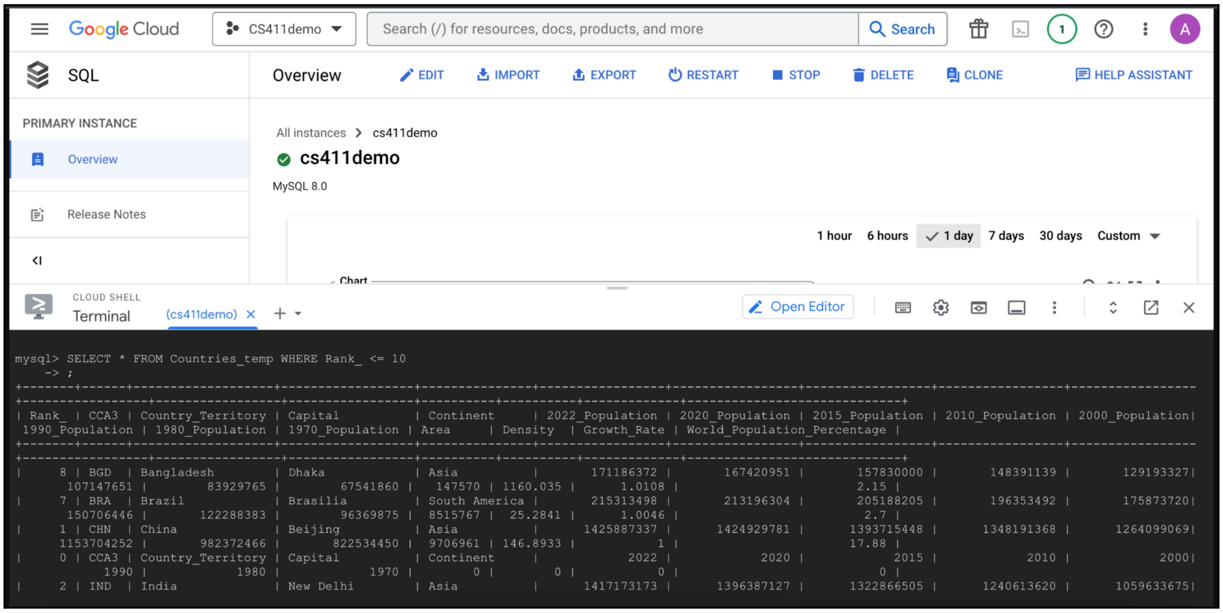
CS 411 - PT1 Stage 3

1. Screenshot of implementing database on GCP:



```
mysql> SELECT * FROM Countries_temp WHERE Rank_ <= 10
    -> ;
+-------+------+-------------------+-------------------+---------------+-----------------+-----------------+-----------------+-----------------+----------------
+-----------------+-----------------+-----------------+----------+----------+-------------+-----------------------------+
| Rank_ | CCA3 | Country_Territory | Capital           | Continent     | 2022_Population | 2020_Population | 2015_Population | 2010_Population | 2000_Population|
 1990_Population | 1980_Population | 1970_Population | Area     | Density  | Growth_Rate | World_Population_Percentage |
+-------+------+-------------------+-------------------+---------------+-----------------+-----------------+-----------------+-----------------+----------------
+-----------------+-----------------+-----------------+----------+----------+-------------+-----------------------------+
|     8 | BGD  | Bangladesh        | Dhaka             | Asia          |       171186372 |       167420951 |       157830000 |       148391139 |       129193327|
  107147651 |        83929765 |        67541860 |   147570 | 1160.035 |      1.0108 |                        2.15 |
|     7 | BRA  | Brazil            | Brasilia          | South America |       215313498 |       213196304 |       205188205 |       196353492 |       175873720|
  150706446 |       122288383 |        96369875 | 8515767  |  25.2841 |      1.0046 |                         2.7 |
|     1 | CHN  | China             | Beijing           | Asia          |      1425887337 |      1424929781 |      1393715448 |      1348191368 |      1264099069|
 1153704252 |       982372466 |       822534450 | 9706961  | 146.8933 |           1 |                       17.88 |
|     0 | CCA3 | Country_Territory | Capital           | Continent     |            2022 |            2020 |            2015 |            2010 |            2000|
       1990 |            1980 |            1970 |        0 |        0 |           0 |                           0 |
|     2 | IND  | India             | New Delhi         | Asia          |      1417173173 |      1396387127 |      1322866505 |      1240613620 |      1059633675|
```

2. DDL command for our four tables:

CREATE TABLE Coaches(name VARCHAR(255) PRIMARY KEY NOT NULL, country_CCA3 VARCHAR(30) , discipline_name VARCHAR(30), event VARCHAR(30), FOREIGN KEY (country_CCA3) REFERENCES Country(CCA3),FOREIGN KEY (discipline_name) REFERENCES Discipline(Name));

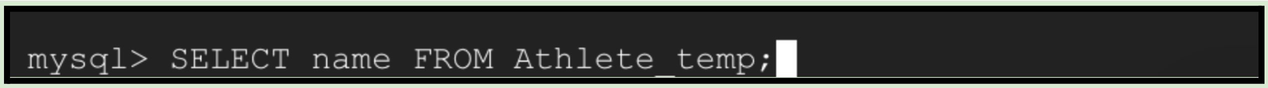
CREATE TABLE ATHLETE (name VARCHAR(255) NOT NULL, country_CCA3 VARCHAR(30), discipline_name VARCHAR(30), PRIMARY KEY(name), FOREIGN KEY (country_CCA3) REFERENCES Country(CCA3), FOREIGN KEY(discipline_name) REFERENCES Discipline(Name));

CREATE TABLE Country (CCA3 VARCHAR(255) NOT NULL, rank_of_population INT, continent VARCHAR(255), population_2020 INT, population_2022 INT, rank_of_medals INT, num_gold_medals INT, num_silver_medals INT, num_bronze_medals INT, num_total_medals INT, PRIMARY KEY (CCA3));

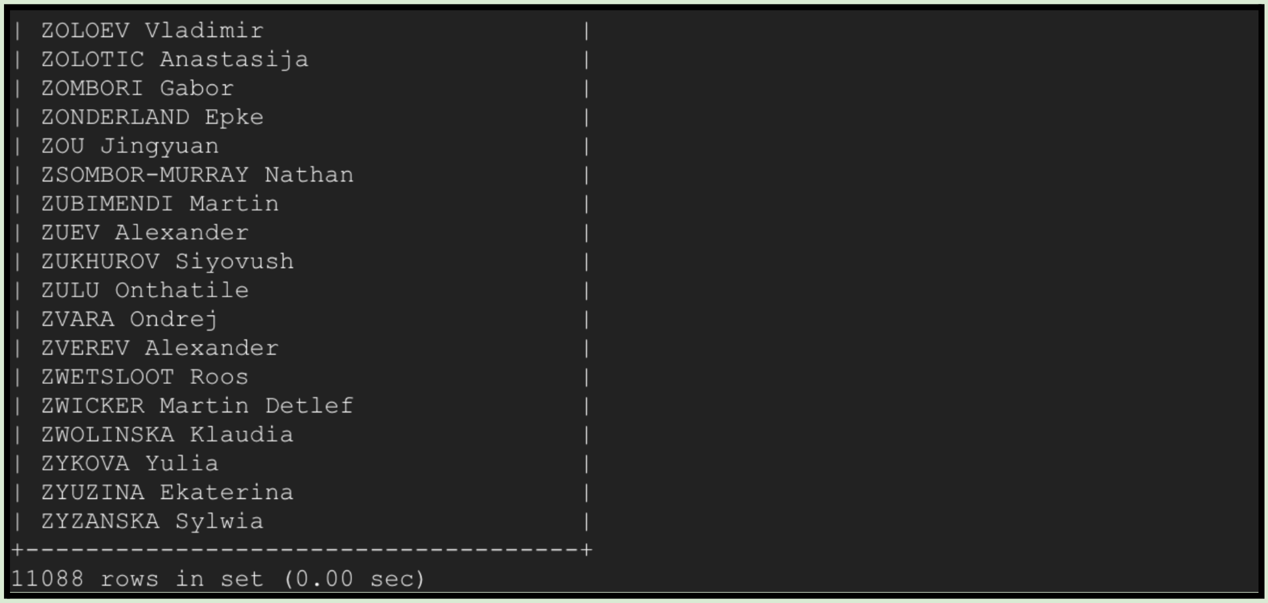
CREATE TABLE Discipline (Name VARCHAR(255) NOT NULL, Male_amt int, Female_amt int, Total_mum int, PRIMARY KEY (Name));

3. Inserting at least 1000 rows in the tables:

Athlete Table:

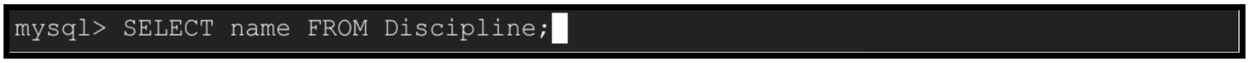


```
mysql> SELECT name FROM Athlete_temp;
```

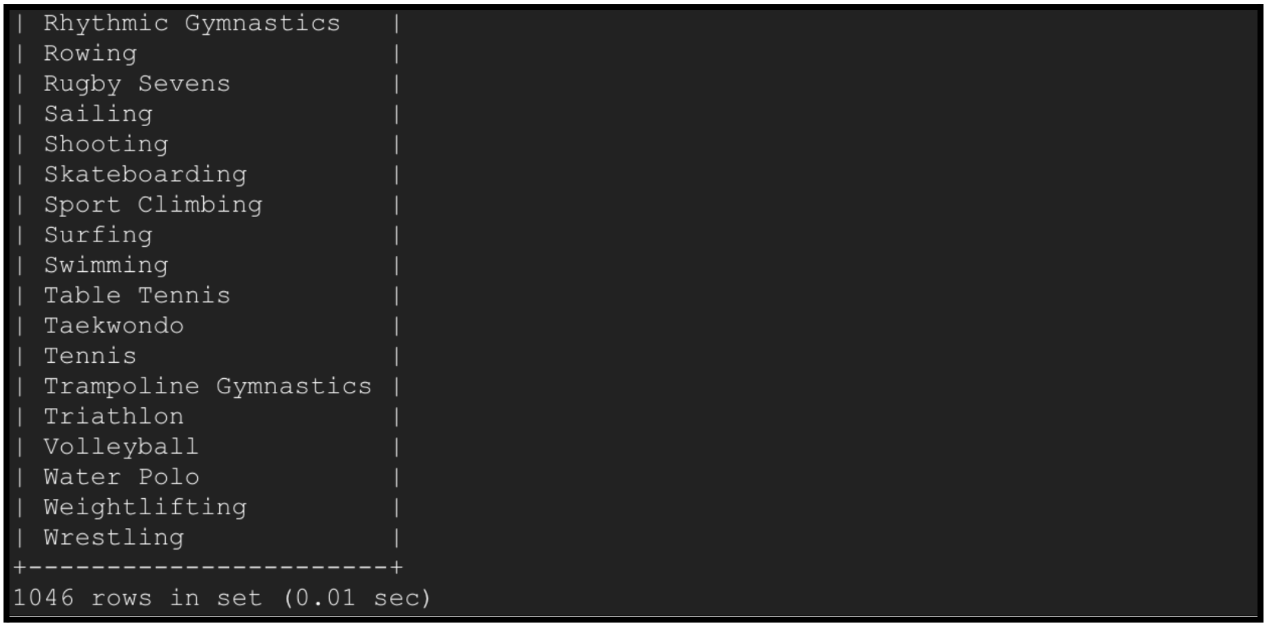


```
| ZOLOEV Vladimir                      |
| ZOLOTIC Anastasija                   |
| ZOMBORI Gabor                        |
| ZONDERLAND Epke                      |
| ZOU Jingyuan                         |
| ZSOMBOR-MURRAY Nathan                |
| ZUBIMENDI Martin                     |
| ZUEV Alexander                       |
| ZUKHUROV Siyovush                    |
| ZULU Onthatile                       |
| ZVARA Ondrej                         |
| ZVEREV Alexander                     |
| ZWETSLOOT Roos                       |
| ZWICKER Martin Detlef                |
| ZWOLINSKA Klaudia                    |
| ZYKOVA Yulia                         |
| ZYUZINA Ekaterina                    |
| ZYZANSKA Sylwia                      |
+--------------------------------------+
11088 rows in set (0.00 sec)
```

Discipline Table:

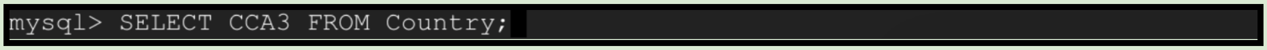


```
mysql> SELECT name FROM Discipline;
```

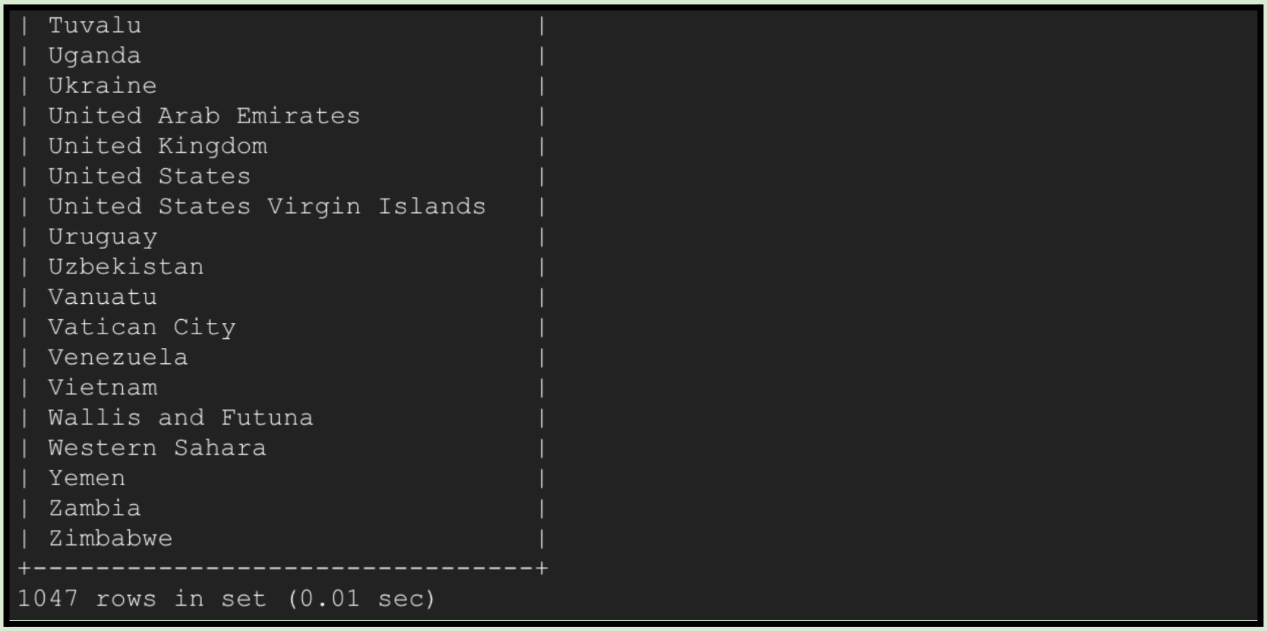


```
| Rhythmic Gymnastics   |
| Rowing                |
| Rugby Sevens          |
| Sailing               |
| Shooting              |
| Skateboarding         |
| Sport Climbing        |
| Surfing               |
| Swimming              |
| Table Tennis          |
| Taekwondo             |
| Tennis                |
| Trampoline Gymnastics |
| Triathlon             |
| Volleyball            |
| Water Polo            |
| Weightlifting         |
| Wrestling             |
+-----------------------+
1046 rows in set (0.01 sec)
```

Country Table:



```
mysql> SELECT CCA3 FROM Country;
```

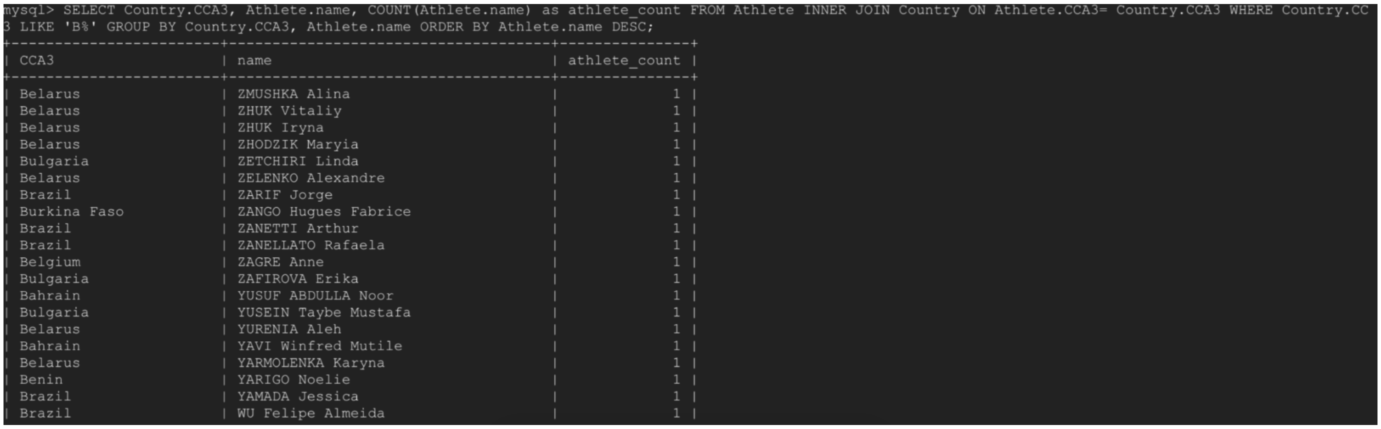


```
| Tuvalu                          |
| Uganda                          |
| Ukraine                         |
| United Arab Emirates            |
| United Kingdom                  |
| United States                   |
| United States Virgin Islands    |
| Uruguay                         |
| Uzbekistan                      |
| Vanuatu                         |
| Vatican City                    |
| Venezuela                       |
| Vietnam                         |
| Wallis and Futuna               |
| Western Sahara                  |
| Yemen                           |
| Zambia                          |
| Zimbabwe                        |
+---------------------------------+
1047 rows in set (0.01 sec)
```

4. Advanced Queries Screenshot

Advanced Query #1: Finds the names of athletes from countries that start with the letter B. Returns a table athlete name, country, and athlete count. Uses inner join operation on country and athlete based on the country CCA3 and groups by country name and athlete name.

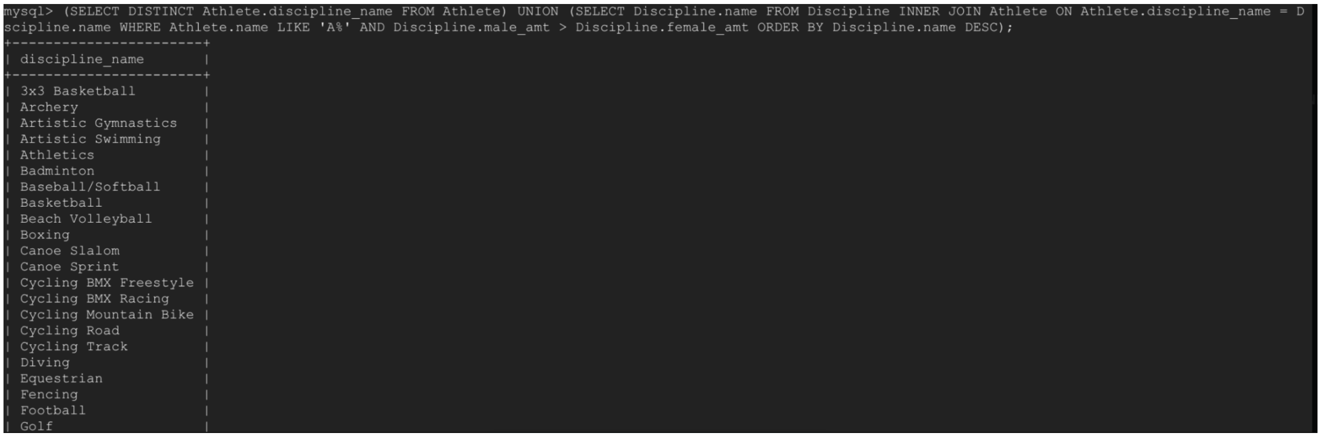
SELECT Country.CCA3, COUNT(Athlete.name) as athlete_count
FROM Athlete INNER JOIN Country ON Athlete.CCA3= Country.CCA3
WHERE Country.CCA3 LIKE 'B%'
GROUP BY Country.CCA3, Athlete.name
ORDER BY Athlete.name DESC;



```
mysql> SELECT Country.CCA3, Athlete.name, COUNT(Athlete.name) as athlete_count FROM Athlete INNER JOIN Country ON Athlete.CCA3= Country.CCA3 WHERE Country.CC
3 LIKE 'B%' GROUP BY Country.CCA3, Athlete.name ORDER BY Athlete.name DESC;
+------------------------+-------------------------------------+---------------+
| CCA3                   | name                                | athlete_count |
+------------------------+-------------------------------------+---------------+
| Belarus                | ZMUSHKA Alina                       |             1 |
| Belarus                | ZHUK Vitaliy                        |             1 |
| Belarus                | ZHUK Iryna                          |             1 |
| Belarus                | ZHODZIK Maryia                      |             1 |
| Bulgaria               | ZETCHIRI Linda                      |             1 |
| Belarus                | ZELENKO Alexandre                   |             1 |
| Brazil                 | ZARIF Jorge                         |             1 |
| Burkina Faso           | ZANGO Hugues Fabrice                |             1 |
| Brazil                 | ZANETTI Arthur                      |             1 |
| Brazil                 | ZANELLATO Rafaela                   |             1 |
| Belgium                | ZAGRE Anne                          |             1 |
| Bulgaria               | ZAFIROVA Erika                      |             1 |
| Bahrain                | YUSUF ABDULLA Noor                  |             1 |
| Bulgaria               | YUSEIN Taybe Mustafa                |             1 |
| Belarus                | YURENIA Aleh                        |             1 |
| Bahrain                | YAVI Winfred Mutile                 |             1 |
| Belarus                | YARMOLENKA Karyna                   |             1 |
| Benin                  | YARIGO Noelie                       |             1 |
| Brazil                 | YAMADA Jessica                      |             1 |
| Brazil                 | WU Felipe Almeida                   |             1 |
```

Advanced Query #2: Finds all discipline names that have an athlete name starting with the letter A and where the discipline has a higher male count than female count

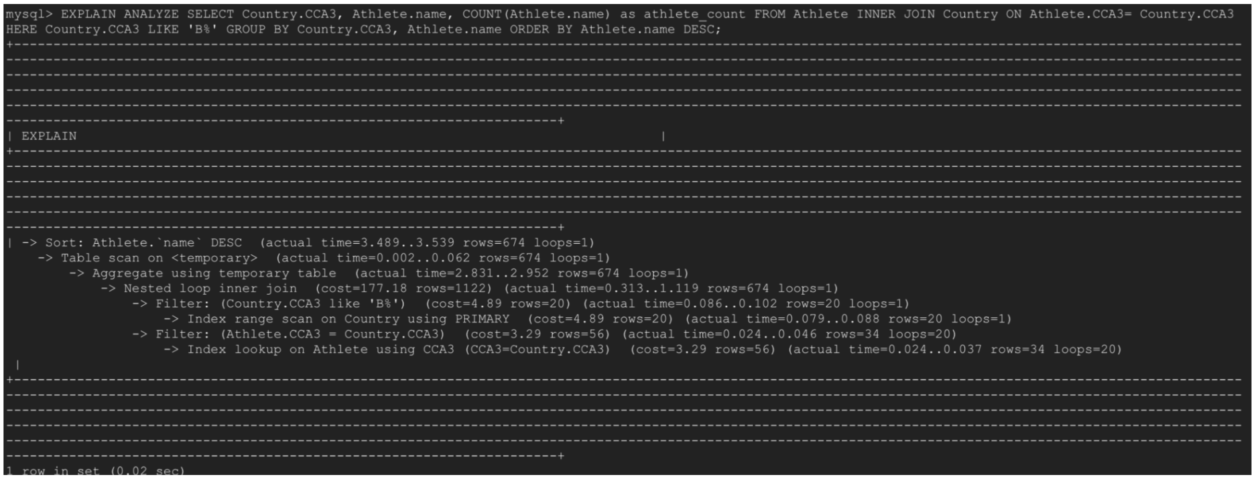
(**SELECT DISTINCT** Athlete.discipline_name FROM Athlete)
UNION
(**SELECT** Discipline.name
FROM Discipline INNER JOIN Athlete ON Athlete.discipline_name = Discipline.name
WHERE Athlete.name LIKE 'A%' AND Discipline.male_amt > Discipline.female_amt
ORDER BY Discipline.name DESC)



```
mysql> (SELECT DISTINCT Athlete.discipline_name FROM Athlete) UNION (SELECT Discipline.name FROM Discipline INNER JOIN Athlete ON Athlete.discipline_name = D
scipline.name WHERE Athlete.name LIKE 'A%' AND Discipline.male_amt > Discipline.female_amt ORDER BY Discipline.name DESC);
+-----------------------+
| discipline_name       |
+-----------------------+
| 3x3 Basketball        |
| Archery               |
| Artistic Gymnastics   |
| Artistic Swimming     |
| Athletics             |
| Badminton             |
| Baseball/Softball     |
| Basketball            |
| Beach Volleyball      |
| Boxing                |
| Canoe Slalom          |
| Canoe Sprint          |
| Cycling BMX Freestyle |
| Cycling BMX Racing    |
| Cycling Mountain Bike |
| Cycling Road          |
| Cycling Track         |
| Diving                |
| Equestrian            |
| Fencing               |
| Football              |
| Golf                  |
```

5. Indexing Analysis

QUERY #1: Explain Analyze on first query

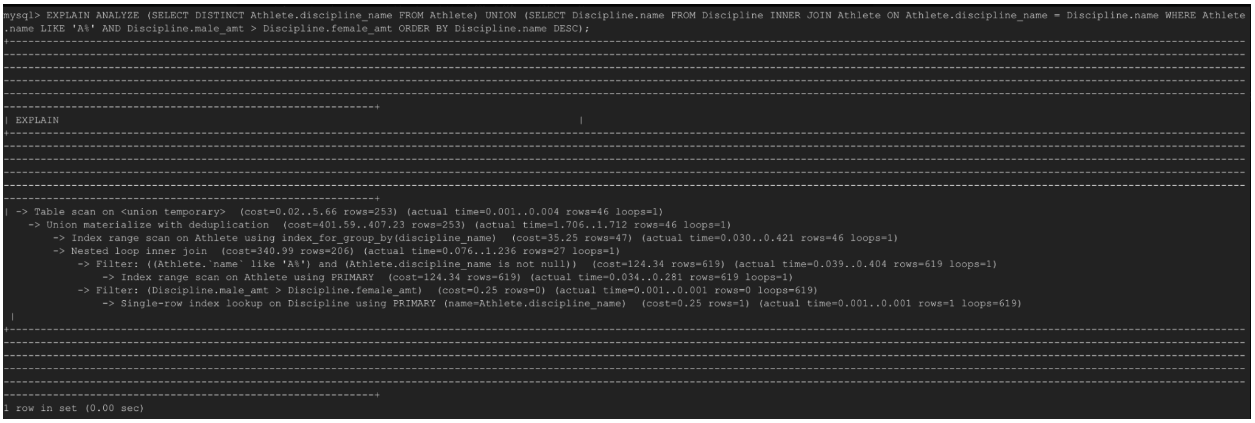


```
mysql> EXPLAIN ANALYZE SELECT Country.CCA3, Athlete.name, COUNT(Athlete.name) as athlete_count FROM Athlete INNER JOIN Country ON Athlete.CCA3= Country.CCA3
HERE Country.CCA3 LIKE 'B%' GROUP BY Country.CCA3, Athlete.name ORDER BY Athlete.name DESC;
+-----------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------+
| EXPLAIN                                                                                  |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------+
| -> Sort: Athlete.`name` DESC  (actual time=3.489..3.539 rows=674 loops=1)
    -> Table scan on <temporary>  (actual time=0.002..0.062 rows=674 loops=1)
        -> Aggregate using temporary table  (actual time=2.831..2.952 rows=674 loops=1)
            -> Nested loop inner join  (cost=177.18 rows=1122) (actual time=0.313..1.119 rows=674 loops=1)
                -> Filter: (Country.CCA3 like 'B%')  (cost=4.89 rows=20) (actual time=0.086..0.102 rows=20 loops=1)
                    -> Index range scan on Country using PRIMARY  (cost=4.89 rows=20) (actual time=0.079..0.088 rows=20 loops=1)
                -> Filter: (Athlete.CCA3 = Country.CCA3)  (cost=3.29 rows=56) (actual time=0.024..0.046 rows=34 loops=20)
                    -> Index lookup on Athlete using CCA3 (CCA3=Country.CCA3)  (cost=3.29 rows=56) (actual time=0.024..0.037 rows=34 loops=20)
 |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------+
1 row in set (0.02 sec)
```

According to the screenshot above, the EXPLAIN ANALYZE tool for the default indexing showed that the first row took 3.489 ms, the second row took 0.002 ms, and so on and so forth. The total number of rows were 56 and our cost was 3.29 ms. Thus, we decided to index using the Country CCA3 attribute and we saw that it was much more efficient and optimal for this query. It took an average of 2.6ms to read the first row, 0.001 ms for the second row, and so on. The total number of rows were reduced to 46 and our cost was 1.9 ms. We then tried to index on the attribute Athlete Name but found that this was slower than both the default indexing and our

Country CCA3 attribute index. It took 4.20 ms to read the first row, 0.006 ms to read the second row, etc. The overall cost was 5.0 ms and the number of rows was 78. We chose to index on Country CCA3 because it was more effective and ideal for our first advanced query. Furthermore, the number of rows that need to be scanned was significantly reduced when indexing using the Country CCA3 attribute.

QUERY #2: Explain Analyze on second query



```
mysql> EXPLAIN ANALYZE (SELECT DISTINCT Athlete.discipline_name FROM Athlete) UNION (SELECT Discipline.name FROM Discipline INNER JOIN Athlete ON Athlete.discipline_name = Discipline.name WHERE Athlete
.name LIKE 'A%' AND Discipline.male_amt > Discipline.female_amt ORDER BY Discipline.name DESC);
+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------+
| EXPLAIN                                                                                      |
+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------+
| -> Table scan on <union temporary>  (cost=0.02..5.66 rows=253) (actual time=0.001..0.004 rows=46 loops=1)
    -> Union materialize with deduplication  (cost=401.59..407.23 rows=253) (actual time=1.706..1.712 rows=46 loops=1)
        -> Index range scan on Athlete using index_for_group_by(discipline_name)  (cost=35.25 rows=47) (actual time=0.030..0.421 rows=46 loops=1)
        -> Nested loop inner join  (cost=340.99 rows=206) (actual time=0.076..1.236 rows=27 loops=1)
            -> Filter: ((Athlete.`name` like 'A%') and (Athlete.discipline_name is not null))  (cost=124.34 rows=619) (actual time=0.039..0.404 rows=619 loops=1)
                -> Index range scan on Athlete using PRIMARY  (cost=124.34 rows=619) (actual time=0.034..0.281 rows=619 loops=1)
            -> Filter: (Discipline.male_amt > Discipline.female_amt)  (cost=0.25 rows=0) (actual time=0.001..0.001 rows=0 loops=619)
                -> Single-row index lookup on Discipline using PRIMARY (name=Athlete.discipline_name)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=619)
 |
+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------+
1 row in set (0.00 sec)
```

According to the screenshot above, the EXPLAIN ANALYZE tool for the default indexing showed that the first row took 124.3 ms, the second row took 1.706 ms, and so on and so forth. The total number of rows were 56 and our cost was 3.29 ms. Thus, we decided to index using the Discipline Name attribute and we saw that it was much more efficient and optimal for this query. It took an average of 0.02 ms to read the first row, 0.001 ms for the second row, and so on. The total number of rows were reduced to 46 and our cost was 1.9 ms. We then tried to index on the attribute Athlete Name but found that this was slower than both the default indexing and our Discipline Name attribute index. It took 4.20 ms to read the first row, 0.006 ms to read the second row, etc. The overall cost was 5.0 ms and the number of rows was 78. We chose to index on Discipline Name because it was more effective and ideal for our first advanced query. Furthermore, the number of rows that need to be scanned was significantly reduced when indexing using the Discipline Name attribute.