BUILD: Project I / Design & Implement a Relational Database

Team: Orkhan Dadashov, Meilin Liu

1. Business Requirement

Our client is a sport event statistics committee who would like to have a database with information of Olympic Games during the past 120 years. Information they are interested in includes how many athletes participated in a game, how many and teams attended in a game, the number of medals for each team in a game, the athletes' information like their age, weight, and height, the number of athletes in an event, how many athletes have participated in multiple Olympic Games, which athlete won the most medals in the history of Olympic Games in a certain event, the general trend of athlete's body information along the years, as well as types of sports in different Olympic Games.

Nouns (In Red):

athletes
teams
game
medals
age
weight
height
event
types
sports
body information

Verbs (In Blue):

participated
attended
won

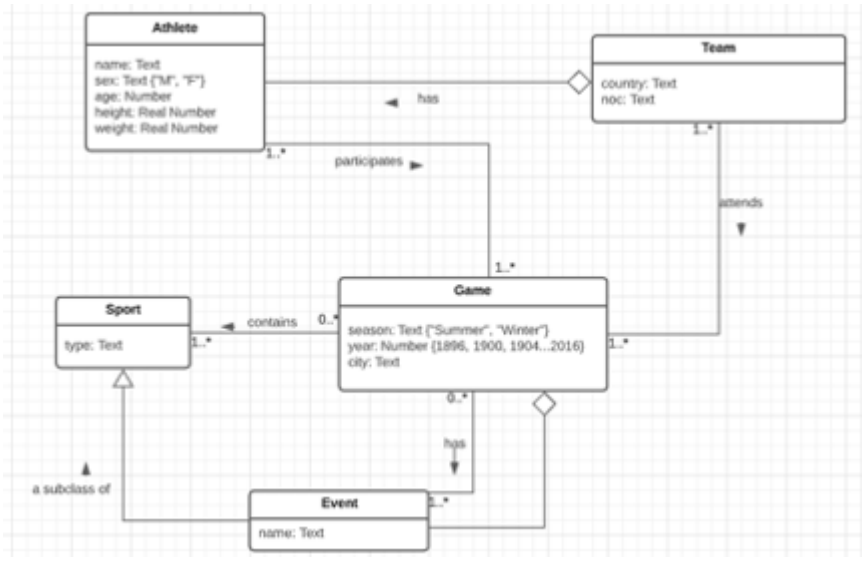
Business Rules

- A team could participate in multiple Olympic Games
- A team must participate in 1 Game to be recorded in the database.
- A team attending a Game must have at least 1 athlete.
- An athlete must participate in 1 Olympic Game to be recorded in the database.

- An athlete could participate in multiple events in an Olympic Games.
- An athlete could win several medals in 1 Game.
- An athlete could participate in Olympic Games for different teams, for instance Mary could participated in 1992 Summer with Team US and with Team Japan in 1996 Summer.
- A Sport type could have multiple events
- An event must belong to a Sport type.
- A Game must have multiple teams participating.

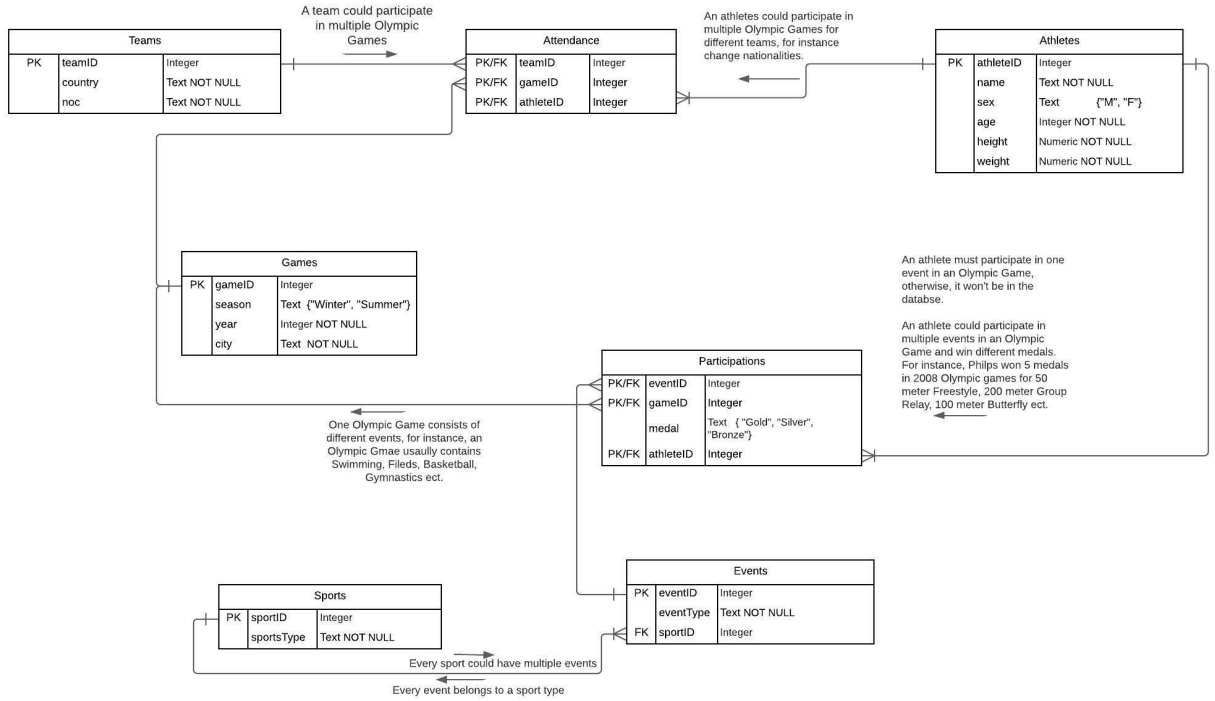
2. Conceptual Model in UML

https://lucid.app/lucidchart/47f13deb-0c7c-49cc-9430-6288f6ab24e9/edit?invitationId=inv_6602d40b-2285-42b7-b5b8-753d12cb2c87&page=sAIlboIWdedJ#



3. Logical Model in ERD

https://lucid.app/lucidchart/47f13deb-0c7c-49cc-9430-6288f6ab24e9/edit?viewport_loc=-596%2C4%2C3669%2C1955%2CRkZlGXn.R~cp&invitationId=inv_6602d40b-2285-42b7-b5b8-753d12cb2c87



4. Relational Schema

Teams{teamID, country, noc}
Attendance{teamID, gameID, athleteID}
Athletes{athleteID, name, sex, age, height, weight}
Participations{eventID, gameID, athleteID, medal}
Events{eventID, eventType, sportID}
Sports{sportID, sportsType}
Games{gameID, season, year, city}

Definition: A relation schema R is in **Boyce-Codd normal form (BCNF)** if, for every FD $X \rightarrow A$ in F, where X is the subset of the attributes of R, and A is an attribute of R, one of the following statements holds

⇒ $X \rightarrow Y$ is a trivial FD, that is, $Y \subseteq X$.
⇒ X is a superkey.

In simple terms, it can be stated as
A relation schema R is in **BCNF** if and only if every non-trivial FD has a candidate key as its determinant.

In the Relation Schema Teams, teamID is unique to every country, and the attribute noc is fully functional dependent on the key {teamID}, therefore, it's in BCNF.

In Attendance, since all the attributes are a subset of primary key, all the FDs are trivial, therefore, it's in BCNF.

In Athletes, an athlete's name, sex, age, height, weight is all determined by athleteID, and athleteID is primary key, which is a minimal super key, so it's in BCNF.

In Participations, an athlete's participation in an event in a game can determine the result of the medal. {eventID, gameID, athleteID} -> medal and the primary key is composite key with eventID, gameID, athleteID. Therefore, it is in BCNF.

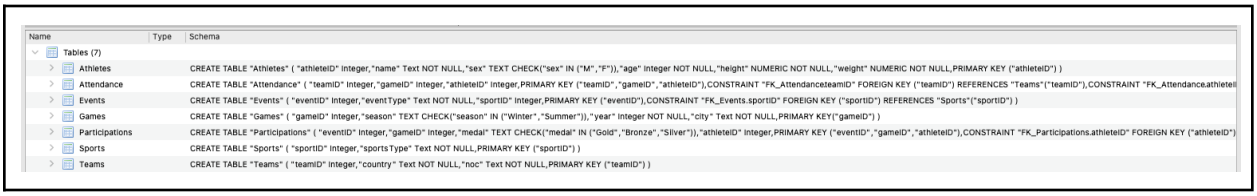
In Events, an eventID uniquely determines the eventType, and an event could only belong to one sport type, which is represented as sportID in this schema. Therefore, it is in BCNF.

In Sports, sportID determines sportsType. No other attributes are involved.

In Games, the season and year determines the city where a game held. {season, year} -> city. Season and year are superkeys. It's in BCNF.

5. Show that the tables were created and conform to the constraints through screen shots or other means.

```
CREATE TABLE IF NOT EXISTS "Teams" (
    "teamID"    Integer,
    "country"   Text NOT NULL,
    "noc"   Text NOT NULL,
    PRIMARY KEY("teamID")
);
CREATE TABLE IF NOT EXISTS "Athletes" (
    "athleteID" Integer,
    "name"  Text NOT NULL,
    "sex"   TEXT CHECK("sex" IN ("M", "F")),
    "age"   Integer NOT NULL,
    "height"    NUMERIC NOT NULL,
    "weight"    NUMERIC NOT NULL,
    PRIMARY KEY("athleteID")
);
CREATE TABLE IF NOT EXISTS "Sports" (
    "sportID"   Integer,
    "sportsType"    Text NOT NULL,
    PRIMARY KEY("sportID")
);
CREATE TABLE IF NOT EXISTS "Games" (
    "gameID"    Integer,
    "season"    TEXT CHECK("season" IN ("Winter", "Summer")),
    "year"  Integer NOT NULL,
    "city"  Text NOT NULL,
    PRIMARY KEY("gameID")
);
CREATE TABLE IF NOT EXISTS "Events" (
    "eventID"   Integer,
    "eventType" Text NOT NULL,
    "sportID"   Integer,
    CONSTRAINT "FK_Events.sportID" FOREIGN KEY("sportID") REFERENCES "Sports"("sportID") ON DELETE CASCADE,
    PRIMARY KEY("eventID")
);
CREATE TABLE IF NOT EXISTS "Participations" (
    "eventID"   Integer,
    "gameID"    Integer,
    "medal" TEXT CHECK("medal" IN ("Gold", "Bronze", "Silver")),
    "athleteID" Integer,
    CONSTRAINT "FK_Participations.athleteID" FOREIGN KEY("athleteID") REFERENCES "Athletes"("athleteID") ON DELETE
CASCADE,
    CONSTRAINT "FK_Participations.gameID" FOREIGN KEY("gameID") REFERENCES "Games"("gameID"),
    CONSTRAINT "FK_Participations.eventID" FOREIGN KEY("eventID") REFERENCES "Events"("eventID"),
    PRIMARY KEY("eventID","gameID","athleteID")
);
CREATE TABLE IF NOT EXISTS "Attendance" (
    "teamID"    Integer,
    "gameID"    Integer,
    "athleteID" Integer,
    CONSTRAINT "FK_Attendance.teamID" FOREIGN KEY("teamID") REFERENCES "Teams"("teamID"),
    CONSTRAINT "FK_Attendance.athleteID" FOREIGN KEY("athleteID") REFERENCES "Athletes"("athleteID") ON DELETE
CASCADE,
    CONSTRAINT "FK_Attendance.gameID" FOREIGN KEY("gameID") REFERENCES "Games"("gameID"),
    PRIMARY KEY("teamID","gameID","athleteID")
);
```



Name	Type	Schema
Tables (7)		
Athletes		CREATE TABLE "Athletes" ("athleteID" Integer,"name" Text NOT NULL,"sex" TEXT CHECK("sex" IN ("M","F")),"age" Integer NOT NULL,"height" NUMERIC NOT NULL,"weight" NUMERIC NOT NULL,PRIMARY KEY ("athleteID"))
Attendance		CREATE TABLE "Attendance" ("teamID" Integer,"gameID" Integer,"athleteID" Integer,PRIMARY KEY ("teamID","gameID","athleteID"),CONSTRAINT "FK_AttendanceteamID" FOREIGN KEY ("teamID") REFERENCES "Teams"("teamID"),CONSTRAINT "FK_Attendance.athlete
Events		CREATE TABLE "Events" ("eventID" Integer,"eventType" Text NOT NULL,"sportID" Integer,PRIMARY KEY ("eventID"),CONSTRAINT "FK_Events.sportID" FOREIGN KEY ("sportID") REFERENCES "Sports"("sportID"))
Games		CREATE TABLE "Games" ("gameID" Integer,"season" TEXT CHECK("season" IN ("Winter","Summer")),"year" Integer NOT NULL,"city" Text NOT NULL,PRIMARY KEY("gameID"))
Participations		CREATE TABLE "Participations" ("eventID" Integer,"gameID" Integer,"medal" TEXT CHECK("medal" IN ("Gold","Bronze","Silver")),"athleteID" Integer,PRIMARY KEY ("eventID","gameID","athleteID"),CONSTRAINT "FK_Participations.athleteID" FOREIGN KEY ("athleteID")
Sports		CREATE TABLE "Sports" ("sportID" Integer,"sportsType" Text NOT NULL,PRIMARY KEY ("sportID"))
Teams		CREATE TABLE "Teams" ("teamID" Integer,"country" Text NOT NULL,"noc" Text NOT NULL,PRIMARY KEY ("teamID"))

--Testing to enter incorrect medal type
INSERT INTO Participations(eventID, gameID, medal, athleteID) VALUES(765, 30, "Copper", 1)

```
Execution finished with errors.
Result: CHECK constraint failed: medal
At line 1:
INSERT INTO Participations(eventID, gameID, medal, athleteID) VALUES(765, 30, "Copper", 1)
```

-- Trying to add again with the same pk
INSERT INTO Participations(eventID, gameID, medal, athleteID) VALUES(765, 30, "Gold", 106890)

```
Execution finished with errors.
Result: UNIQUE constraint failed: Participations.eventID, Participations.gameID, Participations.athleteID
At line 1:
-- Trying to add again with the same pk
INSERT INTO Participations(eventID, gameID, medal, athleteID) VALUES(765, 30, "Gold", 106890)
```

--Testing Fk with non existing eventID
INSERT INTO Participations(eventID, gameID, medal, athleteID) VALUES(766, 30, "Gold", 106890)

```
Execution finished with errors.
Result: FOREIGN KEY constraint failed
At line 2:
INSERT INTO Participations(eventID, gameID, medal, athleteID) VALUES(766, 30, "Gold", 106890)
```

6. Populate the tables with test data

Please see olympic-games-sqlit.db for the populated table. Also, we attached the code to populate the table named “Olympic Games Analysis - Copy3.ipynb”.

7.Define and execute at least five queries that show your database.

```
SELECT *
FROM Athletes
INNER JOIN Participations ON Participations.athleteID = Athletes.athleteID
INNER JOIN Events ON Participations.eventID = Events.eventID
INNER JOIN Sports ON Sports.sportID = Events.sportID
WHERE Sports.sportsType = "Judo"
LIMIT 20



SELECT *
 FROM Athletes
 WHERE athleteID IN (SELECT athleteID
        FROM Athletes
        WHERE Athletes.sex = "M")
LIMIT 20



SELECT *
FROM Athletes
GROUP BY age
HAVING age > 25


SELECT Athletes.name, Athletes.age, Participations.medal, Sports.sportsType, Games.year
FROM Athletes
INNER JOIN Participations ON Participations.athleteID = Athletes.athleteID
INNER JOIN Events ON Participations.eventID = Events.eventID
INNER JOIN Sports ON Sports.sportID = Events.sportID
INNER JOIN Games ON  Games.gameID = Participations.gameID
WHERE Participations.medal = "Gold" AND Sports.sportsType = "Swimming" AND Athletes.age > 30


SELECT Athletes.name, Athletes.age,
CASE
WHEN Athletes.age >= 22 THEN "Old"
ELSE "Young"
END  as agecategory
FROM Athletes
LIMIT 40

-- number of athletes, teams and events in 2008
SELECT COUNT(DISTINCT Participations.eventID) AS nume_events,
                                        COUNT(DISTINCT Attendance.athleteID) AS nume_athletes,
                                        COUNT(DISTINCT Attendance.teamID) AS nume_events
FROM Games, Attendance, Participations
WHERE Games.gameID = Attendance.gameID
                    AND Participations.athleteID = Attendance.athleteID
                    AND Games.year = '2008';

--percentage of women for medal winners in the history of Olympic Games
SELECT women_winner_num, total_winner_num , ROUND(CAST(women_winner_num AS REAL)/
```

```
CAST(total_winner_num AS REAL), 3) * 100 AS 'women_percentage(%)'
FROM ((SELECT COUNT(*) AS women_winner_num
FROM Participations, Athletes
WHERE Participations.athleteID = Athletes.athleteID
        AND Athletes.sex = 'F'
        AND Participations.medal NOTNULL) AS WomenWinnerNum,
(SELECT COUNT(*) AS total_winner_num
FROM Participations, Athletes
WHERE Participations.athleteID = Athletes.athleteID
        AND Participations.medal NOTNULL ) AS TotalWinnerNum)

--average height for athletes participated in men's swimming since 1992,roudned to 2 decimal points
SELECT ROUND(CAST (SUM(Athletes.height) AS REAL) /  COUNT(*), 2) AS
average_height_for_men_swimming_since_92
FROM Athletes, Participations, Games, Sports, Events
WHERE Athletes.athleteID = Participations.athleteID
        AND Games.gameID = Participations.gameID
        AND Participations. eventID = Events.eventID
        AND Events.sportID = Sports.sportID
        AND Games.year >= 1992
        AND Athletes.sex = 'M'
        AND Sports.sportsType = 'Swimming'

--Games where  average age of athletess participated in women's basketball is greater than 26 since
1956
SELECT Games.year, ROUND(CAST (SUM(Athletes.age) AS REAL) /  COUNT(*), 2) AS
average_women_age_since_56
FROM Athletes, Participations, Games, Sports, Events
WHERE Athletes.athleteID = Participations.athleteID
        AND Games.gameID = Participations.gameID
        AND Participations. eventID = Events.eventID
        AND Events.sportID = Sports.sportID
        AND Games.year >= 1956
        AND Athletes.sex = 'F'
        AND sportsType = 'Basketball'
GROUP BY Games.gameID HAVING average_women_age_since_56 > 26;
```