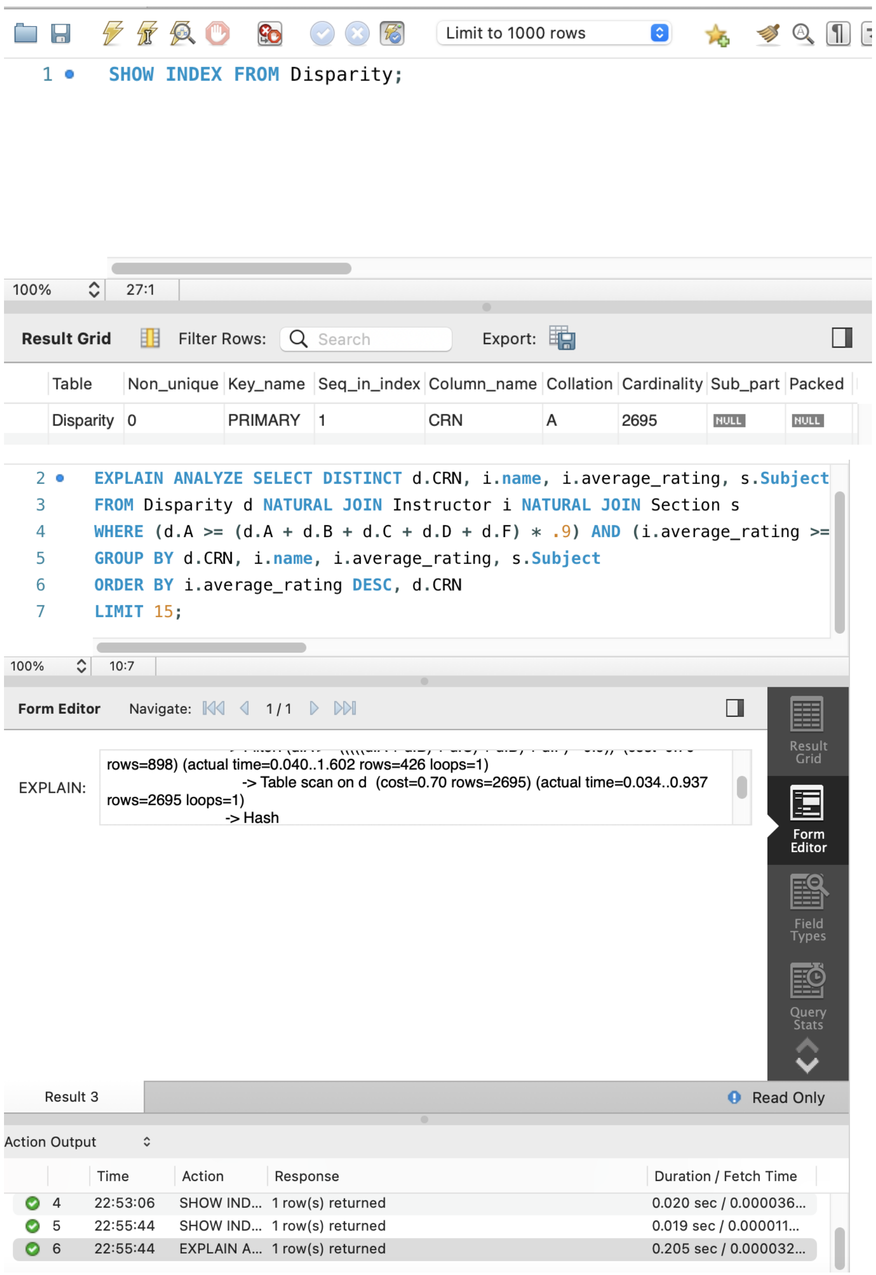
Average_Grade	Course	avg_rat	avg_diff
3.7	494	3.773792093704246	2.843756536289479
3.71	418	3.773792093704246	2.843756536289479
3.64	450	3.773792093704246	2.843756536289479
3.68	461	3.773792093704246	2.843756536289479
3.57	443	3.773792093704246	2.843756536289479
3.63	440	3.773792093704246	2.843756536289479
3.71	440	3.773792093704246	2.843756536289479
3.52	431	3.773792093704246	2.843756536289479
3.66	430	3.773792093704246	2.843756536289479
3.51	428	3.773792093704246	2.843756536289479
3.6	426	3.773792093704246	2.843756536289479
3.64	425	3.773792093704246	2.843756536289479
3.55	400	3.773792093704246	2.843756536289479
3.64	400	3.773792093704246	2.843756536289479
3.73	400	3.773792093704246	2.843756536289479

Result 25

```
#   Time      Action                                                                                          Message            Duration / Fetch
30  22:25:12  SELECT s.Average_Grade, s.Course, AVG(r.student_rating) as avg_rat, AVG(r.student_difficulty) as avg_dif...  0 row(s) returned   0.625 sec / 0.000 sec
31  22:25:18  SELECT s.Average_Grade, s.Course, AVG(r.student_rating) as avg_rat, AVG(r.student_difficulty) as avg_dif...  15 row(s) returned  0.578 sec / 0.000 sec
32  22:25:29  SELECT s.Average_Grade, s.Course, AVG(r.student_rating) as avg_rat, AVG(r.student_difficulty) as avg_dif...  15 row(s) returned  0.719 sec / 0.000 sec
33  22:40:24  SELECT DISTINCT d.CRN, i.name, i.average_rating, s.Subject FROM Disparity d NATURAL JOIN Instruct...        15 row(s) returned  0.219 sec / 0.000 sec
34  22:40:24  SELECT s.Average_Grade, s.Course, AVG(r.student_rating) as avg_rat, AVG(r.student_difficulty) as avg_dif...  15 row(s) returned  0.547 sec / 0.000 sec
35  23:36:03  SELECT s.Average_Grade, s.Course, AVG(r.student_rating) as avg_rat, AVG(r.student_difficulty) as avg_dif...  15 row(s) returned  0.719 sec / 0.000 sec
```

Query Completed

QUERY 1



```
SHOW INDEX FROM Disparity;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed
Disparity	0	PRIMARY	1	CRN	A	2695	NULL	NULL

```
EXPLAIN ANALYZE SELECT DISTINCT d.CRN, i.name, i.average_rating, s.Subject
FROM Disparity d NATURAL JOIN Instructor i NATURAL JOIN Section s
WHERE (d.A >= (d.A + d.B + d.C + d.D + d.F) * .9) AND (i.average_rating >=
GROUP BY d.CRN, i.name, i.average_rating, s.Subject
ORDER BY i.average_rating DESC, d.CRN
LIMIT 15;
```

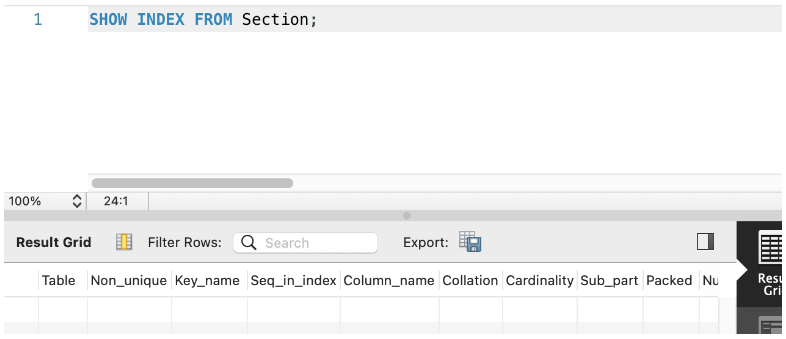
EXPLAIN:
```
rows=898) (actual time=0.040..1.602 rows=426 loops=1)
                    -> Table scan on d  (cost=0.70 rows=2695) (actual time=0.034..0.937
rows=2695 loops=1)
                  -> Hash
```

Result 3 — Read Only

Action Output

		Time	Action	Response	Duration / Fetch Time
✓	4	22:53:06	SHOW IND...	1 row(s) returned	0.020 sec / 0.000036...
✓	5	22:55:44	SHOW IND...	1 row(s) returned	0.019 sec / 0.000011...
✓	6	22:55:44	EXPLAIN A...	1 row(s) returned	0.205 sec / 0.000032...

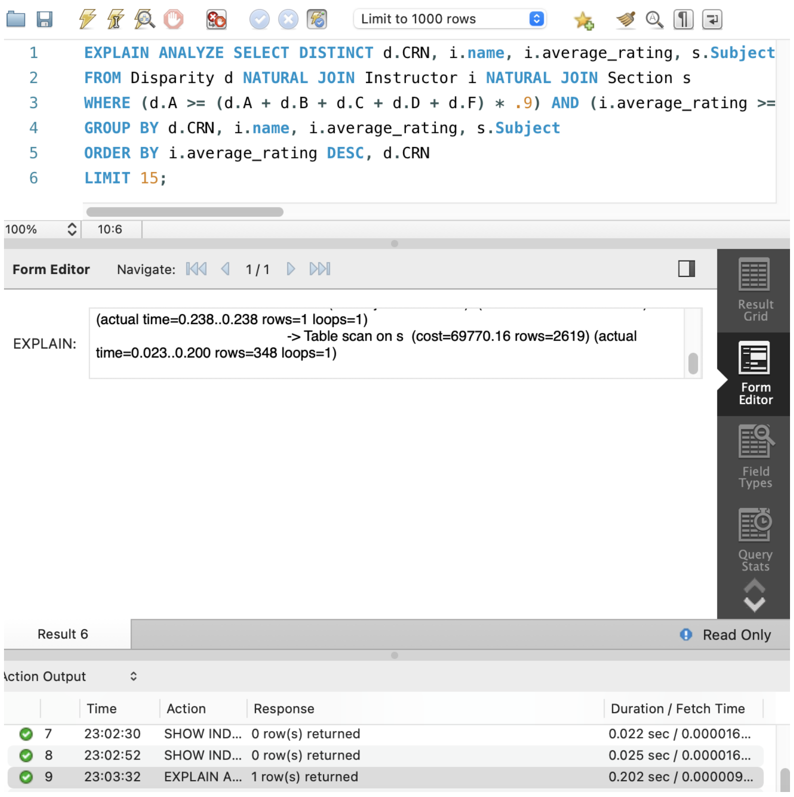
Current index on Disparity is on the primary key of CRN. The cost of this is currently 0.70 and has a runtime of 0.205 seconds.



```
SHOW INDEX FROM Section;
```

Result Grid | Filter Rows: | Export:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Nu



```
EXPLAIN ANALYZE SELECT DISTINCT d.CRN, i.name, i.average_rating, s.Subject
FROM Disparity d NATURAL JOIN Instructor i NATURAL JOIN Section s
WHERE (d.A >= (d.A + d.B + d.C + d.D + d.F) * .9) AND (i.average_rating >=
GROUP BY d.CRN, i.name, i.average_rating, s.Subject
ORDER BY i.average_rating DESC, d.CRN
LIMIT 15;
```

Form Editor Navigate: 1 / 1

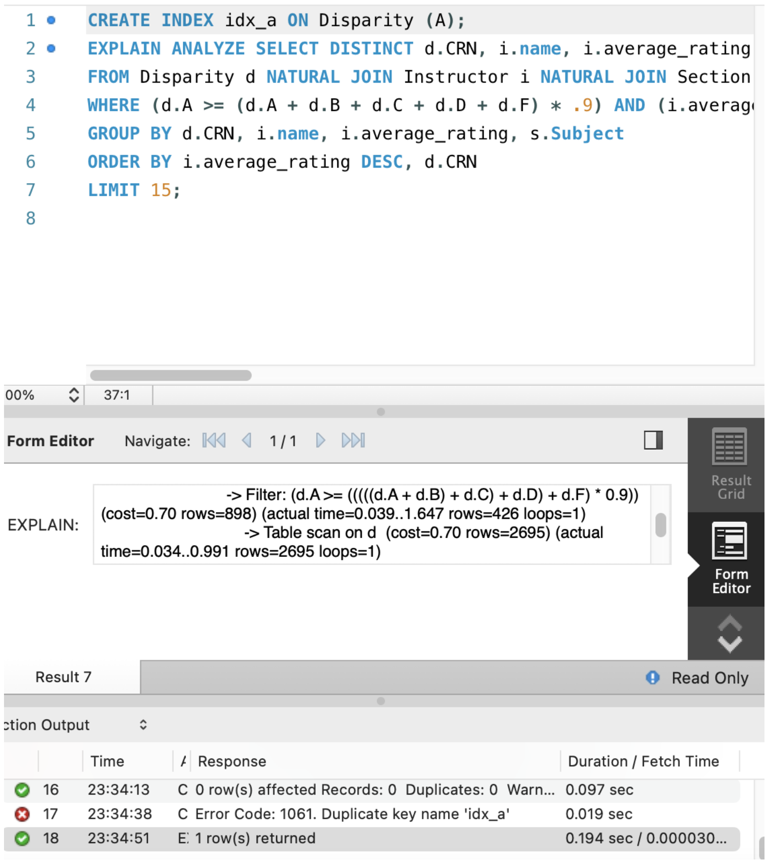
EXPLAIN:
```
(actual time=0.238..0.238 rows=1 loops=1)
                    -> Table scan on s  (cost=69770.16 rows=2619) (actual
time=0.023..0.200 rows=348 loops=1)
```

Result 6   Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
7	23:02:30	SHOW IND...	0 row(s) returned	0.022 sec / 0.000016...
8	23:02:52	SHOW IND...	0 row(s) returned	0.025 sec / 0.000016...
9	23:03:32	EXPLAIN A...	1 row(s) returned	0.202 sec / 0.000009...

Current index on Section doesn’t exist since there isn’t a primary key. The cost of this is currently 69770 and has a runtime of 0.202 seconds.



```
CREATE INDEX idx_a ON Disparity (A);
EXPLAIN ANALYZE SELECT DISTINCT d.CRN, i.name, i.average_rating
FROM Disparity d NATURAL JOIN Instructor i NATURAL JOIN Section
WHERE (d.A >= (d.A + d.B + d.C + d.D + d.F) * .9) AND (i.averag
GROUP BY d.CRN, i.name, i.average_rating, s.Subject
ORDER BY i.average_rating DESC, d.CRN
LIMIT 15;
```

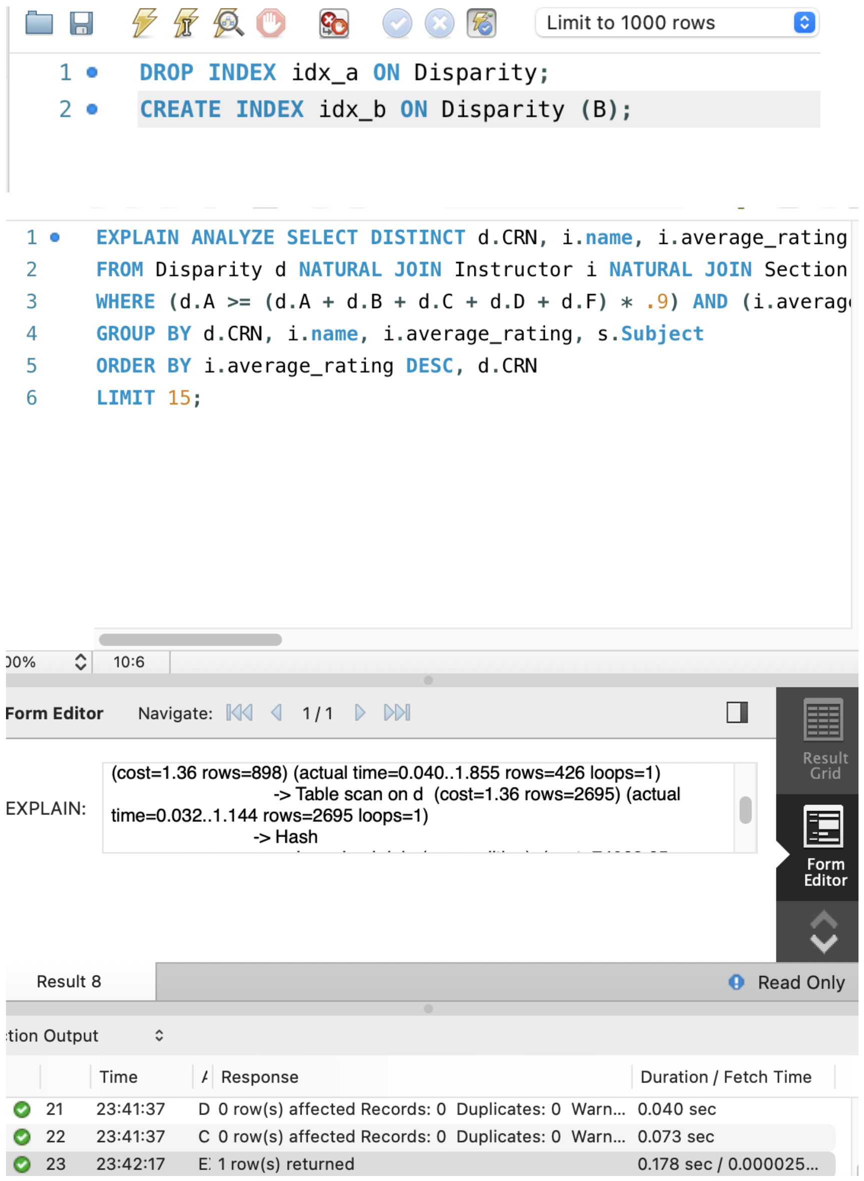
```
00%   37:1
```

Form Editor Navigate: 1 / 1

EXPLAIN:
```
                    -> Filter: (d.A >= (((((d.A + d.B) + d.C) + d.D) + d.F) * 0.9))
(cost=0.70 rows=898) (actual time=0.039..1.647 rows=426 loops=1)
                        -> Table scan on d  (cost=0.70 rows=2695) (actual
time=0.034..0.991 rows=2695 loops=1)
```

```
    Time      A Response                                          Duration / Fetch Time
16  23:34:13  C 0 row(s) affected Records: 0  Duplicates: 0  Warn... 0.097 sec
17  23:34:38  C Error Code: 1061. Duplicate key name 'idx_a'         0.019 sec
18  23:34:51  E 1 row(s) returned                                    0.194 sec / 0.000030...
```

Creating a new index on the attribute A from disparity led to a table scan on d have still only a cost of 0.70, but interestingly this result was returned faster with only a time of 0.194 seconds instead of 0.202.


Limit to 1000 rows

```
DROP INDEX idx_a ON Disparity;
CREATE INDEX idx_b ON Disparity (B);
```

```
EXPLAIN ANALYZE SELECT DISTINCT d.CRN, i.name, i.average_rating
FROM Disparity d NATURAL JOIN Instructor i NATURAL JOIN Section
WHERE (d.A >= (d.A + d.B + d.C + d.D + d.F) * .9) AND (i.averag
GROUP BY d.CRN, i.name, i.average_rating, s.Subject
ORDER BY i.average_rating DESC, d.CRN
LIMIT 15;
```

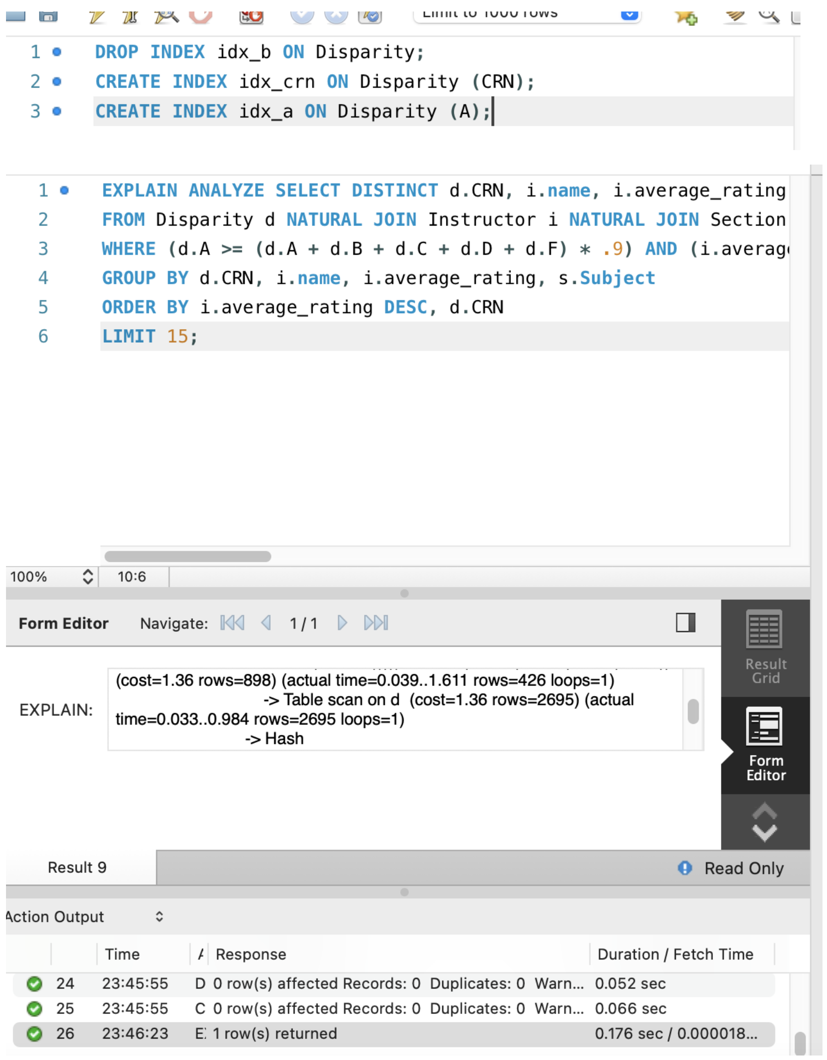
EXPLAIN:

```
(cost=1.36 rows=898) (actual time=0.040..1.855 rows=426 loops=1)
                    -> Table scan on d  (cost=1.36 rows=2695) (actual
time=0.032..1.144 rows=2695 loops=1)
                  -> Hash
```

```
   Time      Response                                                  Duration / Fetch Time
21 23:41:37  D 0 row(s) affected Records: 0  Duplicates: 0  Warn...   0.040 sec
22 23:41:37  C 0 row(s) affected Records: 0  Duplicates: 0  Warn...   0.073 sec
23 23:42:17  E 1 row(s) returned                                      0.178 sec / 0.000025...
```

Now creating a new index on B instead, we see that the cost to scan has gone up to 1.36, but the time has decreased still to 0.178 seconds.

Based on these different index structures on the Disparity table, it seems the original works best. As none of these new indexing structures have a smaller cost, it wouldn't make sense to change the structure from the original.



```sql
DROP INDEX idx_b ON Disparity;
CREATE INDEX idx_crn ON Disparity (CRN);
CREATE INDEX idx_a ON Disparity (A);
```

```sql
EXPLAIN ANALYZE SELECT DISTINCT d.CRN, i.name, i.average_rating
FROM Disparity d NATURAL JOIN Instructor i NATURAL JOIN Section
WHERE (d.A >= (d.A + d.B + d.C + d.D + d.F) * .9) AND (i.averag
GROUP BY d.CRN, i.name, i.average_rating, s.Subject
ORDER BY i.average_rating DESC, d.CRN
LIMIT 15;
```

```
EXPLAIN: (cost=1.36 rows=898) (actual time=0.039..1.611 rows=426 loops=1)
                        -> Table scan on d  (cost=1.36 rows=2695) (actual
time=0.033..0.984 rows=2695 loops=1)
                     -> Hash
```

	Time	Action	Response	Duration / Fetch Time
24	23:45:55	D	0 row(s) affected Records: 0 Duplicates: 0 Warn...	0.052 sec
25	23:45:55	C	0 row(s) affected Records: 0 Duplicates: 0 Warn...	0.066 sec
26	23:46:23	E	1 row(s) returned	0.176 sec / 0.000018...

By creating a new index on Disparity of first CRN and then on A, we see that cost has increased to 1.36 from the original of 0.70, but time has gone down to 0.176.