University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ____2____

Date: _____Oct 20th, 2023_____

Group Number: ____142_____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Aditya Nasam	60401031	b2b6c	adityanasam@gmail.com
David Zhang	13376256	c5l9t	davidzhang561@gmail.com
Ian Hunter Johnson	37012119	b9c1c	johnson.ian.h@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Description:
Our CPSC 304 project will be modeling and storing a database in the domain of professional sports, encompassing: the sport being played, individuals involved in sports (i.e athletes), teams, games, leagues, seasons, sponsors, stadiums and broadcasting networks. The database will model information relevant to those working in professional sports, or fans of professional sports such as statistics of games and players, or the usage of a stadium.

Users will be able to store information in the database through the GUI by inputting the desired values into the necessary fields such as inputting the channel name and number to store a new channel. Users will be able to perform specific queries on the database through

the GUI such as looking for all players born in 1998 or the team with the most winning games. Stored information will persist across different instances of the application, with the user being able to access previously stored information upon relaunching the application.

Application Platform:
Our application will be developed using Java and Oracle, with JDBC and JavaFX used for API calls and GUI development respectively.

Section 3: Changes to ER Diagram:

- Implemented all changes requested by TA from milestone 1.
- Adjusted text to all be lowerCamelCase.
- Added liveStartTime to Broadcasts. Is relevant information to the relationship.
- Added weather to Game. This gave us more FDs for normalisation.
- Split stats in Athlete into height and weight. Needed more specificity that just “stats.”
- Added deathDate, age, and nationality to sportsPerson. This gave us more FDs for normalisation.
- Changed “city” attribute in Stadium to “address.” This removed duplication of city column in TeamHomeStadium relation. This also changed the key to be just address.

Section 4 and 5: Schema and Functional Dependencies (FDs)

Note: Underline = primary key, **bold** = foreign key, (CK) = candidate key.

- Sport (sportName: string, rules: string)
    - sportName → rules
- Stadium(address: string, sName: string, seatCount: int)
    - address → sName, seatCount
- TeamHomeStadium (tName: string, city: string, **address**: string, **sportName**: string, mascot: string) address, sportName are not NULL.
    - tName, city → mascot, address, sportName,
- In (**splD**: int, **tName**: string, **city**: string)
    - No non-trivial F.Ds as all attributes are part of a primary key.
- LeagueOrganises (lName: string, tier: int, revenue: float, **sportName**: string) sportName is not NULL
    - lName → tier, revenue, **sportName**
- SeasonChampion (**lName**: string, year: int, startDate: date, endDate: date, daysRun: int, **tName**: string, **city**: string) tName and city are not NULL.
    - lName, year → startDate, endDate, daysRun, tName, city
    - startDate, endDate → daysRun
    - startDate, daysRun → endDate
    - endDate, daysRun → startDate
    - (rationale: knowing two out of three of startDate, endDate or daysRun lets you calculate the third attribute.)
- PlaysFor (**tName**: string, **city**: string, **lName**: string)
    - No non-trivial F.Ds as all attributes are part of a primary key.

- GameHostsInSeason(**tNameA**: string, **cityA**: string, **tNameB (CK)**: string, **cityB (CK)**: string, Date: date, scoreA: int, scoreB: int, weather: string, **lName**: string, **yearL**: int, **address**: string). tNameB, cityB, lName, yearL, address are not NULL.
    - tNameA, cityA, Date → scoreA, scoreB, weather, address, lName, yearL, tNameB, cityB
    - tNameB, cityB, Date → scoreA, scoreB, weather, address, lName, yearL, tNameA, cityA
    - address, Date → weather
    - lName, tNameA → cityA (rationale: should not have duplicate team names within a league, so if I know the league and team name, I know the team city.)
    - lName, tNameB → cityB (rationale: should not have duplicate team names within a league, so if I know the league and team name, I know the team city.)
    - Date → yearL
- Network(nName: string, channel: int)
    - nName → channel
- Broadcasts(**nName**: string, **tNameA**: string, **cityA**: string, **Date**: date, liveStartTime: string)
    - nName, Date, tNameA, cityA → liveStartTime
- Sponsor(sponName: string, networth: float)
    - sponName → networth
- Funds(**sponName**: string, **lName**: string, monetaryContribution: float)
    - sponName,lName → monetary contribution

- sportsPerson (spID: int, spName: string, phoneNumber (CK): int, birthDate: int, age: int, salary: float, nationality: string, deathDate: date) birthDate, age are not NULL
    - spID → spName, phoneNumber, birthDate, salary, age, deathDate, nationality
    - phoneNumber → spID, spName, birthDate, salary, age, deathDate, nationality
    - birthDate, deathDate → age

- Athlete (**spID**: int, weight: float, height: float, jerseyID: int, numberOfMVPs: int)
    - spID → weight, height, jerseyID, numberOfMVPs

- Coach (**spID**: int, yearsCoaching: int, numberOfChampionships: int)
    - spID → yearsCoaching, numberOfChampionships

- Owner (**spID**: int, netWorth: float)
    - spID → netWorth

Section 6: Normalisation

Schema that are already normalised: Sport, Stadium, TeamHomeStadium, In, LeagueOrganises, PlaysFor, Network, Broadcasts, Sponsor, Funds, Athlete, Coach, Owner
Normalisation:

SeasonChampionis not in 3NF or BCNF as {startDate, endDate} is not a superkey of SeasonChampion and {daysRun} is not part of a minimal key of SeasonChampion.

- Find minimal cover of FDs:

- Put FDs in standard form:
  - lName, year → startDate
  - lName, year → endDate
  - lName, year → daysRun
  - lName, year → tName
  - lName, year → city
  - startDate, endDate → daysRun
  - startDate, daysRun → endDate
  - endDate, daysRun → startDate

- Minimise LHS of each FD:
  - lName, year:
    - {lName, year}+ = {lName, year, startDate, endDate, daysRun, tName, city}
    - {lName}+ = {lName}
    - {year}+ ={ year}
    - **Can't be minimised any further.**
  - startDate, endDate:
    - {startDate, endDate}+ = {startDate, endDate, daysRun}
    - {startDate}+ = {startDate}
    - {endDate}+ = {endDate}
    - **Can't be minimised any further**
  - startDate, daysRun:
    - {startDate, daysRun}+ = {startDate, endDate, daysRun}
    - {startDate}+ = {startDate}
    - {daysRun}+ = {daysRun}
    - **Can't be minimised any further**
  - endDate, daysRun:
    - {endDate, daysRun}+ = {startDate, endDate, daysRun}
    - {endDate}+ = {endDate}
    - {daysRun}+ = {daysRun}
    - **Can't be minimised any further**

- Remove redundant FDs:
  - Consider {lName, year}+ ignoring FDs with {lName, year} as the LHS:
    - {lName, year}+ = {lName, year}
    - **All FDs with LHS of {lName, year}+ are not redundant**
  - Consider {startDate, endDate}+ ignoring startDate, endDate → daysRun:
    - {startDate, endDate}+ = {startDate, endDate}
    - **startDate, endDate → daysRun is not redundant.**
  - Consider{startDate, daysRun}+ ignoring startDate, daysRun → endDate:
    - {startDate, daysRun}+ = {startDate, daysRun}
    - **startDate, daysRun → endDate is not redundant.**
  - Consider{endDate, daysRun}+ ignoring endDate, daysRun → startDate:
    - {endDate, daysRun}+ = {endDate, daysRun}

- **endDate, daysRun → startDate is not redundant.**

- **Standard form FDs already minimal, proceed with normalisation.**
  - startDate, endDate → daysRun violates 3NF, decompose to BCNF.
    - StartEndDaysRun(startDate, endDate. daysRun)
      - Primary key: {startDate, endDate}
      - Candidate key: {startDate, daysRun}
      - Candidate key: {endDate, daysRun}
    - SeasonChampion2(lName, year, startDate, endDate, tName, city)
      - Primary key: {lName, year}
    - StartEndDaysRun and SeasonChampion2 are now both in BCNF, all FDs are still preserved.
- StartEndDaysRun has been separated from SeasonChampion:
  - StartEndDaysRun (startDate: date, endDate: date, daysRun int) daysRun is NOT NULL
  - SeasonChampion (**lName**: string, year: int, **startDate**: date, **endDate**: date, **tName**: string, **city**: string)

GameHostsInSeason is not in 3NF or BCNF

- Find minimal key:
  -

Left	Middle	Right
Date	tNameA, tNameB, cityA, cityB, address, lName,	Weather, yearL, scoreA, scoreB

  - Closures:
    - {Date}+ = {Date, yearL}
    - {Date, tNameA}+ = {Date, yearL, tNameA}
    - {Date, tNameB}+ = {Date, yearL, tNameB}
    - {Date, cityA}+ = {Date, yearL, cityA}
    - {Date, cityB}+ = {Date, yearL, cityB}
    - {Date, address}+ = {Date, yearL, address, weather}
    - {Date, lName}+ = {Date, yearL, lName}
    - {Date, tNameA, tNameB}+ = {Date, yearL, tNameA, tNameB}
    - {Date, tNameA, cityA}+ = {Date, yearL, tNameA, cityA, tNameB, cityB, address, lName, scoreA, scoreB, weather} → minimal Key
    - Other Minimal keys:
      - {Date, tNameB, cityB}+ = {Date, yearL, tNameA, cityA, tNameB, cityB, address, lName, scoreA, scoreB, weather}
      - {Date, tNameA, lName}+ = {Date, yearL, tNameA, lName, tNameB, cityB, scoreA, scoreB, address, weather, cityA}
      - {Date, tNameB, lName}+ = {Date, yearL, tNameA, lName, tNameB, cityB, scoreA, scoreB, address, weather, cityA}
    - {lName, tNameA}+ = (lName, tNameA, cityA}
    - {lName, tNameB}+ = (lName, tNameB, cityB}

- We see that at least one FD is not BCNF, Date → yearL. And this is also not 3NF because yearL is not part of any minimal key.

- Convert to 3NF:
  - Find minimal cover:
    - Standard forms: red = found to be redundant
      - tNameA, cityA, Date → scoreA
      - tNameA, cityA, Date → scoreB
      - tNameA, cityA, Date → weather
      - tNameA, cityA, Date → address
      - tNameA, cityA, Date → lName
      - tNameA, cityA, Date → yearL
      - tNameA, cityA, Date → tNameB
      - tNameA, cityA, Date → cityB
      - tNameB, cityB, Date → scoreA
      - tNameB, cityB, Date → scoreB
      - tNameB, cityB, Date → weather
      - tNameB, cityB, Date → address
      - tNameB, cityB, Date → lName
      - tNameB, cityB, Date → yearL
      - tNameB, cityB, Date → tNameA
      - tNameB, cityB, Date → cityA
      - address, Date → weather
      - lName, tNameA → cityA
      - lName, tNameB → cityB
      - Date → yearL
    - Minimise LHS of FDs:
      - tNameA, cityA, Date:
        - {tNameA, cityA}+ = {tNameA, cityA}
        - {tNameA, Date}+ = {tNameA, Date, yearL}
        - {cityA, Date}+ = {cityA, Date, yearL}
        - Nothing to minimise (likewise for {tNameB, cityB, Date})
      - address, Date:
        - {Date}+ = {Date, yearL}
        - {address}+ = {address}
        - Nothing to minimise
      - lName, tNameB:
        - {lName}+ = {lName}
        - {tNameB}+ = {tNameB}
        - Nothing to minimise (likewise for {lName, tNameA})
    - Find and delete redundant FDs (find closure without considering the given FD):
      - tNameA, cityA, Date → scoreB
        - {tNameA, cityA, Date}+ → {tNameA, cityA, Date, scoreA, weather, address, lName, yearL, tNameB, cityB, scoreB}
        - Redundant
      - tNameA, cityA, Date → weather

        - {tNameA, cityA, Date}+ → {tNameA, cityA, Date, scoreA, weather, address, lName, yearL, tNameB, cityB, scoreB}
        - Redundant
    - tNameA, cityA, Date → address
        - {tNameA, cityA, Date}+ → {tNameA, cityA, Date, scoreA, weather, lName, yearL, tNameB, cityB, scoreB}
        - Redundant
    - tNameA, cityA, Date → lName
        - {tNameA, cityA, Date}+ → {tNameA, cityA, Date, scoreA, weather, lName, yearL, tNameB, cityB, scoreB}
        - Redundant
    - tNameA, cityA, Date → yearL
        - {tNameA, cityA, Date}+ → {tNameA, cityA, Date, scoreA, weather, lName, yearL, tNameB, cityB, scoreB}
        - Redundant
    - tNameA, cityA, Date → tNameB
        - {tNameA, cityA, Date}+ → {tNameA, cityA, Date, scoreA, weather, yearL, cityB}
        - Required
    - tNameA, cityA, Date → cityB
        - {tNameA, cityA, Date}+ → {tNameA, cityA, Date, scoreA, weather, yearL, tNameB}
        - Required
    - tNameB, cityB, Date → scoreA
        - {tNameB, cityB, Date}+ → {tNameB, cityB, Date, weather, yearL, scoreB, address, lName, tNameA, cityA, scoreA}
        - Redundant
    - tNameB, cityB, Date → scoreB
        - {tNameB, cityB, Date}+ → {tNameB, cityB, Date, weather, yearL, address, lName, tNameA, cityA, scoreA}
        - Required
    - tNameB, cityB, Date → weather
        - {tNameB, cityB, Date}+ → {tNameB, cityB, Date, weather, yearL, scoreB, address, lName, tNameA, cityA, scoreA}
        - Redundant
    - tNameB, cityB, Date → address
        - {tNameB, cityB, Date}+ → {tNameB, cityB, Date, yearL, scoreB, lName, tNameA, cityA, scoreA}
        - Required
    - tNameB, cityB, Date → lName
        - {tNameB, cityB, Date}+ → {tNameB, cityB, Date, yearL, scoreB, address, tNameA, cityA, scoreA}
        - Required
    - tNameB, cityB, Date → yearL

            - {tNameB, cityB, Date}+ → {tNameB, cityB, Date, weather, yearL, scoreB, address, lName, tNameA, cityA, scoreA}
            - Redundant
        - tNameB, cityB, Date → tNameA
            - {tNameB, cityB, Date}+ → {tNameB, cityB, Date, weather, yearL, scoreB, address, lName, cityA}
            - Required
        - tNameB, cityB, Date → cityA
            - {tNameB, cityB, Date}+ → {tNameB, cityB, Date, weather, yearL, scoreB, address, lName, tNameA, cityA, scoreA}
            - Redundant
        - address, Date → weather
            - {address, Date}+ = {address, Date, yearL}
            - Required
        - lName, tNameA → cityA
            - {lName, tNameA}+ = (lName, tNameA}
            - Required
        - lName, tNameB → cityB
            - {lName, tNameB}+ = (lName, tNameB}
            - Required
        - Date → yearL
            - {Date}+ = (Date}
            - Required
- Minimal cover:
    - tNameA, cityA, Date → scoreA
    - tNameA, cityA, Date → tNameB
    - tNameA, cityA, Date → cityB
    - tNameB, cityB, Date → scoreB
    - tNameB, cityB, Date → address
    - tNameB, cityB, Date → lName
    - tNameB, cityB, Date → tNameA
    - address, Date → weather
    - lName, tNameA → cityA
    - lName, tNameB → cityB
    - Date → yearL
- Decomposition of GameHostsInSeason(**tNameA**, **cityA**, **tNameB (CK)**, **cityB (CK)**, Date, scoreA, scoreB, weather, **lName**, **yearL**, **address**) to BCNF:
    - Decompose on Date → yearL:
        - GameDateYear(Date, **yearL**)
        - GameHostsInSeason(**tNameA**, **cityA**, **tNameB (CK)**, **cityB (CK)**, Date, scoreA, scoreB, weather, **lName**, **address**)
    - Further decompose on address, Date → weather:
        - GameAddressDate(**address**, Date, weather)
        - GameHostsInSeason(**tNameA**, **cityA**, **tNameB (CK)**, **cityB (CK)**, Date, scoreA, scoreB, **lName**, **address**)

        - Further decompose on tNameA, cityA, Date → scoreA:
            - GameTeamAScore(**tNameA**, **cityA**, Date, scoreA)
            - GameHostsInSeason(**tNameA**, **cityA**, **tNameB (CK)**, **cityB (CK)**, Date, scoreB, **lName**, **address**)
        - Further decompose on tNameB, cityB, Date → scoreB
            - GameTeamBScore(**tNameB**, **cityB**, Date, scoreB)
            - GameHostsInSeason(**tNameA**, **cityA**, **tNameB (CK)**, **cityB (CK)**, Date, **lName**, **address**)
        - Further decompose on tNameB, cityB, Date → address
            - GameAddress(**tNameB**, **cityB**, Date, **address**)
            - GameHostsInSeason(**tNameA**, **cityA**, **tNameB (CK)**, **cityB (CK)**, Date, **lName**)
        - Further decompose on lName, tNameB → cityB:
            - LeagueTeamB(**lName**, **tNameB**, **cityB**)
            - GameHostsInSeason(**tNameA**, **cityA**, **tNameB (CK)**, Date, **lName**)
        - Further decompose on tNameA, cityA, Date → tNameB:
            - GameTeamAOpponent(**tNameA**, **cityA**, Date, **tNameB**)
            - GameHostsInSeason(**tNameA**, **cityA**, Date, **lName**)
        - Further decompose on lName, tNameA → cityA:
            - LeagueTeamA(**lName**, **tNameA**, **cityA**)
            - GameHostsInSeason(**tNameA**, Date, **lName**)
        - We have lost 3 relationships, so add them back to be in 3NF:
            - GameTeamAOpponentCity(**tNameA**, **cityA**, Date, **cityB**)
            - GameTeamBLeague(**tNameB**, **cityB**, Date, **lName**)
            - GameTeamBOpponent(**tNameB**, **cityB**, Date, **tNameA**)
    - Final 3NF tables:
        - GameHostsInSeason(**tNameA**: string, Date: date, **lName**: string)
        - GameDateYear(**Date**: date, **yearL:** int) yearL is not NULL
        - GameAddressDate(**address**: string, **Date**: date, weather: string)
        - GameTeamAScore(**tNameA**: string, **cityA**: string, **Date**: date, scoreA)
        - GameTeamBScore(**tNameB**, **cityB**, Date: date, scoreB)
        - GameAddress(**tNameB**, **cityB**, **Date**: date, **address**: string) address is not NULL
        - LeagueTeamB(**lName**: string, **tNameB**: string, **cityB**: string) cityB is not NULL
        - GameTeamAOpponent(**tNameA**: string, **cityA**: string, **Date**: date, **tNameB**) tNameB is not NULL
        - LeagueTeamA(**lName**: string, **tNameA**: string, **cityA**: string) cityA is not NULL
        - GameTeamAOpponentCity(**tNameA**: string, **cityA**: string, **Date**: date, **cityB**) cityB is not NULL
        - GameTeamBLeague(**tNameB**: string, **cityB**: string, **Date**: date, **lName**: string) lName is not NULL
        - GameTeamBOpponent(**tNameB**: string, **cityB**: string, **Date**: date, **tNameA**) tNameA is not NULL

sportsPerson is not in both BCNF and 3NF.

- Find the minimal key

Left	Middle	Right
	spID, phoneNumber, birthDate, deathDate	spName, age, salary, nationality

- Find closures
    - spID+ = {spID,spName, phoneNumber, birthDate, deathDate, age, salary, nationality}
    - {PhoneNumber}+ = {spID, spName, phoneNumber , birthDate, deathDate, age, salary}
    - {birthDate, deathDate}+ = {age}

We see that (birthDate,deathDate) is not part of the superkey and also age is not part of any minimal key.

- Find the minimal cover
    - Step1 (standard Form)
        - spID → spName
        - spID → phoneNumber
        - spID → birthDate
        - spID → deathDate
        - spID → salary
        - spID → age
        - spID → nationality
        - phoneNumber → spID
        - phoneNumber→ spName
        - phoneNumber→ birthDate
        - phoneNumber→ deathDate
        - phoneNumber→ salary
        - phoneNumber→ age
        - phoneNumber→ nationality
        - birthDate, deathDate → age
    - Step2 (Minimise LHS)
        - Nothing can be minimised
    - Step3 (Delete Redundant FD's)
        - spID → spName
          spID+ = {spID, phoneNumber, birthDate, deathDate, age, salary, nationality}
          required
        - spID → phoneNumber
          spID+ ={spID,spName,birthDate, deathDate, age, salary, nationality}
          required
        - spID → birthDate
          spID+ ={spID,spName, phoneNumber, birthDate, deathDate, age, salary, nationality}
          redundant

- spID → deathDate
  spID+ ={spID,spName, phoneNumber, birthDate, age, salary, nationality}
  required
- spID → salary
  spID+={spID,spName, phoneNumber, birthDate, deathDate, age, nationality}
  required
- spID → age
  spID+={spID,spName, phoneNumber, birthDate, deathDate, age, salary, nationality}
  redundant
- spID → nationality
  spID+ ={spID,spName, phoneNumber, birthDate, deathDate, age, salary}
  required
- phoneNumber → spID
  PhoneNumber+ ={spName, phoneNumber, birthDate, deathDate, age, salary, nationality}
  required
- phoneNumber→ spName
  phoneNumber+={spID,spName, phoneNumber, birthDate, deathDate, age, salary, nationality}
  redundant
- phoneNumber→ birthDate
  phoneNumber+={spID,spName, phoneNumber, deathDate, age, salary, nationality}
  required
- phoneNumber→ deathDate
  phoneNumber+= {spID,spName, phoneNumber, birthDate, deathDate, age, salary, nationality}
  redundant
- phoneNumber→ salary
  phoneNumber+={spID,spName, phoneNumber, birthDate, deathDate, age, salary, nationality}
  redundant
- phoneNumber→ age
  phoneNumber+= {spID,spName, phoneNumber, birthDate, deathDate, age, salary, nationality}
  redundant
- phoneNumber→ nationality
  phoneNumber+= {spID,spName, phoneNumber, birthDate, deathDate, age, salary, nationality}
  redundant
- birthDate, deathDate → age
  {birthDate, deathDate}+ ={spID,spName, phoneNumber, birthDate, deathDate, salary, nationality}
  required

- Minimal Cover
    - spID→salary
    - spID→nationality
    - birthDate, deathDate→age
    - phoneNumber→ birthDate
    - spID →spName
    - spID → deathDate
    - spID → phoneNumber

- Decomposition on sportsPerson(spID,spName,phoneNumber(CK), birthDate, deathDate, age, salary, nationality) into BCNF:
    - Decompose on spID → salary
        - sportsPersonSalary(spID,salary)
        - sportsPerson(spID,spName,phoneNumber(CK), birthDate, deathDate,age, nationality)
    - Further decompose on spID→ nationality
        - sportsPersonNationality(spID,nationality)
        - sportsPerson(spID,spName,phoneNumber(CK), birthDate, deathDate,age)
    - Further decompose on birthDate, deathDate → age
        - sportsPersonAge(birthDate,deathDate,age)
        - sportsPerson(spID,spName,phoneNumber(CK), birthDate, deathDate)
    - Further decompose on phoneNumber → birthDate
        - sportsPersonDOB(phoneNumber,birthDate)
        - sportsPerson(spID,spName,phoneNumber(CK), deathDate)
    - Further decompose on spID→spName
        - sportsPersonPlayingSport(spID,spName)
        - sportsPerson(spID,phoneNumber(CK),deathDate)
    - Further decompose on spID→deathDate
        - sportsPersonDeathDate(spID, deathDate)
        - sportsPerson(spID,phoneNumber(CK))

- Final 3NF relations:
    - sportsPerson(spID,phoneNumber(CK))
    - sportsPersonSalary(spID,salary)
    - sportsPersonNationality(spID,nationality)
    - sportsPersonAge(birthDate,deathDate,age)
    - sportsPersonDOB(phoneNumber,birthDate)
    - sportsPersonPlayingSport(spID,spName)
    - sportsPersonDeathDate(spID, deathDate)

Section 7 and 8: SQL DDL Statements and INSERT Statements

*Oracle does not support ON UPDATE CASCADE. We want functionality like ON UPDATE CASCADE. For this milestone, we will not update (NO ACTION) and then implement these updates with our own functions.

- CREATE TABLE Sport (
  sportName VARCHAR(255) PRIMARY KEY,
  rules VARCHAR(4095));
  - INSERT
    INTO Sport(sportName, rules)
    VALUES (“Ice Hockey”, “Puck into net”);
  - INSERT
    INTO Sport(sportName, rules)
    VALUES (“Soccer”, “Ball into net”);
  - INSERT
    INTO Sport(sportName, rules)
    VALUES (“Baseball”, “Bat hit ball”);
  - INSERT
    INTO Sport(sportName, rules)
    VALUES (“American Football”, “Run into each other”);
  - INSERT
    INTO Sport(sportName, rules)
    VALUES (“Basketball”, “Ball through net”);

- CREATE TABLE Stadium
  (address VARCHAR(255) PRIMARY KEY,
  sName VARCHAR(255),
  seatCount INT);
  - INSERT
    INTO Stadium(address, sName, seatCount)
    VALUES (‘800 Griffiths Way, Vancouver, BC V6B 6G1, Canada’,
    ‘Rogers Arena’, ‘19700’);
  - INSERT
    INTO Stadium(address, sName, seatCount)
    VALUES (‘10220 104 Ave NW, Edmonton, AB T5J 0H6, Canada’,
    ‘Rogers Place’, ‘18347’);
  - INSERT
    INTO Stadium(address, sName, seatCount)
    VALUES (‘Sir Matt Busby Way, Old Trafford, Stretford, Manchester
    M16 0RA, UK’, ‘Old Trafford’, ‘74310’);
  - INSERT
    INTO Stadium(address, sName, seatCount)
    VALUES (‘Etihad Stadium, Ashton New Rd, Manchester M11 3FF, UK’,
    ‘Etihad Stadium’, ‘‘54400’’);
  - INSERT
    INTO Stadium(address, sName, seatCount)
    VALUES (‘6288 Stadium Rd, Vancouver, BC V6T 1Z3, Canada’,
    ‘Thunderbird Stadium’, ‘5054’);
  - INSERT

INTO Stadium(address, sName, seatCount)
VALUES (‘334 1st Ave N, Seattle, WA 98109, USA’,
‘Climate Pledge Arena, ‘18300’);

- CREATE TABLE TeamHomeStadium (
  tName VARCHAR(255),
  city, VARCHAR(255),
  address VARCHAR(255) NOT NULL,
  sportName VARCHAR(255) NOT NULL,
  mascot VARCHAR(100),

  PRIMARY KEY (tName, city),
  FOREIGN KEY(address) REFERENCES Stadium(address)
  ON DELETE CASCADE
  ON UPDATE CASCADE*,
  FOREIGN KEY (sportName) REFERENCES Sport(sportName)
  ON DELETE CASCADE
  ON UPDATE CASCADE*);
  - INSERT
    INTO TeamHomeStadium(tName, city, address, sportName, mascot)
    VALUES (‘Canucks’, ‘Vancouver’, ‘800 Griffiths Way, Vancouver, BC V6B 6G1, Canada’, ‘Ice Hockey’, ‘Fin the Whale’)
  - INSERT
    INTO TeamHomeStadium(tName, city, address, sportName, mascot)
    VALUES (‘Oilers’, ‘Edmonton’, ‘10220 104 Ave NW, Edmonton, AB T5J 0H6, Canada’, ‘Ice Hockey’, ‘Hunter the Lynx’)
  - INSERT
    INTO TeamHomeStadium(tName, city, address, sportName)
    VALUES (‘Manchester United F.C.’, ‘Manchester’, ‘Sir Matt Busby Way, Old Trafford, Stretford, Manchester M16 0RA, UK’, ‘Soccer’)
  - INSERT
    INTO TeamHomeStadium(tName, city, address, sportName, mascot)
    VALUES (‘Manchester City F.C.’, ‘Manchester’, ‘Etihad Stadium, Ashton New Rd, Manchester M11 3FF, UK’, ‘Soccer’)
  - INSERT
    INTO TeamHomeStadium(tName, city, address, sportName, mascot)
    VALUES (‘UBC Thunderbirds’, ‘Vancouver’, ‘6288 Stadium Rd, Vancouver, BC V6T 1Z3, Canada’, ‘American Football’, ‘Thunderbird’)
  - INSERT
    INTO TeamHomeStadium(tName, city, address, sportName, mascot)
    VALUES (‘Kraken’, ‘Seattle’, ‘334 1st Ave N, Seattle, WA 98109, USA’, ‘Ice Hockey’, ‘Buoy’)

- CREATE TABLE In (
  spID INT,
  tName VARCHAR(255),
  city VARCHAR(255)

PRIMARY KEY (spID, tName, city),
FOREIGN KEY (spID) REFERENCES SportsPerson(spID)
ON DELETE CASCADE
ON UPDATE CASCADE*,
FOREIGN KEY (tName) REFERENCES TeamHomeStadium(tName)
ON DELETE CASCADE
ON UPDATE CASCADE*,
FOREIGN KEY (city) REFERENCE TeamHomeStadium(city)
ON DELETE CASCADE
ON UPDATE CASCADE*);

- INSERT
  INTO In(spID, tName, city)
  VALUES (111111, “Canucks”, “Vancouver);
- INSERT
  INTO In(spID, tName, city)
  VALUES (222222, “Canucks”, “Vancouver);
- INSERT
  INTO In(spID, tName, city)
  VALUES (333333, “Oilers”, “Edmonton);
- INSERT
  INTO In(spID, tName, city)
  VALUES (444444, “Oilers”, “Edmonton);
- INSERT
  INTO In(spID, tName, city)
  VALUES (55555, “UBC Thunderbirds”, “Vancouver);

- CREATE TABLE LeagueOrganises {
  lName VARCHAR(255),
  tier INT,
  revenue FLOAT,
  sportName VARCHAR(255) NOT NULL,

  PRIMARY KEY (lName),
  FOREIGN KEY (sportName) REFERENCES Sport(sportName)
  ON DELETE CASCADE
  ON UPDATE CASCADE*);
  - INSERT
    INTO LeagueOrganises(lName, tier, revenue, sportName)
    VALUES (“NHL”, 1, 111.11 “Ice Hockey”);
  - INSERT
    INTO LeagueOrganises(lName, tier, revenue, sportName)
    VALUES (“Premier League, 1, 222.22, “Soccer”);
  - INSERT
    INTO LeagueOrganises(lName, tier, revenue, sportName)
    VALUES (“U Sport”, 2, 333.33, “American Football”);
  - INSERT
    INTO LeagueOrganises(lName, tier, revenue, sportName)

VALUES (“NBA”, 1, 444.44, “Basketball”);

- INSERT
  INTO LeagueOrganises(lName, tier, revenue, sportName)
  VALUES (“MLB”, 1, 555.55, “Baseball”);

- CREATE TABLE StartEndDaysRun (
  startDate DATE,
  endDate DATE,
  daysRun Date NOT NULL,

  PRIMARY KEY (startDate, endDate));
  - INSERT
    INTO StartEndDaysRun(startDate, endDate)
    VALUES (“2012-01-01”, “2012-12-31”, 366);
  - INSERT
    INTO StartEndDaysRun(startDate, endDate)
    VALUES (“2012-01-01”, “2012-01-01”, 1);
  - INSERT
    INTO StartEndDaysRun(startDate, endDate)
    VALUES (“2012-01-01”, “2012-01-02”, 2);
  - INSERT
    INTO StartEndDaysRun(startDate, endDate)
    VALUES (“2012-01-01”, “2012-01-03”, 3);
  - INSERT
    INTO StartEndDaysRun(startDate, endDate)
    VALUES (“2012-01-01”, “2012-01-04”, 4);

- CREATE TABLE SeasonChampion (
  lName VARCHAR(255),
  year INT,
  startDate DATE,
  endDate DATE,
  tName VARCHAR(255) NOT NULL,
  city VARCHAR(255) NOT NULL,

  PRIMARY KEY (lName, year),
  FOREIGN KEY (lName) REFERENCES LeagueOrganises(lName)
  ON DELETE CASCADE
  ON UPDATE CASCADE*,
  FOREIGN KEY (startDate) REFERENCES StartEndDaysRun(startDate)
  ON UPDATE CASCADE*,
  FOREIGN KEY (startDate) REFERENCES StartEndDaysRun(startDate)
  ON UPDATE CASCADE*,
  FOREIGN KEY (tName) REFERENCES TeamHomeStadium(tName)
  ON DELETE CASCADE
  ON UPDATE CASCADE*,
  FOREIGN KEY (city) REFERENCES TeamHomeStadium(city))

ON DELETE CASCADE
ON UPDATE CASCADE*);

- INSERT
  INTO PlaysFor(lName, year, startDate, endDate, tName, city)
  VALUES (“NHL”, 2012, “2012-01-01”, “2012-12-31”, “Canuck”, “Vancouver”);
- INSERT
  INTO PlaysFor(lName, year, startDate, endDate, tName, city)
  VALUES (“NHL”, 2013, “2013-01-01”, “2013-12-31”, “Canuck”, “Vancouver”);
- INSERT
  INTO PlaysFor(lName, year, startDate, endDate, tName, city)
  VALUES (“NHL”, 2014, “2014-01-01”, “2014-12-31”, “Canuck”, “Vancouver”);
- INSERT
  INTO PlaysFor(lName, year, startDate, endDate, tName, city)
  VALUES (“NHL”, 2015, “2015-01-01”, “2015-12-31”, “Canuck”, “Vancouver”);
- INSERT
  INTO PlaysFor(lName, year, startDate, endDate, tName, city)
  VALUES (“NHL”, 2016, “2016-01-01”, “2016-12-31”, “Canuck”, “Vancouver”);

- CREATE TABLE PlaysFor (
  tName VARCHAR(255),
  city VARCHAR(255),
  lName VARCHAR(255),

  PRIMARY KEY (tName, city, lName)
  FOREIGN KEY (tName) REFERENCES TeamHomeStadium(tName)
  ON DELETE CASCADE
  ON UPDATE CASCADE*,
  FOREIGN KEY (city) REFERENCES TeamHomeStadium(city)
  ON DELETE CASCADE
  ON UPDATE CASCADE*,
  FOREIGN KEY (lName) REFERENCES LeagueOrganises(lName)
  ON DELETE CASCADE
  ON UPDATE CASCADE*);

  - INSERT
    INTO PlaysFor(tName, city, lName)
    VALUES (“Canucks”, “Vancouver”, “NHL”);
  - INSERT
    INTO PlaysFor(tName, city, lName)
    VALUES (“Oilers”, “Edmonton”, “NHL”);
  - INSERT

INTO PlaysFor(tName, city, lName)
VALUES (“Manchester United F.C”, “Manchester”, “Premier League”);

- INSERT
  INTO PlaysFor(tName, city, lName)
  VALUES (“UBC Thunderbirds”, “Vancouver”, “U Sports”);
- INSERT
  INTO PlaysFor(tName, city, lName)
  VALUES (“Kraken”, “Seattle”, “NHL”);

- CREATE TABLE Network (
  nNameVARCHAR(255) PRIMARY KEY
  channel INT);
  - INSERT
    INTO Network(nName)
    VALUES (‘ESPN’)
  - INSERT
    INTO Network(nName, channel)
    VALUES (‘Sportsnet ONE’, ‘418’)
  - INSERT
    INTO Network(nName)
    VALUES (‘BBC’)
  - INSERT
    INTO Network(nName)
    VALUES (‘CBC Sports’)
  - INSERT
    INTO Network(nName)
    VALUES (‘FOX Sports 1’)

- CREATE TABLE Broadcasts (
  nNameVARCHAR(255),
  tNameA VARCHAR(255),
  cityA VARCHAR(255),
  Date DATE,
  liveStartTime CHAR[5],
  PRIMARY KEY (nName, tNameA, cityA, Date),
  FOREIGN KEY (nName) REFERENCES Network(nName),
  ON DELETE CASCADE
  ON UPDATE CASCADE*,
  FOREIGN KEY (tNameA, cityA) REFERENCES TeamHomeStadium(tName, city),
  ON DELETE CASCADE
  ON UPDATE CASCADE*,
  FOREIGN KEY (Date) REFERENCES GameHostsInSeason(Date)
  ON DELETE CASCADE
  ON UPDATE CASCADE*);
  - INSERT
    INTO Broadcasts(nName, tNameA, cityA, Date, liveStartTime)

VALUES ('Sportsnet ONE', 'Canucks'', ''Vancouver'', '12-OCT-2022', '19:00')
- INSERT INTO Broadcasts(nName, tNameA, cityA, Date, liveStartTime) VALUES ('ESPN', 'Kraken', 'Seattle', '22-OCT-2022', '19:00')
- INSERT INTO Broadcasts(nName, tNameA, cityA, Date, liveStartTime) VALUES ('BBC', 'Manchester United F.C.', ''Manchester'', '22-MAR-2018', '18:30')
- INSERT INTO Broadcasts(nName, tNameA, cityA, Date, liveStartTime) VALUES ('BBC', 'Manchester City F.C.', ''Manchester'', '04-FEB-2018', '17:30')
- INSERT INTO Broadcasts(nName, tNameA, cityA, Date) VALUES ('CBC Sports', 'Oilers', 'Edmonton'', '12-MAR-1999)

- CREATE TABLE Sponsor (
  sponName VARCHAR(255) PRIMARY KEY,
  networth FLOAT);
  - INSERT INTO Sponsor(sponName, networth) VALUES ('Tesla', '808000000000')
  - INSERT INTO Sponsor(sponName, networth) VALUES ('Ford Motor Company', '54000000000')
  - INSERT INTO Sponsor(sponName, networth) VALUES ('Nike', '157000000000')
  - INSERT INTO Sponsor(sponName, networth) VALUES ('Adidas', '36000000000')
  - INSERT INTO Sponsor(sponName, networth) VALUES ('Scotiabank', '51000000000')

- CREATE TABLE Funds (
  sponName VARCHAR(255),
  lName VARCHAR(255),
  monetaryContribution FLOAT,
  PRIMARY KEY (sponName, lName),
  FOREIGN KEY (sponName) REFERENCES Sponsor(sponName));
  - INSERT INTO Funds(sponName, lName, monetaryContribution) VALUES (''Adidas'', 'Premier League', '1500000')
  - INSERT INTO Funds(sponName, lName, monetaryContribution) VALUES ('Scotiabank', 'NHL', '12000000')

    - INSERT
      INTO Funds(sponName, lName)
      VALUES (‘Tesla’, ‘National Basketball League’)
    - INSERT
      INTO Funds(sponName, lName, monetaryContribution)
      VALUES (‘Adidas’, ‘Premier League’, ‘15000000’)
    - INSERT
      INTO Funds(sponName, lName)
      VALUES (‘Ford Motor Company’, ‘Major League Baseball’)

- CREATE TABLE GameHostsInSeason(
      tNameA VARCHAR(255),
      Date DATE,
      lName VARCHAR(255),
      PRIMARY KEY (tNameA, Date, lName),
      FOREIGN KEY (tName) REFERENCES TeamHomeStadium(tName),
          ON DELETE CASCADE
          ON UPDATE CASCADE*,
      FOREIGN KEY (lName) REFERENCES LeagueOrganises(lName),
          ON DELETE CASCADE
          ON UPDATE CASCADE*);
    - INSERT
      INTO GameHostsInSeason(tNameA, Date, lName)
      VALUES (‘Canucks’, ‘12-OCT-2022’, ‘NHL’)
    - INSERT
      INTO GameHostsInSeason(tNameA, Date, lName)
      VALUES (‘Kraken’, ‘22-OCT-2022’, ‘NHL’)
    - INSERT
      INTO GameHostsInSeason(tNameA, Date, lName)
      VALUES (‘Manchester United F.C.’, ‘22-MAR-2018’, ‘Premier League’)
    - INSERT
      INTO GameHostsInSeason(tNameA, Date, lName)
      VALUES (‘Manchester City F.C.’, ‘04-FEB-2018’, ‘Premier League’)
    - INSERT
      INTO GameHostsInSeason(tNameA, Date, lName)
      VALUES (‘Oilers’, ‘12-MAR-1999, ‘NHL’)

- CREATE TABLE GameDateYear(
      Date DATE PRIMARY KEY,
      yearL INT NOT NULL,
      FOREIGN KEY (Date) REFERENCES GameHostsInSeason(Date),
          ON DELETE CASCADE
          ON UPDATE CASCADE*,
      FOREIGN KEY (Date) REFERENCES SeasonChampion(Date),
          ON DELETE CASCADE
          ON UPDATE CASCADE*);
    - INSERT
      INTO GameDateYear(Date, yearL)

VALUES ('12-OCT-2022', '2022')
- INSERT
  INTO GameDateYear(Date, yearL)
  VALUES ('22-OCT-2023', '2022')
- INSERT
  INTO GameDateYear(Date, yearL)
  VALUES ('22-MAR-2018', '2018')
- INSERT
  INTO GameDateYear(Date, yearL)
  VALUES ('04-FEB-2018', '2018')
- INSERT
  INTO GameDateYear(Date, yearL)
  VALUES ('12-MAR-1999', '2018')

- CREATE TABLE GameAddressDate(
  address VARCHAR(255),
  Date DATE,
  weather VARCHAR(255),
  PRIMARY KEY (address, Date),
  FOREIGN KEY (address) REFERENCES Stadium(address),
  ON DELETE CASCADE
  ON UPDATE CASCADE*),
  FOREIGN KEY (Date) REFERENCES SeasonChampion(Date),
  ON DELETE CASCADE
  ON UPDATE CASCADE*);
  - INSERT
    INTO GameAddressDate(address, Date, weather)
    VALUES ('800 Griffiths Way, Vancouver, BC V6B 6G1, Canada',
    '12-OCT-2022', 'Rain')
  - INSERT
    INTO GameAddressDate(address, Date, weather)
    VALUES ('334 1st Ave N, Seattle, WA 98109, USA',
    '22-OCT-2022', 'Rain')
  - INSERT
    INTO GameAddressDate(address, Date, weather)
    VALUES ('Sir Matt Busby Way, Old Trafford, Stretford, Manchester M16 0RA, UK', '22-MAR-2018', 'Rain')
  - INSERT
    INTO GameAddressDate(address, Date, weather)
    VALUES ('Etihad Stadium, Ashton New Rd, Manchester M11 3FF, UK',
    '04-FEB-2018', 'Clear')
  - INSERT
    INTO GameAddressDate(address, Date, weather)
    VALUES ('10220 104 Ave NW, Edmonton, AB T5J 0H6, Canada',
    '12-MAR-1999', 'Snow')

- CREATE TABLE GameTeamAScore(
  tNameA VARCHAR(255),

cityA VARCHAR(255),
Date DATE,
scoreA INT,
PRIMARY KEY (tNameA, cityA, Date),
FOREIGN KEY (tNameA, cityA) REFERENCES
TeamHomeStadium(tName, city),
ON DELETE CASCADE
ON UPDATE CASCADE*),
FOREIGN KEY (Date) REFERENCES SeasonChampion(Date),
ON DELETE CASCADE
ON UPDATE CASCADE*);

- INSERT
  INTO GameTeamAScore(tNameA, cityA, Date, scoreA)
  VALUES (‘Canucks’, ‘Vancouver’, ‘12-OCT-2022’, ‘4’)
- INSERT
  INTO GameTeamAScore(tNameA, cityA, Date, scoreA)
  VALUES (‘Kraken’, ‘Seattle’, ‘22-OCT-2022’, ‘7’)
- INSERT
  INTO GameTeamAScore(tNameA, cityA, Date, scoreA)
  VALUES (‘Manchester United F.C.’, ‘Manchester’, ‘22-MAR-2018’, ‘1’)
- INSERT
  INTO GameTeamAScore(tNameA, cityA, Date, scoreA)
  VALUES (‘Manchester City F.C.’, ‘Manchester’, ‘04-FEB-2018’, ‘1’)
- INSERT
  INTO GameTeamAScore(tNameA, cityA, Date, scoreA)
  VALUES (‘Oilers’, ‘Edmonton’, ‘12-MAR-1999’, ‘0’)

- CREATE TABLE GameTeamBScore(
  tNameB VARCHAR(255),
  cityB VARCHAR(255),
  Date DATE,
  scoreB INT,
  PRIMARY KEY (tNameB, cityB, Date),
  FOREIGN KEY (tNameB, cityB) REFERENCES
  TeamHomeStadium(tName, city),
  ON DELETE CASCADE
  ON UPDATE CASCADE*),
  FOREIGN KEY (Date) REFERENCES SeasonChampion(Date),
  ON DELETE CASCADE
  ON UPDATE CASCADE*);
  - INSERT
    INTO GameTeamBScore(tNameB, cityB, Date, scoreB)
    VALUES (‘Oilers’, ‘Edmonton’, ‘12-OCT-2022’, ‘3’)
  - INSERT
    INTO GameTeamBScore(tNameB, cityB, Date, scoreB)
    VALUES (‘Canucks’, ‘Vancouver’, ‘22-OCT-2022’, ‘2’)
  - INSERT
    INTO GameTeamBScore(tNameB, cityB, Date, scoreB)

VALUES ('Manchester City F.C.', '', '22-MAR-2018', '0')
- INSERT
  INTO GameTeamBScore(tNameB, cityB, Date, scoreB)
  VALUES ('Manchester United F.C.', '', '04-FEB-2018', '1')
- INSERT
  INTO GameTeamBScore(tNameB, cityB, Date, scoreB)
  VALUES ('Canucks', 'Vancouver', '12-MAR-1999', '5')

- CREATE TABLE GameAddress(
  tNameB VARCHAR(255),
  cityB VARCHAR(255),
  Date DATE,
  address VARCHAR(255) NOT NULL,
  PRIMARY KEY (tNameB, cityB, Date),
  FOREIGN KEY (tNameB, cityB) REFERENCES
  TeamHomeStadium(tName, city),
  ON DELETE CASCADE
  ON UPDATE CASCADE*),
  FOREIGN KEY (Date) REFERENCES SeasonChampion(Date),
  ON DELETE CASCADE
  ON UPDATE CASCADE*,
  FOREIGN KEY (address) REFERENCES Stadium(address),
  ON DELETE CASCADE
  ON UPDATE CASCADE*);
  - INSERT
    INTO GameAddress(tNameB, cityB, Date, address)
    VALUES ('Oilers', 'Edmonton', '12-OCT-2022', '800 Griffiths Way, Vancouver, BC V6B 6G1, Canada')
  - INSERT
    INTO GameAddress(tNameB, cityB, Date, address)
    VALUES ('Canucks', 'Vancouver', '22-OCT-2022', '334 1st Ave N, Seattle, WA 98109, USA')
  - INSERT
    INTO GameAddress(tNameB, cityB, Date, address)
    VALUES ('Manchester City F.C.', '', '22-MAR-2018', 'Sir Matt Busby Way, Old Trafford, Stretford, Manchester M16 0RA, UK')
  - INSERT
    INTO GameAddress(tNameB, cityB, Date, address)
    VALUES ('Manchester United F.C.', '', '04-FEB-2018', 'Etihad Stadium, Ashton New Rd, Manchester M11 3FF, UK')
  - INSERT
    INTO GameAddress(tNameB, cityB, Date, address)
    VALUES ('Canucks', 'Vancouver', '12-MAR-1999', '10220 104 Ave NW, Edmonton, AB T5J 0H6, Canada')

- CREATE TABLE LeagueTeamB(
  lName VARCHAR(255),
  tNameB VARCHAR(255),

cityB VARCHAR(255) NOT NULL,
PRIMARY KEY (lName, tNameB),
FOREIGN KEY (lName) REFERENCES LeagueOrganises(lName),
ON DELETE CASCADE
ON UPDATE CASCADE*,
FOREIGN KEY (tNameB, cityB) REFERENCES
TeamHomeStadium(tName, city),
ON DELETE CASCADE
ON UPDATE CASCADE*);

- INSERT
  INTO LeagueTeamB(lName, tNameB, cityB)
  VALUES ('NHL', 'Oilers', 'Edmonton')
- INSERT
  INTO LeagueTeamB(lName, tNameB, cityB)
  VALUES ('NHL', 'Canucks', 'Vancouver')
- INSERT
  INTO LeagueTeamB(lName, tNameB, cityB)
  VALUES ('Premier League', 'Manchester City F.C.', 'Manchester')
- INSERT
  INTO LeagueTeamB(lName, tNameB, cityB)
  VALUES ('Premier League', 'Manchester United F.C.', 'Manchester')
- INSERT
  INTO LeagueTeamB(lName, tNameB, cityB)
  VALUES ('NHL', 'Canucks', 'Vancouver')

- CREATE TABLE GameTeamAOpponent(
  tNameA VARCHAR(255),
  cityA VARCHAR(255),
  Date DATE,
  tNameB VARCHAR(255) NOT NULL,
  PRIMARY KEY (tNameA, cityA, Date),
  FOREIGN KEY (tNameA, cityA) REFERENCES
  TeamHomeStadium(tName, city),
  ON DELETE CASCADE
  ON UPDATE CASCADE*),
  FOREIGN KEY (Date) REFERENCES SeasonChampion(Date),
  ON DELETE CASCADE
  ON UPDATE CASCADE*,
  FOREIGN KEY (tNameB) REFERENCES TeamHomeStadium(tName),
  ON DELETE CASCADE
  ON UPDATE CASCADE*);
  - INSERT
    INTO GameTeamAOpponent(tNameA, cityA, Date, tNameB)
    VALUES ('Canucks', 'Vancouver', '12-OCT-2022', 'Oilers')
  - INSERT
    INTO GameTeamAOpponent(tNameA, cityA, Date, tNameB)
    VALUES ('Kraken', 'Seattle', '22-OCT-2022', 'Canucks')
  - INSERT

INTO GameTeamAOpponent(tNameA, cityA, Date, tNameB)
VALUES ('Manchester United F.C.', 'Manchester', '22-MAR-2018', 'Manchester City F.C.')

- INSERT
  INTO GameTeamAOpponent(tNameA, cityA, Date, tNameB)
  VALUES ('Manchester City F.C.', 'Manchester', '04-FEB-2018', 'Manchester United F.C.')
- INSERT
  INTO GameTeamAOpponent(tNameA, cityA, Date, tNameB)
  VALUES ('Oilers', 'Edmonton', '12-MAR-1999', 'Canucks')

- CREATE TABLE LeagueTeamA(
  lName VARCHAR(255),
  tNameA VARCHAR(255),
  cityA VARCHAR(255) NOT NULL,
  PRIMARY KEY (lName, tNameA),
  FOREIGN KEY (lName) REFERENCES LeagueOrganises(lName),
  ON DELETE CASCADE
  ON UPDATE CASCADE*,
  FOREIGN KEY (tNameA, cityA) REFERENCES
  TeamHomeStadium(tName, city),
  ON DELETE CASCADE
  ON UPDATE CASCADE*);
  - INSERT
    INTO LeagueTeamA(lName, tNameA, cityA)
    VALUES ('NHL', 'Canucks', 'Vancouver')
  - INSERT
    INTO LeagueTeamA(lName, tNameA, cityA)
    VALUES ('NHL', 'Kraken', 'Seattle')
  - INSERT
    INTO LeagueTeamA(lName, tNameA, cityA)
    VALUES ('Premier League', 'Manchester United F.C.', 'Manchester')
  - INSERT
    INTO LeagueTeamA(lName, tNameA, cityA)
    VALUES ('Premier League', 'Manchester City F.C.', 'Manchester')
  - INSERT
    INTO LeagueTeamA(lName, tNameA, cityA)
    VALUES ('NHL', 'Oilers', 'Edmonton')

  NOTE TO TA: Seems like we can just have one LeagueTeam table instead of 2 for team A and B, but normalisation gave us both. Can we just delete one? Or is this critical for lossless join?

- CREATE TABLE GameTeamAOpponentCity(
  tNameA VARCHAR(255),
  cityA VARCHAR(255),
  Date DATE,
  cityB VARCHAR(255) NOT NULL,

PRIMARY KEY (tNameA, cityA, Date),
FOREIGN KEY (tNameA, cityA) REFERENCES
TeamHomeStadium(tName, city),
ON DELETE CASCADE
ON UPDATE CASCADE*),
FOREIGN KEY (Date) REFERENCES SeasonChampion(Date),
ON DELETE CASCADE
ON UPDATE CASCADE*,
FOREIGN KEY (cityB) REFERENCES TeamHomeStadium(city),
ON DELETE CASCADE
ON UPDATE CASCADE*);

- INSERT
  INTO GameTeamAOpponentCity(tNameA, cityA, Date, cityB)
  VALUES (‘Canucks’, ‘Vancouver’, ‘12-OCT-2022’, ‘Oilers’)
- INSERT
  INTO GameTeamAOpponentCity(tNameA, cityA, Date, cityB)
  VALUES (‘Kraken’, ‘Seattle’, ‘22-OCT-2022’, ‘Vancouver’)
- INSERT
  INTO GameTeamAOpponentCity(tNameA, cityA, Date, cityB)
  VALUES (‘Manchester United F.C.’, ‘Manchester’, ‘22-MAR-2018’, ‘Manchester’)
- INSERT
  INTO GameTeamAOpponentCity(tNameA, cityA, Date, cityB)
  VALUES (‘Manchester City F.C.’, ‘Manchester’, ‘04-FEB-2018’, ‘Manchester’)
- INSERT
  INTO GameTeamAOpponentCity(tNameA, cityA, Date, cityB)
  VALUES (‘Oilers’, ‘Edmonton’, ‘12-MAR-1999’, ‘Vancouver’)

- CREATE TABLE GameTeamBLeague (
  tNameB VARCHAR(255),
  cityB VARCHAR(255),
  Date DATE,
  lName VARCHAR(255) NOT NULL,

  PRIMARY KEY (tNameB, cityB, Date),
  FOREIGN KEY (tNameB, cityB) REFERENCES
  TeamHomeStadium(tName, city)
  ON DELETE CASCADE
  ON UPDATE CASCADE*),
  FOREIGN KEY (Date) REFERENCES SeasonChampion(Date)
  ON DELETE CASCADE
  ON UPDATE CASCADE*,
  FOREIGN KEY (lName) REFERENCES LeagueOrganises(lName)
  ON DELETE CASCADE
  ON UPDATE CASCADE*);
  - INSERT

INTO GameTeamBLeague(tNameB, cityB, Date, lName)
VALUES (‘Oilers’, ‘Edmonton’, ‘12-OCT-2022’, ‘National Hockey League’)

- INSERT
  INTO GameTeamBLeague(tNameB, cityB, Date, lName)
  VALUES (‘Canucks’, ‘Vancouver’, ‘22-OCT-2022’, ‘National Hockey League’)
- INSERT
  INTO GameTeamBLeague(tNameB, cityB, Date, lName)
  VALUES (‘Manchester City F.C.’, ‘’, ‘22-MAR-2018’, ‘Premier League’)
- INSERT
  INTO GameTeamBLeague(tNameB, cityB, Date, lName)
  VALUES (‘Manchester United F.C.’, ‘’, ‘04-FEB-2018’, ‘Premier League’)
- INSERT
  INTO GameTeamBLeague(tNameB, cityB, Date, lName)
  VALUES (‘Canucks’, ‘Vancouver’, ‘12-MAR-1999’, ‘National Hockey League’)

- CREATE TABLE GameTeamBOpponent(
  tNameB VARCHAR(255),
  cityB VARCHAR(255),
  Date DATE,
  tNameA VARCHAR(255) NOT NULL,
  PRIMARY KEY (tNameB, cityB, Date),
  FOREIGN KEY (tNameB, cityB) REFERENCES
  TeamHomeStadium(tName, city),
  ON DELETE CASCADE
  ON UPDATE CASCADE*),
  FOREIGN KEY (Date) REFERENCES SeasonChampion(Date),
  ON DELETE CASCADE
  ON UPDATE CASCADE*,
  FOREIGN KEY (tNameA) REFERENCES TeamHomeStadium(tNameA),
  ON DELETE CASCADE
  ON UPDATE CASCADE*);
  - INSERT
    INTO GameTeamBOpponent(tNameB, cityB, Date, tNameA)
    VALUES (‘Oilers’, ‘Edmonton’, ‘12-OCT-2022’, ‘Canucks’)
  - INSERT
    INTO GameTeamBOpponent(tNameB, cityB, Date, tNameA)
    VALUES (‘Canucks’, ‘Vancouver’, ‘22-OCT-2022’, ‘Kraken’)
  - INSERT
    INTO GameTeamBOpponent(tNameB, cityB, Date, tNameA)
    VALUES (‘Manchester City F.C.’, ‘’, ‘22-MAR-2018’, ‘Manchester United F.C.’)
  - INSERT
    INTO GameTeamBOpponent(tNameB, cityB, Date, tNameA)
    VALUES (‘Manchester United F.C.’, ‘’, ‘04-FEB-2018’, ‘Manchester City

F.C.')

- INSERT
  INTO GameTeamBOpponent(tNameB, cityB, Date, tNameA)
  VALUES ('Canucks', 'Vancouver', '12-MAR-1999', 'Oilers')

- CREATE TABLE sportsPerson (
  spID INTEGER,
  phoneNumber BIGINT UNIQUE,
  PRIMARY KEY(spID));
  - INSERT INTO sportsPerson VALUES (100012,9982991031);
  - INSERT INTO sportsPerson VALUES (10221,2232112313);
  - INSERT INTO sportsPerson VALUES (223422,4533513411);
  - INSERT INTO sportsPerson VALUES (233134,4224121445);
  - INSERT INTO sportsPerson VALUES (758230,2908394142);

- CREATE TABLE sportsPersonSalary (
  spID INTEGER,
  salary FLOAT,
  PRIMARY KEY(spID),
  FOREIGN KEY(spID) REFERENCES sportsPerson(spID) ON
  DELETE CASCADE);
  - INSERT INTO sportsPersonSalary VALUES(100012,23333332.24);
  - INSERT INTO sportsPersonSalary VALUES(223422,332424);
  - INSERT INTO sportsPersonSalary VALUES(758230,4758392);
  - INSERT INTO sportsPersonSalary VALUES(10221,245353242.2);
  - INSERT INTO sportsPersonSalary VALUES(233134,2242);

- CREATE TABLE sportsPersonNationality(
  spID INTEGER,
  nationality CHAR(255),
  PRIMARY KEY(spID),
  FOREIGN KEY(spID) REFERENCES sportsPerson(spID) ON
  DELETE CASCADE);
  - INSERT INTO sportsPersonNationality VALUES(100012,'China');
  - INSERT INTO sportsPersonNationality VALUES(223422,'Russia');
  - INSERT INTO sportsPersonNationality VALUES(10221,'Germany');
  - INSERT INTO sportsPersonNationality VALUES(233134,'Canada');
  - INSERT INTO sportsPersonNationality VALUES(758230,'Brazil');

- CREATE TABLE sportsPersonDOB(
  phoneNumber BIGINT,
  birthDate DATE NOT NULL,
  PRIMARY KEY(phoneNumber),
  FOREIGN KEY (phoneNumber) REFERENCES sportsPerson(phoneNumber)
  ON DELETE CASCADE ON UPDATE CASCADE);
  - INSERT INTO sportsPersonDOB VALUES (9982991031,'2000-10-12');
  - INSERT INTO sportsPersonDOB VALUES (2232112313,'2000-02-01');
  - INSERT INTO sportsPersonDOB VALUES (4533513411,'1980-10-11');

    - INSERT INTO sportsPersonDOB VALUES (4224121445,'1999-05-19');
    - INSERT INTO sportsPersonDOB VALUES (2908394142,'2001-04-01');

- CREATE TABLE sportsPersonPlayingSport(
    spID INTEGER,
    spName CHAR(255),
    PRIMARY KEY(spID),
    FOREIGN KEY(spID) REFERENCES sportsPerson(spID) ON DELETE CASCADE);
    - INSERT INTO sportsPersonPlayingSport VALUES (100012,'Basketball');
    - INSERT INTO sportsPersonPlayingSport VALUES (10221,'Soccer');
    - INSERT INTO sportsPersonPlayingSport VALUES (223422,'Volleyball');
    - INSERT INTO sportsPersonPlayingSport VALUES (233134,'Hockey');
    - INSERT INTO sportsPersonPlayingSport VALUES (758230,'Cricket');

- CREATE TABLE sportsPersonDeathDate(
    spID INTEGER,
    deathDate DATE,
    PRIMARY KEY(spID),
    FOREIGN KEY (spID) REFERENCES sportsPerson(spID) ON DELETE CASCADE);
    - INSERT INTO sportsPersonDeathDate VALUES (100012,NULL);
    - INSERT INTO sportsPersonDeathDate VALUES (10221,NULL);
    - INSERT INTO sportsPersonDeathDate VALUES (223422,'2013-11-20');
    - INSERT INTO sportsPersonDeathDate VALUES (233134,NULL);
    - INSERT INTO sportsPersonDeathDate VALUES (758230,NULL);

- CREATE TABLE sportsPersonAge (
    birthDate DATE,
    deathDate DATE,
    age INTEGER NOT NULL,
    PRIMARY KEY (birthDate, deathDate),
    FOREIGN KEY (birthDate) REFERENCES sportsPersonDOB(birthDate),
        FOREIGN KEY (deathDate) REFERENCES sportsPersonDeathDate(deathDate));

    - INSERT INTO sportsPersonAge VALUES ('2000-10-12', NULL, 23);
    - INSERT INTO sportsPersonAge VALUES ('2001-04-02', NULL, 22);
    - INSERT INTO sportsPersonAge VALUES ('1999-05-23', NULL, 24);
    - INSERT INTO sportsPersonAge VALUES ('1980-10-11', '2013-11-12', 33);
    - INSERT INTO sportsPersonAge VALUES ('2000-02-01', NULL, 23);

- CREATE TABLE Athlete (
    spID INTEGER,
    weight FLOAT,
    height FLOAT,
    jerseyID INTEGER,

numberOfMVPs INTEGER,
PRIMARY KEY (spID),
FOREIGN KEY (spID) REFERENCES sportsPerson(spID) ON DELETE CASCADE);

- INSERT INTO Athlete VALUES(1223,80,167,21,2);
- INSERT INTO Athlete VALUES(4323232,100,200,20,0);
- INSERT INTO Athlete VALUES(623232,139,180,1,1);
- INSERT INTO Athlete VALUES(782298,75,170,122,1);
- INSERT INTO Athlete VALUES(77,80,169,99,0);

- CREATE TABLE Coach (
  spID INTEGER,
  yearsCoaching INTEGER,
  numberOfChampionships INTEGER,
  PRIMARY KEY (spID),
  FOREIGN KEY (spID) REFERENCES sportsPerson(spID) ON DELETE CASCADE);
  - INSERT INTO Coach VALUES(12231,10,2);
  - INSERT INTO Coach VALUES(22,2,2);
  - INSERT INTO Coach VALUES(22324,30,4);
  - INSERT INTO Coach VALUES(45321,20,1);
  - INSERT INTO Coach VALUES(5231,10,5);

- CREATE TABLE Owner (
  spID INTEGER,
  netWorth FLOAT,
  PRIMARY KEY (spID),
  FOREIGN KEY (spID) REFERENCES sportsPerson(spID) ON DELETE CASCADE);
  - INSERT INTO Owner VALUES(346473,773465863);
  - INSERT INTO Owner VALUES(234,23333);
  - INSERT INTO Owner VALUES(5345,787823);
  - INSERT INTO Owner VALUES(3233,900);
  - INSERT INTO Owner VALUES(14325,283283898);