Student Names and Student IDs
Morgan Law #37577153
Emil Aberg #92305986
Naman Bansal #29328144
Thomas Welsh #41953944

Project: Database Schema and SQL DDL
Note: This document is laid out in the form (schema, DDL) for every relation
CHANGES TO FINAL VERSION: country attribute was added to athlete to allow sorting by country for specific queries

All relations are already in BCNF and 3NF form because there are no functional dependencies among the attributes in any given relations. No attribute(s) can determine another apart from the key of each relation.

Sport(sportID: integer, name: char(20))

- Represents entity Sports
- Primary key: sportID

```
CREATE TABLE Sport(
        sportID INTEGER,
        name CHAR(20),
        PRIMARY KEY (sportID))
```

SBSport(**sportID**: integer, WR: float, unit: char(6))

- Represents entity SBSport (score-based sports)
- Primary key: sportID
- Unit is one of (cm, m, points)
- Foreign keys:
    - sportID references Sport

```
CREATE TABLE SBSport(
        sportID INTEGER,
        WR FLOAT,
        unit CHAR(6),
        PRIMARY KEY (sportID),
        FOREIGN KEY (sportID) REFERENCES Sport
        )
```

TBSport(**sportID**: integer, WR: time)

- Represents entity TBSport (time-based sports)
- Primary key: sportID
- Foreign keys:
    - sportID references Sport

```
CREATE TABLE TBSport(
```

sportID INTEGER,
WR TIME,
PRIMARY KEY (sportID),
FOREIGN KEY (sportID) REFERENCES Sport)

Division(<u>divisionID</u>: integer, maxAge: integer, minAge: integer, gender: string)

- Represents Division entity
- Primary key: divisionID
- minAge && maxAge > 0, maxAge >= minAge
- gender is one of {M, F, O} (male, female, other)

```
CREATE TABLE Division(
    divisionID INTEGER,
    maxAge INTEGER,
    minAge INTEGER,
    gender CHAR(1),
    PRIMARY KEY (divisionID)
    )
```

Competition(<u>competitionID</u>: integer, name: char(40), startDate: date, endDate: date, inviteOnly: boolean, **country**: char(20), **province_state**: char(20), **city**: char(20)**,** **competitionOrganizerID**: char(20)

- Represents competition entity
- inviteOnly indicates whether or not only invited athletes can compete
- Primary key: competitionID
- Foreign keys:
    - (Country, province_state, city) reference Location - cannot be NULL
    - competitionOrganizerID references Person - cannot be NULL

```
CREATE TABLE Competition(
    competitionID INTEGER,
    name CHAR(40),
    startDate DATE,
    endDate DATE,
    inviteOnly BOOLEAN,
    country CHAR(20) NOT NULL,
    province_state CHAR(20) NOT NULL,
    city CHAR(20) NOT NULL,
    competitionOrganizerID INTEGER NOT NULL,
    PRIMARY KEY (competitionID),
    FOREIGN KEY (country, province_state, city) REFERENCES Location
        ON DELETE NO ACTION
        ON UPDATE CASCADE
    FOREIGN KEY (competitionOrganizerID) REFERENCES Person
        ON DELETE NO ACTION
        ON UPDATE CASCADE)  )
```

HasTB(spots: integer, **sportID**: integer, **divisionID**: integer, **competitionID**: integer)

- Represents hasTB relation/aggregation
- Primary key: (sportID, divisionID, competitionID)
- spots must be > 0
- Foreign keys:
    - sportID references TBSport
    - divisionID references Division
    - competitionID references Competition

```
CREATE TABLE HasTB(
        spots INTEGER,
        sportID INTEGER,
        divisionID INTEGER,
        competitionID INTEGER,
        PRIMARY KEY (sportID, divisionID, competitionID),
        FOREIGN KEY (sportID) references TBSport
                ON DELETE CASCADE
        FOREIGN KEY (divisionID) references Division
                ON DELETE CASCADE
        FOREIGN KEY (competitionID) references Competition
                ON DELETE CASCADE)
```

HasSB(spots: integer, **sportID**: integer, **divisionID**: integer, **competitionID**: integer)

- Represents hasSB entity
- Primary key: (sportID, divisionID, competitionID)
- Spots must be > 0
- Foreign keys:
    - sportID references SBSport
    - divisionID references Division
    - competitionID references Competition

```
CREATE TABLE HasSB(
        spots INTEGER,
        sportID INTEGER,
        divisionID INTEGER,
        competitionID INTEGER,
        PRIMARY KEY (sportID, divisionID, competitionID),
        FOREIGN KEY (sportID) REFERENCES SBSport
                ON DELETE CASCADE
        FOREIGN KEY (divisionID) REFERENCES Division
                ON DELETE CASCADE
        FOREIGN KEY (competitionID) REFERENCES Competition
                ON DELETE CASCADE)
```

ParticipatesTB(status: char(20), result: time, **athleteID**: integer, **sportID**: integer**,** **divisionID**: integer, **competitionID**: integer)

- Represents participatesTB entity
- Primary key: (athleteID, sportID, divisionID, competitionID)
- Foreign keys:
    - athleteID references athlete
    - sportID references TBSport
    - divisionID references Division
    - competitionID references Competition

```
CREATE TABLE ParticipatesTB(
        status CHAR(20),
        result TIME,
        athleteID INTEGER,
        sportID INTEGER,
        divisionID INTEGER,
        competitionID INTEGER,
        PRIMARY KEY (athleteID, sportID, divisionID, competitionID),
        FOREIGN KEY (athleteID) REFERENCES Athlete
                ON DELETE CASCADE
        FOREIGN KEY (sportID) REFERENCES TBSport
                ON DELETE CASCADE
        FOREIGN KEY (divisionID) REFERENCES Division
                ON DELETE CASCADE
        FOREIGN KEY (competitionID) REFERENCES Competition
                ON DELETE CASCADE)
```

ParticipatesSB(status: char(20), result: time, **athleteID**: integer, **sportID**: integer**,** **divisionID**: integer, **competitionID**: integer)

- Represents participatesSB entity
- Primary key: (athleteID, sportID, divisionID, competitionID)
- Foreign keys:
    - athleteID references athlete
    - sportID references SBSport
    - divisionID references Division
    - competitionID references Competition

```
CREATE TABLE ParticipatesSB(
        status CHAR(20),
        result FLOAT,
        athleteID INTEGER,
        sportID INTEGER,
        divisionID INTEGER,
        competitionID INTEGER,
```

```
    PRIMARY KEY (athleteID, sportID, divisionID, competitionID),
    FOREIGN KEY (athleteID) REFERENCES Athlete
        ON DELETE CASCADE
    FOREIGN KEY (sportID) REFERENCES SBSport
        ON DELETE CASCADE
    FOREIGN KEY (divisionID) REFERENCES Division
        ON DELETE CASCADE
    FOREIGN KEY (competitionID) REFERENCES Competition
        ON DELETE CASCADE)
```

Rank(**athleteID**: integer, **sportID**: integer, points: integer)

- Represents rank entity
- Primary key: (athleteID, sportID)
- Foreign keys:
    - sportID references Sport
    - athleteID references Person

```
CREATE TABLE Rank(
    athleteID INTEGER,
    sportID INTEGER,
    points INTEGER,
    PRIMARY KEY (athleteID, sportID),
    FOREIGN KEY (athleteID) REFERENCES Athlete
        ON DELETE CASCADE
    FOREIGN KEY (sportID) REFERENCES Sport
    )
```

Teaches(**coachID**: integer, **sportID**: integer)

- Represents Teaches relationship
- Primary key: (coachID, sportID)
- Foreign keys:
    - sportID references Sport
    - coachID references Coach

```
CREATE TABLE teaches(
    coachID INTEGER,
    sportID INTEGER,
    PRIMARY KEY (coachID, sportID),
    FOREIGN KEY (coachID) REFERENCES coach
        ON DELETE CASCADE
    FOREIGN KEY (sportID) REFERENCES Sport
    )
```

RequiresTB(**testID:**integer, **sportID:**integer**, divisionID:**integer, **competitionID:**integer)

- Represents requiresTB relationship
- Primary key: (testID, sportID, divisionID, competitionID)
- Foreign keys:
    - testID references DrugTest
    - sportID references TBSport
    - divisionID references Division
    - competitionID references Competition

```
CREATE TABLE requiresTB(
        testID INTEGER,
        sportID INTEGER,
        divisionID INTEGER,
        competitionID INTEGER,
        PRIMARY KEY (testID, sportID, divisionID, competitionID),
        FOREIGN KEY (testID) REFERENCES test
                ON DELETE CASCADE
        FOREIGN KEY (sportID) REFERENCES TBSport
                ON DELETE CASCADE
        FOREIGN KEY (divisionID) REFERENCES competition
                ON DELETE CASCADE
        FOREIGN KEY (competitionID) REFERENCES competition
                ON DELETE CASCADE)
```

RequiresSB(**testID**:integer, **sportID:**integer**, divisionID:**integer, **competitionID:**integer)

- Represents requiresSB relationship
- Primary key: (testID, sportID, divisionID, competitionID)
- Foreign keys:
    - testID references DrugTest
    - sportID references SBSport
    - divisionID references Division
    - competitionID references Competition

```
CREATE TABLE requiresSB(
        testID INTEGER,
        sportID INTEGER,
        divisionID INTEGER,
        competitionID INTEGER,
        PRIMARY KEY (testID, sportID, divisionID, competitionID),
```

```
    FOREIGN KEY (testID) REFERENCES Test
        ON DELETE CASCADE
    FOREIGN KEY (sportID) REFERENCES SBSport
    FOREIGN KEY (divisionID) REFERENCES Division
        ON DELETE CASCADE
    FOREIGN KEY (competitionID) REFERENCES competition
        ON DELETE CASCADE)
```

For(**competitionID:**integer, **teamID:**integer)

- Represents for relationship
- Primary key: (competitionID, teamID)
- Foreign keys:
    - competitionID references Competition
    - teamID references Team

```
CREATE TABLE For(
    competitionID INTEGER,
    teamID INTEGER,
    PRIMARY KEY (competitionID, teamID),
    FOREIGN KEY (competitionID) REFERENCES Competition
        ON DELETE CASCADE
    FOREIGN KEY (teamID) REFERENCES Team
        ON DELETE CASCADE)
```

Location(country: char(40), province_state: char(40), city: char(40))

- Represents Location entity
- Primary key: (country, province_state, city)

```
CREATE TABLE location(
    country CHAR(40),
    province_state CHAR(40),
    city CHAR(40),
    PRIMARY KEY (country, province_state, city)
)
```

Team(teamID:integer, name: char(20), **country:** char(40)**, province_state:** char(40)**, city:** char(40))

- Represents team entity
- Primary key: teamID
- Foreign keys:

    - (country, province_state, city) references Location - cannot be NULL

```
CREATE TABLE Team(
        teamID INTEGER,
        name CHAR(20),
        country CHAR(40) NOT NULL,
        province_state CHAR(40) NOT NULL,
        city CHAR(40) NOT NULL,
        PRIMARY KEY (teamID),
        FOREIGN KEY (country, province_state, city) REFERENCES Location
                ON DELETE NO ACTION
                ON UPDATE CASCADE
)
```

Person(personID:integer, name: char(40), gender: char(1), birthDate:date, isCompetitionManager: boolean)

- Represents Person entity
- Primary key: personID
- gender is one of {M, F, O}
- isCompetitionManager accounts for competition managers

```
CREATE TABLE Person(
        personID INTEGER,
        name CHAR(40),
        gender CHAR(1),
        birthDate DATE,
        isCompetitionManager BOOLEAN,
        PRIMARY KEY (personID)
        )
```

Athlete(weight: integer, height: integer, **personID:** integer, **teamID:** integer, country: char(20))

- Represents athlete entity
- Primary key: personID
- Foreign keys:
    - personID references Person
    - teamID references Team

```
CREATE TABLE Athlete(
        weight INTEGER,
        height INTEGER,
        personID INTEGER,
        teamID INTEGER,
        country CHAR(20),
```

```
    PRIMARY KEY (personID),
    FOREIGN KEY (personID) REFERENCES Person
        ON DELETE CASCADE
    FOREIGN KEY (teamID) REFERENCES Team
        ON DELETE SET NULL
    )
```

Coach(experienceLevel: char(40), **personID:** integer, **country:** char(40)**, province_state:** char(40)**, city:** char(40))

- Represents Coach entity
- Primary key: personID
- Foreign keys:
    - personID references Person
    - (country, province_state, city) references Location - cannot be NULL

```
CREATE TABLE Coach(
    personID INTEGER,
    experienceLevel CHAR(40),
    country CHAR(40) NOT NULL,
    province_state CHAR(40) NOT NULL,
    city CHAR(40) NOT NULL,
    PRIMARY KEY (personID),
    FOREIGN KEY (personID) references person
        ON DELETE CASCADE
    FOREIGN KEY (country, province_state, city) REFERENCES Location
        ON DELETE NO ACTION
        ON UPDATE CASCADE
    )
```

TrainingPlan(trainingID: integer, startDate: date, endDate: date, instructions: text, **coachID:** integer**, athleteID:** integer)

- Represents TrainingPlan entity
- Primary key: trainingID
- Foreign keys:
    - coachID references Coach - cannot be NULL
    - athleteID references Athlete - cannot be NULL

```
CREATE TABLE TrainingPlan(
    trainingID INTEGER,
    startDate DATE,
    endDate DATE,
    instructions TEXT,
```

```
    coachID INTEGER,
    athleteID INTEGER,
    PRIMARY KEY (trainingID),
    FOREIGN KEY  (coachID) REFERENCES Coach
        ON DELETE CASCADE
    FOREIGN KEY (athleteID) REFERENCES Athlete
        ON DELETE CASCADE
    )
```

DrugTest(testID:integer, testName:char(20))

- Represents DrugTest entity
- Primary key: testID

```
CREATE TABLE  DrugTest(
    testID   INTEGER,
    testName CHAR(20),
    PRIMARY KEY (testID)
    )
```

AthleteTakesTest(expDate: date, result: char(40), **athleteID:** integer, **testID:** integer)

- Represents Takes relationship
- Primary key: athleteID, testID
- Foreign keys:
    - athleteID references Athlete
    - testID references DrugTest

```
CREATE TABLE AthleteTakesTest(
    expDate DATE,
    result CHAR(40),
    athleteID INTEGER,
    testID  INTEGER,
    PRIMARY KEY (athleteID,testID),
    FOREIGN KEY  (athleteID) REFERENCES Athlete
        ON DELETE CASCADE
    FOREIGN KEY  (testID)  REFERENCES DrugTest
        ON DELETE CASCADE
    )
```

invitesTB(**athleteID:** integer, **competitionManagerID:** integer, **sportID:** integer, **divisionID:** integer, **competitionID:** integer)

- Represents invitesTB relationship

- Primary key: (athleteID, competitionManagerID, sportID, divisionID, competitionID)
- Foreign keys:
    - athleteID references Athlete
    - competitionManagerID references Person
    - sportID references TBSport
    - divisionID references Division
    - competitionID references Competition

CREATE TABLE invitesTB(
competitionManagerID INTEGER,
athleteID INTEGER,
sportID INTEGER,
divisionID INTEGER,
competitionID INTEGER,
PRIMARY KEY (competitionManagerID, athleteID, sportID, divisionID, competitionID)
FOREIGN KEY (competitionManagerID) REFERENCES Person
FOREIGN KEY (athleteID) REFERENCES Athlete
FOREIGN KEY (sportID) REFERENCES TBSport
FOREIGN KEY (divisionID) REFERENCES Division
FOREIGN KEY (competitionID) REFERENCES Competition)

invitesSB(**athleteID:** integer, **competitionManagerID**: integer, **sportID:** integer, **divisionID:** integer, **competitionID:** integer)

- Represents invitesSB relationship
- Primary key: (athleteID, competitionManagerID, sportID, divisionID, competitionID)
- Foreign keys:
    - athleteID references Athlete
    - competitionManagerID references Person
    - sportID references SBSport
    - divisionID references Division
    - competitionID references Competition

CREATE TABLE invitesSB(
competitionManagerID INTEGER,
athleteID INTEGER,
sportID INTEGER,
divisionID INTEGER,
competitionID INTEGER,
PRIMARY KEY (competitionManagerID, athleteID, sportID, divisionID, competitionID)
FOREIGN KEY (competitionManagerID) REFERENCES Person
FOREIGN KEY (athleteID) REFERENCES Athlete
FOREIGN KEY (sportID) REFERENCES SBSport
FOREIGN KEY (divisionID) REFERENCES Division
FOREIGN KEY (competitionID) REFERENCES Competition)