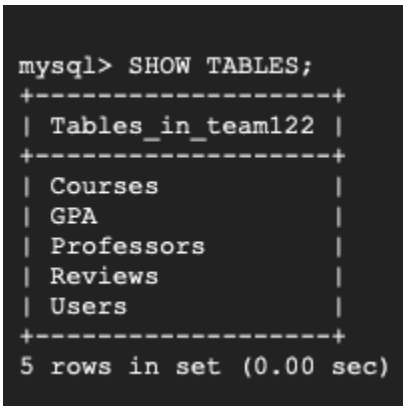
CS 411: Project Track 1 Stage 3

Part 1:

1. We implemented 5 tables in GCP: Users, Courses, GPA, Professors, and Reviews



2. The DDL translation for these was uploaded to GitHub, a screenshot is also included here:

```sql
CREATE TABLE Users (
    Username VARCHAR (244),
    Password VARCHAR(255),
    FirstName VARCHAR(255),
    LastName VARCHAR(255),
    IsAdmin BIT,Cours
    PRIMARY KEY(UserName)
);

CREATE TABLE Courses (
    CourseNumber VARCHAR(255),
    CourseName VARCHAR(255),
    Department VARCHAR(255),
    CreditHours TINYINT,
    Rating DECIMAL,
    PRIMARY KEY(CourseNumber)
);


CREATE TABLE GPA (
    SectionID VARCHAR(255),
    CourseName VARCHAR(255),
    Professor VARCHAR(255),
    GPAData DECIMAL,
    PRIMARY KEY(SectionID)
);

CREATE TABLE Professors (
    ProfessorID VARCHAR(255),
    FirstName VARCHAR(255),
    LastName VARCHAR(255),
    TimesRatedExcellent TINYINT,
    Rating DECIMAL,
    PRIMARY KEY(ProfessorID)
);

CREATE TABLE Reviews (
    Username VARCHAR(255),
    CourseNumber VARCHAR(255),
    ProfessorID VARCHAR(255),
    Workload TINYINT,
    AmountLearned TINYINT,
    Difficulty TINYINT,
    LectureQuality TINYINT,
    Management TINYINT,
    ReviewText VARCHAR(255),
    PRIMARY KEY(UserName, CourseNumber, ProfessorID),
    FOREIGN KEY(Username) REFERENCES Users(Username),
    FOREIGN KEY(CourseNumber) REFERENCES Courses(CourseNumber),
    FOREIGN KEY(ProfessorID) REFERENCES Professors(ProfessorID)
);
```

3. Used three real datasets we found online, and generated custom data for parts of the Users, GPA, Reviews, and Professors tables.

```
mysql> select count(CourseNumber) from Courses;
+----------------------+
| count(CourseNumber) |
+----------------------+
|                  610 |
+----------------------+
1 row in set (0.02 sec)

mysql> select count(ProfessorID) from Professors;
+--------------------+
| count(ProfessorID) |
+--------------------+
|              22643 |
+--------------------+
1 row in set (0.01 sec)

mysql> select count(GPAData) from GPA;
+----------------+
| count(GPAData) |
+----------------+
|           5798 |
+----------------+
1 row in set (0.01 sec)

mysql> select count(UsersName) from Users;
ERROR 1054 (42S22): Unknown column 'UsersName' in 'field list'
mysql> select count(UserName) from Users;
+-----------------+
| count(UserName) |
+-----------------+
|            1001 |
+-----------------+
1 row in set (0.00 sec)

mysql> select count(CourseNumber) from Reviews;
+----------------------+
| count(CourseNumber) |
+----------------------+
|                  100 |
+----------------------+
1 row in set (0.00 sec)
```

Advanced SQL Queries:

Query#1 (Elements: JOIN and GROUP BY):

SELECT Courses.CourseName, count(Reviews.CourseNumber) FROM Courses LEFT JOIN Reviews ON Courses.CourseNumber = Reviews.CourseNumber WHERE Courses.CreditHours > 3 GROUP BY Courses.CourseName ORDER BY count(Reviews.CourseNumber);

Result (First 15 Rows):

```
mysql> select Courses.CourseName, count(Reviews.CourseNumber) from Courses left join Reviews on Courses.CourseNumber = Reviews.CourseNumber where Courses.CreditHours > 3 group by Courses.CourseNam
e limit 15;
+--------------------------------------+---------------------------+
| CourseName                           | count(Reviews.CourseNumber) |
+--------------------------------------+---------------------------+
| Data Science Discovery               |                         1 |
| Introduction to Spanish Grammar      |                         0 |
| Modern Europe and the World - ACP    |                         0 |
| US History to 1877-ACP               |                         0 |
| Intermediate Chinese I               |                         0 |
| Animal Reproduction and Growth       |                         0 |
| Fundamental Organic Chem I           |                         0 |
| Elementary Wolof II                  |                         0 |
| Forensic Science                     |                         0 |
| Elementary Zulu II                   |                         0 |
| Organismal Biology                   |                         0 |
| Introduction to Mediated Communication |                       0 |
| Race and Cultural Diversity          |                         0 |
| Synoptic Weather Forecasting         |                         0 |
| Behavior of Domestic Animals         |                         0 |
+--------------------------------------+---------------------------+
15 rows in set (0.01 sec)
```

*The reason why there is only one course with a review, is because our review dataset (which we hand-generated) is relatively small (100 entries). It is one only two tables in our database which are < 1000 entries long.

EXPLAIN ANALYZE:

```
mysql> Explain analyze SELECT Courses.CourseName, count(Reviews.CourseNumber) FROM Courses LEFT JOIN Reviews ON Courses.CourseNumber = Reviews.CourseNumber WHERE Courses.CreditHours > 3 GROUP BY C
ourses.CourseName;
+------------------------------------------------------------------+
| EXPLAIN
                                                                   |
+------------------------------------------------------------------+
| -> Table scan on <temporary>  (actual time=1.677..1.693 rows=132 loops=1)
    -> Aggregate using temporary table  (actual time=1.675..1.675 rows=132 loops=1)
        -> Nested loop left join  (cost=197.13 rows=549) (actual time=0.896..1.464 rows=135 loops=1)
            -> Filter: (Courses.CreditHours > 3)  (cost=62.50 rows=203) (actual time=0.119..0.372 rows=132 loops=1)
                -> Table scan on Courses  (cost=62.50 rows=610) (actual time=0.109..0.311 rows=610 loops=1)
            -> Covering index lookup on Reviews using CourseNumber (CourseNumber=Courses.CourseNumber)  (cost=0.39 rows=3) (actual time=0.002..0.002 rows=0 loops=132)
 |
+------------------------------------------------------------------+
1 row in set (0.02 sec)
```

Query 2 (Elements: JOIN and UNION):

Select Distinct CourseName, GPA.GPAData, Courses.CreditHours, Reviews.Workload from ((Select Distinct CourseNumber from Courses where CreditHours > 3) union (select distinct CourseNumber from Reviews where Workload < 5)) as c join GPA where GPAData >=3 limit 15;

```
mysql> Select Distinct CourseName, GPA.GPAData from ((Select Distinct CourseNumber from Courses where CreditHours > 3) union (select distinct CourseNumber from Reviews where Workload < 5)) as c jo
in GPA where GPAData >= 3 limit 15;
+-------------------------------+---------+
| CourseName                    | GPAData |
+-------------------------------+---------+
| Special Topics in Aviation    |       3 |
| Intro BioControl Systems      |       4 |
| Elements of Syntax            |       3 |
| Collaboration and Teaming     |       4 |
| Navigating the Job Market     |       4 |
| The Balkans                   |       3 |
| Intro to Amer Indian Studies  |       4 |
| Intro to Computational Ling   |       4 |
| Intro to Human Development    |       4 |
| Assessment for Learning       |       4 |
| Negotiations                  |       3 |
| Theories of Measurement I     |       4 |
| Medieval Lit and Culture      |       3 |
| Health Policy: United States  |       4 |
| Human Biological Variation    |       4 |
+-------------------------------+---------+
15 rows in set (0.00 sec)
```

EXPLAIN ANALYZE:

```
| -> Limit: 15 row(s)  (cost=58296.68..58296.86 rows=15) (actual time=2.097..2.100 rows=15 loops=1)
    -> Table scan on <temporary>  (cost=58296.68..60096.96 rows=143824) (actual time=2.096..2.098 rows=15 loops=1)
        -> Temporary table with deduplication  (cost=58296.67..58296.67 rows=143824) (actual time=2.095..2.095 rows=15 loops=1)
            -> Limit table size: 15 unique row(s)
                -> Inner hash join (no condition)  (cost=43914.28 rows=143824) (actual time=0.817..1.000 rows=2185 loops=1)
                    -> Filter: (GPA.GPAData >= 3)  (cost=3.56 rows=1823) (actual time=0.051..0.058 rows=15 loops=1)
                        -> Table scan on GPA  (cost=3.56 rows=5471) (actual time=0.042..0.046 rows=15 loops=1)
                    -> Hash
                        -> Table scan on c  (cost=99.77..105.20 rows=237) (actual time=0.716..0.733 rows=156 loops=1)
                            -> Union materialize with deduplication  (cost=99.75..99.75 rows=237) (actual time=0.715..0.715 rows=156 loops=1)
                                -> Filter: (Courses.CreditHours > 3)  (cost=62.50 rows=203) (actual time=0.082..0.380 rows=132 loops=1)
                                    -> Table scan on Courses  (cost=62.50 rows=610) (actual time=0.069..0.324 rows=610 loops=1)
                                -> Group (no aggregates)  (cost=13.58 rows=33) (actual time=0.167..0.269 rows=27 loops=1)
                                    -> Filter: (Reviews.Workload < 5)  (cost=10.25 rows=33) (actual time=0.163..0.254 rows=57 loops=1)
                                        -> Index scan on Reviews using CourseNumber  (cost=10.25 rows=100) (actual time=0.161..0.245 rows=100 loops=1)
```

Query 3 (Elements: JOIN and GROUP BY): This query shows the Departments which have given out the most “A’s”:

SELECT Department, SUM(A_count) AS Cumulative_As

FROM Courses NATURAL JOIN GPA

GROUP BY Department

ORDER BY Cumulative_As DESC

LIMIT 15;

```
mysql> SELECT Department, SUM(A_count) AS Cumulative_As
    -> FROM Courses NATURAL JOIN GPA GROUP BY Department
    -> ORDER BY Cumulative_As DESC
    -> LIMIT 15;
+------------+---------------+
| Department | Cumulative_As |
+------------+---------------+
| BADM       |         79997 |
| CS         |         59720 |
| CHEM       |         53073 |
| KIN        |         45320 |
| CHLH       |         40951 |
| STAT       |         33846 |
| MATH       |         33764 |
| ECON       |         32386 |
| ACCY       |         31301 |
| FIN        |         29548 |
| PSYC       |         26262 |
| ECE        |         26153 |
| RST        |         24436 |
| FSHN       |         20526 |
| SHS        |         18461 |
+------------+---------------+
15 rows in set (0.02 sec)
```

```
| -> Limit: 15 row(s)  (actual time=15.376..15.378 rows=15 loops=1)
    -> Sort: Cumulative_As DESC, limit input to 15 row(s) per chunk  (actual time=15.374..15.375 rows=15 loops=1)
        -> Table scan on <temporary>  (actual time=15.290..15.316 rows=142 loops=1)
            -> Aggregate using temporary table  (actual time=15.287..15.287 rows=142 loops=1)
                -> Nested loop inner join  (cost=2047.15 rows=4537) (actual time=0.109..13.664 rows=2094 loops=1)
                    -> Table scan on Courses  (cost=459.20 rows=4537) (actual time=0.075..1.576 rows=4455 loops=1)
                    -> Single-row index lookup on GPA using PRIMARY (CourseNumber=Courses.CourseNumber)  (cost=0.25 rows=1) (actual time=0.003..0.003 rows=0 loops=4455)
```

Query 4 (Elements: JOIN and GROUP BY): This query shows the Professors which have given out the most “A’s”, along with their Rating and Last Ranked Term:

SELECT ProfessorName, Rating, LastRankedTerm, SUM(A_count) AS Cumulative_As
FROM Professors INNER JOIN GPA ON Professors.ProfessorName = GPA.Professor

GROUP BY ProfessorName HAVING Cumulative_As >= 500 ORDER BY Rating DESC LIMIT 15;

```
mysql> SELECT ProfessorName, Rating, LastRankedTerm, SUM(A_count) AS Cumulative_As
    -> FROM Professors INNER JOIN GPA ON Professors.ProfessorName = GPA.Professor
    -> GROUP BY ProfessorName
    -> HAVING Cumulative_As >= 500
    -> ORDER BY Rating DESC
    -> LIMIT 15;
+---------------+--------+----------------+---------------+
| ProfessorName | Rating | LastRankedTerm | Cumulative_As |
+---------------+--------+----------------+---------------+
| S MICHAEL     |     88 | sp2019         |          2487 |
| J PAUL        |     66 | sp2017         |           588 |
| B KELLY       |     55 | sp2011         |          6262 |
| S ALEXANDER   |     30 | sp2001         |          1185 |
| S THOMAS      |     30 | sp2020         |          8750 |
| Y CHEN        |     27 | sp2020         |           589 |
| H CHANG       |     19 | sp2017         |           757 |
| L FENG        |     18 | fa2019         |           844 |
| J THOMAS      |     16 | sp2020         |           705 |
| M MICHAEL     |     16 | su2019         |          9441 |
| B CLIFTON     |     14 | sp2016         |          2650 |
| T ALI         |     10 | sp2018         |          1368 |
| J RYAN        |      8 | sp2003         |           912 |
| S KELLY       |      8 | fa2017         |           803 |
| M DAVID       |      4 | sp2001         |          1463 |
+---------------+--------+----------------+---------------+
15 rows in set (0.01 sec)
```

```
| -> Limit: 15 row(s)  (actual time=8.878..8.881 rows=15 loops=1)
    -> Sort: Professors.Rating DESC, limit input to 15 row(s) per chunk  (actual time=8.877..8.878 rows=15 loops=1)
        -> Table scan on <temporary>  (actual time=8.815..8.828 rows=76 loops=1)
            -> Aggregate using temporary table  (actual time=8.811..8.811 rows=76 loops=1)
                -> Nested loop inner join  (cost=2063.75 rows=4575) (actual time=0.128..8.593 rows=170 loops=1)
                    -> Filter: (GPA.Professor is not null)  (cost=462.50 rows=4575) (actual time=0.092..1.711 rows=4575 loops=1)
                        -> Covering index scan on GPA using idx_A_count_prof  (cost=462.50 rows=4575) (actual time=0.089..1.364 rows=4575 loops=1)
                    -> Single-row index lookup on Professors using PRIMARY (ProfessorName=GPA.Professor)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=0 loops=4575)
|
```