

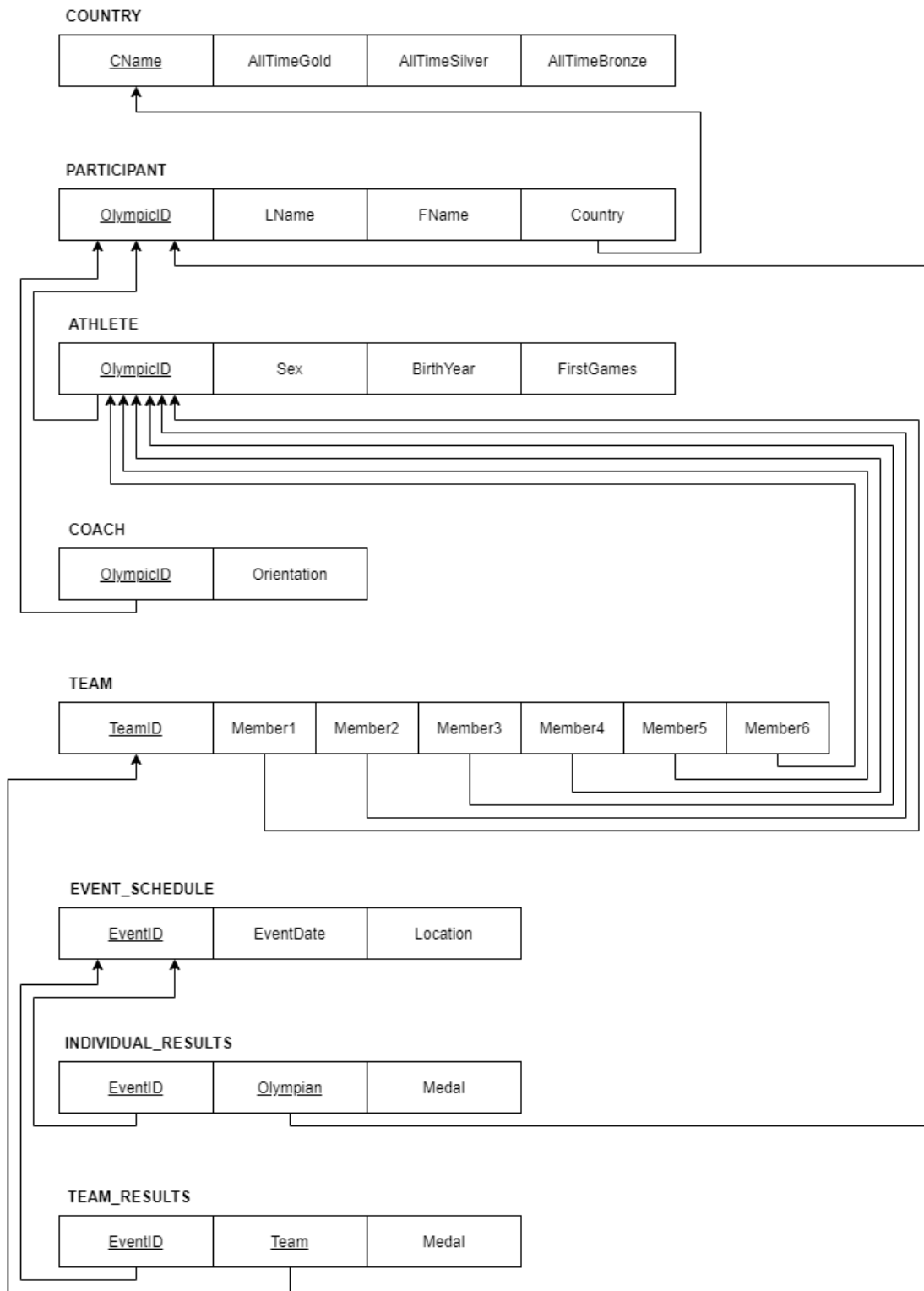
# ENSF 608 – Fall 2021

Lab 3 – Friday, November 12

**Student Name:** Bhavyai Gupta

**Submission date:** November 12, 2021

## Question 1 - Relational Data Model



## Question 2 - SQL Script

```
USE OLYMPICARCHERY;

-- QUESTION 1
SELECT      P.FName, P.LName, P.Country
FROM        PARTICIPANT P
RIGHT JOIN  ATHLETE A ON P.OlympicID = A.OlympicID
ORDER BY   P.Country, P.FName, P.LName;

-- QUESTION 2
SELECT      P.FName, P.LName, P.Country
FROM        PARTICIPANT P
RIGHT JOIN  COACH C ON P.OlympicID = C.OlympicID
WHERE       C.Orientation='Pending'
ORDER BY   P.Country, P.FName, P.LName;

-- QUESTION 3
SELECT      P.Country, COUNT(*) "Athletes_Count"
FROM        PARTICIPANT P
RIGHT JOIN  ATHLETE A ON P.OlympicID = A.OlympicID
GROUP BY   P.Country
ORDER BY   P.Country;

-- QUESTION 4
SELECT      P.OlympicID, A.BirthYear
FROM        PARTICIPANT P
LEFT JOIN   ATHLETE A ON P.OlympicID = A.OlympicID
ORDER BY   A.BirthYear;

-- QUESTION 5
SELECT      P.Country
FROM        PARTICIPANT P
RIGHT JOIN  ATHLETE A ON P.OlympicID = A.OlympicID
GROUP BY   P.Country
HAVING      COUNT(*) > 1
ORDER BY   P.Country;

-- QUESTION 6
```

```

SELECT      DISTINCT P.FName, P.LName
FROM        INDIVIDUAL_RESULTS IR
LEFT JOIN   ATHLETE A ON IR.Olympian = A.OlympicID
LEFT JOIN   PARTICIPANT P ON IR.Olympian = P.OlympicID
ORDER BY    P.FName, P.LName;

-- QUESTION 7
SELECT      C5.CName Country
FROM        (SELECT C.CName, (C.AllTimeGold + C.AllTimeSilver + C.AllTimeBronze)
AS Tally FROM COUNTRY C) C5
WHERE       C5.Tally >=5;

-- QUESTION 8
SELECT      IRTR.Country, COUNT(*) "Archery_Medals"
FROM        (
              (SELECT P.Country FROM INDIVIDUAL_RESULTS IR LEFT JOIN
PARTICIPANT P ON IR.Olympian = P.OlympicID) UNION ALL
              (SELECT P.Country FROM TEAM_RESULTS TR LEFT JOIN TEAM T ON
TR.Team = T.TeamID LEFT JOIN PARTICIPANT P ON T.Member1 = P.OlympicId)
            ) IRTR
GROUP BY    IRTR.Country
ORDER BY    IRTR.Country;

-- QUESTION 9
SELECT      P.FName, P.LName
FROM        PARTICIPANT P
RIGHT JOIN  ATHLETE A ON    P.OlympicID = A.OlympicID
WHERE       A.FirstGames = 'Tokyo 2020';

-- QUESTION 10
SELECT      P.FName, P.LName
FROM        (
              (SELECT * FROM ATHLETE A WHERE A.BirthYear=(SELECT MAX(BirthYear)
FROM ATHLETE)) UNION ALL
              (SELECT * FROM ATHLETE A WHERE A.BirthYear=(SELECT MIN(BirthYear)
FROM ATHLETE))
            ) MM
LEFT JOIN   PARTICIPANT P ON MM.OlympicID = P.OlympicID;

-- QUESTION 11
CREATE OR REPLACE VIEW TEAM_ATHLETES AS

```

```

SELECT      P.FName, P.LName, A.BirthYear
FROM        (SELECT DISTINCT AM.MemberID
              FROM      (
                        SELECT T.Member1 "MemberID" FROM TEAM T UNION ALL
                        SELECT T.Member2 "MemberID" FROM TEAM T UNION ALL
                        SELECT T.Member3 "MemberID" FROM TEAM T UNION ALL
                        SELECT T.Member4 "MemberID" FROM TEAM T UNION ALL
                        SELECT T.Member5 "MemberID" FROM TEAM T UNION ALL
                        SELECT T.Member6 "MemberID" FROM TEAM T
                        ) AS AM
              WHERE     AM.MemberID IS NOT NULL
              ) AMID
LEFT JOIN    ATHLETE A ON AMID.MemberID = A.OlympicID
LEFT JOIN    PARTICIPANT P ON P.OlympicID = AMID.MemberID
ORDER BY     A.BirthYear DESC;

-- QUESTION 12
CREATE TABLE INDIVID_W AS
SELECT DISTINCT * FROM
(
    (SELECT      ES.EventDate, ES.Location "EventVenue", P.LName "LastName",
P.Country
    FROM          EVENT_SCHEDULE ES
    INNER JOIN    INDIVIDUAL_RESULTS IR ON IR.EventID = ES.EventID
    LEFT JOIN     PARTICIPANT P ON P.OlympicID = IR.Olympian)
UNION ALL
    (SELECT      ES.EventDate, ES.Location "EventVenue", P.LName "LastName",
P.Country
    FROM          EVENT_SCHEDULE ES
    INNER JOIN    TEAM_RESULTS TR ON TR.EventID = ES.EventID
    LEFT JOIN     (
                SELECT *
                FROM      (
                        SELECT T.Member1 "MemberID", T.TeamID FROM TEAM T
UNION ALL
                        SELECT T.Member2 "MemberID", T.TeamID FROM TEAM T
UNION ALL
                        SELECT T.Member3 "MemberID", T.TeamID FROM TEAM T
UNION ALL
                        SELECT T.Member4 "MemberID", T.TeamID FROM TEAM T
UNION ALL
                        SELECT T.Member5 "MemberID", T.TeamID FROM TEAM T
UNION ALL
                        SELECT T.Member6 "MemberID", T.TeamID FROM TEAM T
                )
            )
    )
)

```

```

        ) AS AM
        WHERE AM.MemberID IS NOT NULL
    ) AMI ON AMI.TeamID = TR.Team
    LEFT JOIN PARTICIPANT P ON P.OlympicID = AMI.MemberID)
) AK
WHERE AK.EventDate = 'July 30'
ORDER BY AK.LastName;

```

```
SELECT * from INDIVID_W;
```

-- QUESTION 13

/\*

The command wont execute and throw error.

The attribute COACH.OlympicID is a foreign key referencing to PARTICIPANT.OlympicID. So, according to referential integrity constraint, value of FK must be present as PK in the PARTICIPANT table, or else FK must be null. But, in our query, we pass FK as 'T2020\_046', which is not present in PARTICIPANT table, hence error.

\*/

-- QUESTION 14

/\*

The command wont execute and throw error.

We are trying to delete a row from table PARTICIPANT whose PK is referenced by another row in the table ATHLETE. If that row is deleted, it would be in violation of referential integrity constraint, since the FK in table ATHLETE cannot point to a non-existing PK.

\*/

-- QUESTION 15

/\*

One possible constraint for the TEAM table could be to restrict the member of every TEAM by their age. For example, in every TEAM all the members must belong to age bucket [25, 35]. This constraint would involve processing BirthYear from ATHLETE table.

Another possible constraint for the TEAM table could be to have atleast one experienced player and atleast one fresh player. Experienced player is the one who is not competing in the Olympic games for the first time. This constraint would involve processing FirstGames from ATHLETE table.

\*/