

CPSC 304 Project Cover Page

Milestone #: 02

Date: October 20th 2023

Group Number: 76

Name	Student Number	CS Alias (Userid)	Preferred Email Address
Sneha Mungre	57348765	y6d0f	sneha051@student.ubc.ca
Surabhi Nag	64692726	n0s6i	snag2001@student.ubc.ca
Utsav Singh	62865753	m5c8m	utsav02@student.ubc.ca

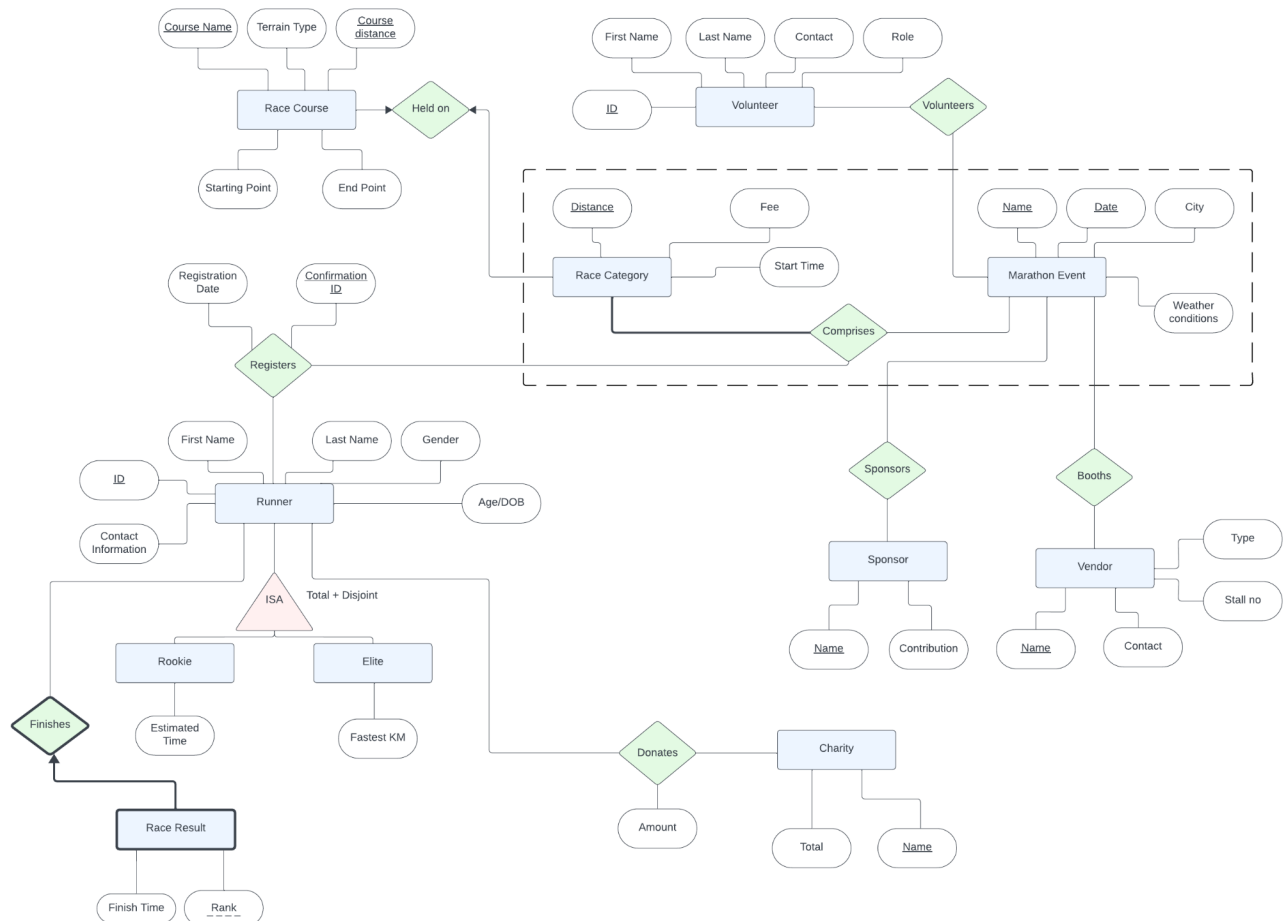
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your email address, 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 Summary:

We are building a Race Event and Logistics Manager application. This would help Runners get access to the result of past races and donations. Organizers are able to keep track of each race, race course, volunteers, runners, sponsors.

ER Diagram:



Changes made from Milestone 1:

- Addition of 'Estimated Time' Attribute to **Rookie**
- Change 'Personal Record' Attribute to 'Fastest KM' in **Elite**
- Addition of 'Weather Conditions' Attribute to **Marathon Event**
- Addition of an aggregation around **Race Category** and **Marathon Event**
- Addition of new entity **Vendor**
- Removal of **Registration** and replacing it with a relation connecting the aggregate to **Runner**
- **Comprises** relation is a many to many rather than a many to one

Table Schema and Constraints:

1. volunteer (
 volunteerID:INTEGER,
 firstName:VARCHAR[20],
 lastName:VARCHAR[20],
 volunteerContact:VARCHAR[20],
 volunteerRole:VARCHAR[20]
)
 - volunteerID - PK, NOT NULL
 - firstName, lastName, volunteerContact - NOT NULL
 - We assume that volunteerRole doesn't have the NOT NULL constraint as they might not be assigned one from the beginning.

2. volunteers (
 volunteerID:INTEGER,
 marathonName:VARCHAR[20],
 marathonDate:DATE
)
 - volunteerID, marathonName, marathonDate - PK, NOT NULL automatically and all three are foreign keys referencing attributes either in volunteer or in marathonEvent.

3. marathonEvent (
 marathonName:VARCHAR[20],
 marathonDate:DATE,
 city:VARCHAR[20],
 weatherConditions:VARCHAR[20]
)
 - marathonName, marathonDate - PK, NOT NULL
 - city - NOT NULL
 - weatherConditions can be NULL if they are not known

4. sponsors (
 marathonName:VARCHAR[20],
 marathonDate:DATE,
 sponsorName:VARCHAR[20]
)
 - marathonName, marathonDate, sponsorName - PK, NOT NULL automatically and all three are foreign keys referencing attributes either in sponsor or in marathonEvent.

5. sponsor (
 sponsorName:VARCHAR[20],
 contribution: FLOAT
)
 - sponsorName - PK. Assumption: Sponsor names will be unique in the real world.
 - contribution - NOT NULL, greater than 0

6. raceCategory (
 distance:INTEGER,
 fee:INTEGER,
 startTime:DATETIME
)
 - distance - PK, NOT NULL, distance in km greater than 0
 - fee - NOT NULL, greater than 0
 - startTime - NOT NULL

7. marathonEvent_comprises_raceCategory (
 distance:INTEGER,
 marathonName:VARCHAR[20],
 marathonDate:DATE
)
 - distance, marathonName, marathonDate - PK, NOT NULL automatically and all three are foreign keys referencing attributes either in raceCategory or in marathonEvent

8. raceCourse_category (
 courseName:VARCHAR[20],
 terrainType:VARCHAR[20],
 courseDistance:INTEGER,
 stP:VARCHAR[20], *This is the start point*
 endP:VARCHAR[20], *This is the end point*
 raceCategoryDistance:INTEGER
)
 - courseName, courseDistance - PK, NOT NULL
 - stP, EndP - NOT NULL, raceCategoryDistance will be FK (raceCategory table)
 - terrainType can be NULL because some terrains are mixed

This is the table for a one to one relationship due to which the names have been combined in the schema

9. charity (
 charityName:VARCHAR[20],
 total:INTEGER
)
 - name - PK, NOT NULL. Assumption: Charity Names will be Unique in real world
 - total - NOT NULL, Greater than 0.

10. donates (

amount: FLOAT,
runnerID:INTEGER,
charityName:VARCHAR[20]
)

- amount, runnerID, charityName - NOT NULL. amount greater than 0. PK: runnerID, charityName

11. registers (

confirmationID:INTEGER,
registrationDate:DATE,
runnerID:INTEGER,
distance:INTEGER,
marathonName:VARCHAR[20],
marathonDate:DATE
)

- confirmationID, runnerID, marathonName, marathonDate - PK, NOT NULL
- distance - distance in km greater than 0
- registrationDate - NOT NULL

12. elite (

runnerID:INTEGER,
contact:VARCHAR[20],
firstName:VARCHAR[20],
lastName:VARCHAR[20],
gender:VARCHAR[20],
Age:INTEGER,
fastestKM:TIMESTAMP)

- runnerID - PK, NOT NULL
- contact, firstName, lastName, age, - NOT NULL
- age - greater than 8 (there is an age limit in order to enter the races)

13. rookie (

runnerID:INTEGER,
contact:VARCHAR[20],
firstName:VARCHAR[20],
lastName:VARCHAR[20],
gender:VARCHAR[20],
age:INTEGER
estimatedTime:TIMESTAMP)

- runnerID - PK, NOT NULL
- contact, firstName, lastName, age - NOT NULL

14. runnerResult(

rank:INTEGER,
finishTime:TIMESTAMP,
runnerID:INTEGER
)

- rank, finishTime: NOT NULL, greater than 0
- runnerID: INTEGER: NOT NULL. Foreign Key

15. Vendor (

name:INTEGER,
contact:VARCHAR[20],
stallNo:VARCHAR[20],
type: :VARCHAR[20],
)

- Name PK, contact and Type: NOT NULL
- Stall No. can be null as a vendor might not have stall

16. Booths (

vendorName:INTEGER,
marathonName:VARCHAR[20],
marathonDate:DATE,
)

- vendorName, marathonName, marathonDate - PK, NOT NULL automatically and all three are foreign keys referencing attributes either in vendor or in marathonEvent

Functional Dependencies:

1. volunteer
 - $\text{volunteerID} \rightarrow \text{firstName}, \text{lastName}, \text{volunteerContact}, \text{volunteerRole}$
 - $\text{volunteerContact} \rightarrow \text{firstName}, \text{lastName}, \text{volunteerRole}$
2. volunteers
3. marathonEvent
 - $\text{marathonName} \rightarrow \text{marathonDate}, \text{city}, \text{weatherConditions}$
 - $\text{marathonName}, \text{marathonDate} \rightarrow \text{city}$
 - $\text{city}, \text{marathonDate} \rightarrow \text{weatherConditions}$
4. sponsors
5. sponsor
 - $\text{sponsorName} \rightarrow \text{contribution}$
6. raceCategory
 - $\text{distance} \rightarrow \text{fee}, \text{startTime}$
7. marathonEvent_comprises_raceCategory
8. raceCourse_category
 - $\text{courseName}, \text{courseDistance} \rightarrow \text{terrainType}, \text{stP}, \text{endP}, \text{raceCategoryDistance}$
 - $\text{stP}, \text{endP} \rightarrow \text{courseDistance}$
 - $\text{courseName} \rightarrow \text{terrainType}$
 - $\text{raceCategoryDistance} \rightarrow \text{courseDistance}$
 - $\text{courseDistance} \rightarrow \text{raceCategoryDistance}$
9. charity
 - $\text{name} \rightarrow \text{total}$
10. donates
 - $\text{runnerID}, \text{charityName} \rightarrow \text{amount}$
11. registers
 - $\text{confirmationID} \rightarrow \text{registrationDate}, \text{runnerID}, \text{distance}, \text{marathonName}, \text{marathonDate}$
 - $\text{registrationDate}, \text{runnerID}, \text{marathonName}, \text{marathonDate} \rightarrow \text{confirmationID}$
12. elite
 - $\text{runnerID} \rightarrow \text{contact}, \text{firstName}, \text{lastName}, \text{gender}, \text{age}, \text{fastestKM}$
 - $\text{contact} \rightarrow \text{runnerID}, \text{firstName}, \text{lastName}, \text{gender}, \text{age}, \text{fastestKM}$
13. rookie
 - $\text{runnerID} \rightarrow \text{contact}, \text{firstName}, \text{lastName}, \text{gender}, \text{age}$
 - $\text{contact} \rightarrow \text{runnerID}, \text{firstName}, \text{lastName}, \text{gender}, \text{age}$
14. runnerResult
 - $\text{runnerID}, \text{rank} \rightarrow \text{finishTime}$
 - $\text{finishTime} \rightarrow \text{rank}$
15. vendor
 - $\text{name} \rightarrow \text{contact}, \text{stallNo}, \text{type}$
 - $\text{stallNo} \rightarrow \text{name}$
 - $\text{contact} \rightarrow \text{name}, \text{stallNo}, \text{type}$
16. booths

Normalization:

Normalize each of your tables to be in 3NF or BCNF. Give list of tables, their PKs and CKs and FKs after normalization

1. volunteer (volunteerID, firstName, lastName, volunteerContact, volunteerRole)

Minimal Cover (not in 3NF):

1. Put FDs in standard form:

volunteerID \rightarrow firstName

volunteerID \rightarrow lastName

volunteerID \rightarrow volunteerContact

volunteerID \rightarrow volunteerRole

volunteerContact \rightarrow firstName

volunteerContact \rightarrow lastName

volunteerContact \rightarrow volunteerRole

2. Minimize LHS: Not possible

3. Delete Redundant FDs:

volunteerID \rightarrow volunteerContact

volunteerContact \rightarrow firstName

volunteerContact \rightarrow lastName

volunteerContact \rightarrow volunteerRole

Decomposition into 3NF using synthesis method:

VolunteerContact(volunteerID, volunteerContact)

VolunteerFirstName(volunteerContact, firstName)

VolunteerLastName(volunteerContact, lastName)

VolunteerRole(volunteerContact, volunteerRole)

2. marathonEvent

Minimal Cover (not in 3NF)

1. Put FDs in standard form

marathonName, marathonDate \rightarrow city

marathonName, marathonDate \rightarrow weatherConditions

city, marathonDate \rightarrow weatherConditions

2. Minimize LHS - not possible to minimize

3. Delete Redundant FDs

marathonName, marathonDate \rightarrow city

city, marathonDate \rightarrow weatherConditions

Decomposition into 3NF using synthesis method:

MarathonCity(marathonName, marathonDate, city),

MarathonWeather(city, marathonDate, weatherConditions)

3. sponsor

sponsorName \rightarrow contribution

Already in 3NF as sponsorName is a key for the relation

Sponsor(sponsorName, contribution)

4. raceCategory

- distance \rightarrow fee, startTime

Already in 3NF as distance is a key for the relation

RaceCategory(distance, fee, startTime)

5. raceCourse_category

Minimal cover (for 3NF)

Standard form:

- courseName, courseDistance \rightarrow terrainType
- courseName, courseDistance \rightarrow stP
- courseName, courseDistance \rightarrow endP
- courseName, courseDistance \rightarrow raceCategoryDistance
- stP, endP \rightarrow courseDistance
- courseName \rightarrow terrainType
- courseDistance \rightarrow raceCategoryDistance

Minimize LHS:

- courseName, ~~courseDistance~~ \rightarrow terrainType
- ~~courseName~~, courseDistance \rightarrow raceCategoryDistance

Delete Redundant FDs:

- courseName \rightarrow terrainType
- courseName, courseDistance \rightarrow stP
- courseName, courseDistance \rightarrow endP
- stP, endP \rightarrow courseDistance
- courseDistance \rightarrow raceCategoryDistance

Decomposition into 3NF using synthesis method

CourseNameDistance(courseDistance, courseName),

CourseTerrain(courseName, terrainType),

CourseStartPoint(courseName, courseDistance, stP),

CourseEndPoint(courseName, courseDistance, endP),

CourseCategoryDistance(courseDistance, raceCategoryDistance),

CourseDistance(stP, endP, courseDistance),

6. charity

- charityName \rightarrow total

Already in 3NF as name is a superkey for the relation

Charity(charityName, total)

7. donates

- runnerID, charityName \rightarrow amount

Already in 3NF as runnerID, charityName is a superkey for the relation

Donation(runnerID, charityName, amount)

8. registers

- confirmationID \rightarrow registrationDate, runnerID, distance, marathonName, marathonDate
- registrationDate, runnerID, marathonName, marathonDate \rightarrow confirmationID

Already in 3NF as confirmationID is a superkey for the relation

Registration(confirmationID,registrationDate,runnerID, distance,
marathonName, marathonDate)

9. elite

Minimal cover (for 3NF)

Standard form:

- runnerID \rightarrow firstName
- runnerID \rightarrow lastName
- runnerID \rightarrow gender
- runnerID \rightarrow age
- runnerID \rightarrow contact
- runnerID \rightarrow fastestKM
- contact \rightarrow firstName
- contact \rightarrow lastName
- contact \rightarrow gender
- contact \rightarrow age
- contact \rightarrow fastestKM

Minimize LHS: not possible

Delete Redundant FDs:

- runnerID \rightarrow contact
- contact \rightarrow firstName
- contact \rightarrow lastName
- contact \rightarrow gender
- contact \rightarrow age
- contact \rightarrow fastestKM

Decomposition into 3NF using synthesis method

EliteContact(runnerID, contact) ,
EliteFirstName(contact, firstName) ,
EliteLastName(contact, lastName),
EliteGender(contact, gender),
EliteAge(contact, age),
EliteFastestKm(contact, fastestKM)

10. rookie

Minimal cover (for 3NF)

Standard form:

- runnerID \rightarrow firstName
- runnerID \rightarrow lastName
- runnerID \rightarrow gender
- runnerID \rightarrow age
- runnerID \rightarrow contact
- runnerID \rightarrow estimatedTime
- contact \rightarrow firstName
- contact \rightarrow lastName
- contact \rightarrow gender
- contact \rightarrow age
- contact \rightarrow estimatedTime

Delete Redundant FDs:

- runnerID \rightarrow contact
- contact \rightarrow firstName
- contact \rightarrow lastName
- contact \rightarrow gender
- contact \rightarrow age
- contact \rightarrow estimatedTime

Decomposition into 3NF using synthesis method

RookieContact(runnerID, contact) ,
RookieFirstName(contact, firstName) ,
RookieLastName(contact, lastName),
RookieGender(contact, gender),
RookieAge(contact, age),
RookieEstimateTime(contact, estimatedTime)

11. runnerResult

- o runnerID, rank \rightarrow finishTime
- o finishTime \rightarrow rank

Already in 3NF as runnerID, rank is a superkey for the relation
Result(runnerID, rank, finishTime)

12. vendor

- o name \rightarrow contact, stallNo, type
- o stallNo \rightarrow name
- o contact \rightarrow name, stallNo, type

Already in 3NF as name is a superkey for the relation
Vendor(name, contact, stallNo, type)

13. booths

Booths(vendorName, marathonName, marathonDate)

14. sponsors

Sponsors(sponsorName, marathonName, marathonDate)

15. volunteers

Volunteers(volunteerID, marathonName, marathonDate)

16. marathonEvent_comprises_raceCategory

MarathonEventRaceCategory(categoryDistance, marathonName,
marathonDate)

DDL Statements:

1. VolunteerContact

```
CREATE TABLE VolunteerContact (  
    volunteerID INTEGER PRIMARY KEY,  
    volunteerContact VARCHAR(20) NOT NULL UNIQUE  
);
```

2. VolunteerFirstName

```
CREATE TABLE VolunteerFirstName (  
    volunteerContact VARCHAR(20) NOT NULL,  
    firstName VARCHAR(20) NOT NULL,  
    PRIMARY KEY (volunteerContact),  
    FOREIGN KEY (volunteerContact) REFERENCES  
VolunteerContact(volunteerContact) ON UPDATE CASCADE  
);
```

3. VolunteerLastName

```
CREATE TABLE VolunteerLastName (  
    volunteerContact VARCHAR(20) NOT NULL,  
    lastName VARCHAR(20) NOT NULL,  
    PRIMARY KEY (volunteerContact),  
    FOREIGN KEY (volunteerContact) REFERENCES  
VolunteerContact(volunteerContact) ON UPDATE CASCADE  
);
```

4. VolunteerRole

```
CREATE TABLE VolunteerRole (  
    volunteerContact VARCHAR(20) NOT NULL,  
    volunteerRole VARCHAR(20) NOT NULL,  
    PRIMARY KEY (volunteerContact),  
    FOREIGN KEY (volunteerContact) REFERENCES  
VolunteerContact(volunteerContact) ON UPDATE CASCADE  
);
```

5. MarathonCity

```
CREATE TABLE MarathonCity (  
    marathonName VARCHAR(20) NOT NULL,  
    marathonDate DATE NOT NULL,  
    city VARCHAR(20) NOT NULL,  
    PRIMARY KEY (marathonName, marathonDate)  
);
```

6. MarathonWeather

```
CREATE TABLE MarathonWeather (  
    city VARCHAR(20) NOT NULL,  
    marathonDate DATE NOT NULL,  
    weatherConditions VARCHAR(50),  
    PRIMARY KEY (city, marathonDate),  
    FOREIGN KEY (city, marathonDate) REFERENCES MarathonCity(city,  
marathonDate) ON UPDATE CASCADE  
);
```

7. Sponsor

```
CREATE TABLE Sponsor (  
    name VARCHAR(20) PRIMARY KEY,  
    contribution INTEGER  
);
```

8. RaceCategory

```
CREATE TABLE RaceCategory (  
    categoryDistance INTEGER PRIMARY KEY,  
    fee INTEGER,  
    startTime DATETIME  
);
```

9. CourseNameDistance

```
CREATE TABLE CourseNameDistance (  
    courseDistance INTEGER,  
    courseName VARCHAR(20),  
    PRIMARY KEY (courseDistance, courseName)  
);
```

10. CourseTerrain

```
CREATE TABLE CourseTerrain (  
    courseName VARCHAR(20) PRIMARY KEY,  
    terrainType VARCHAR(20),  
    PRIMARY KEY (courseName),  
    FOREIGN KEY (courseName) REFERENCES  
CourseNameDistance(courseName) ON UPDATE CASCADE  
);
```

11. CourseStartPoint

```
CREATE TABLE CourseStartPoint (  
    courseName VARCHAR(20),  
    courseDistance INTEGER,  
    stP VARCHAR(20),  
    PRIMARY KEY (courseName, courseDistance),  
    FOREIGN KEY (courseName, courseDistance) REFERENCES  
CourseNameDistance(courseName, courseDistance) ON UPDATE CASCADE  
);
```

12. CourseEndPoint

```
CREATE TABLE CourseEndPoint (  
    courseName VARCHAR(20),  
    courseDistance INTEGER,  
    endP VARCHAR(20),  
    PRIMARY KEY (courseName, courseDistance),  
    FOREIGN KEY (courseName, courseDistance) REFERENCES  
CourseNameDistance(courseName, courseDistance) ON UPDATE CASCADE  
);
```

13. CourseCategoryDistance

```
CREATE TABLE CourseCategoryDistance (  
    courseDistance INTEGER,  
    raceCategoryDistance INTEGER,  
    PRIMARY KEY (courseDistance),  
    FOREIGN KEY (courseDistance) REFERENCES  
CourseNameDistance(courseDistance) ON UPDATE CASCADE  
);
```

14. CourseDistance

```
CREATE TABLE CourseDistance (  
    stP VARCHAR(20),  
    endP VARCHAR(20),  
    courseDistance INTEGER,  
    PRIMARY KEY (stP, endP),  
    FOREIGN KEY (stP) REFERENCES CourseStartPoint(stP) ON UPDATE  
CASCADE,  
    FOREIGN KEY (endP) REFERENCES CourseEndPoint(endP) ON UPDATE  
CASCADE);
```

15. Charity

```
CREATE TABLE Charity (  
    charityName VARCHAR(20) PRIMARY KEY,  
    total INTEGER  
);
```

16. Donation

```
CREATE TABLE Donation (  
    runnerID INTEGER,  
    charityName VARCHAR(20),  
    amount INTEGER,  
    PRIMARY KEY (runnerID, charityName)  
);
```

17. Registration

```
CREATE TABLE Registration (  
    confirmationID INTEGER PRIMARY KEY,  
    registrationDate DATE,  
    runnerID INTEGER,  
    distance INTEGER,  
    marathonName VARCHAR(20),  
    marathonDate DATE  
);
```

18. EliteContact

```
CREATE TABLE EliteContact (  
    runnerID INTEGER PRIMARY KEY,  
    contact VARCHAR(20)  
);
```

19. EliteFirstName

```
CREATE TABLE EliteFirstName (  
    contact VARCHAR(20) PRIMARY KEY,  
    firstName VARCHAR(20),  
    FOREIGN KEY (contact) REFERENCES EliteContact(contact) ON  
UPDATE CASCADE  
);
```

20. EliteLastName

```
CREATE TABLE EliteLastName (  
    contact VARCHAR(20) PRIMARY KEY,  
    lastName VARCHAR(20),  
    FOREIGN KEY (contact) REFERENCES EliteContact(contact) ON  
UPDATE CASCADE  
);
```

21. EliteGender

```
CREATE TABLE EliteGender (  
    contact VARCHAR(20) PRIMARY KEY,  
    gender VARCHAR(20),  
    FOREIGN KEY (contact) REFERENCES EliteContact(contact) ON  
UPDATE CASCADE  
);
```

22. EliteAge

```
CREATE TABLE EliteAge (  
    contact VARCHAR(20) PRIMARY KEY,  
    age INTEGER,  
    FOREIGN KEY (contact) REFERENCES EliteContact(contact) ON  
UPDATE CASCADE  
);
```


23. EliteFastestKm

```
CREATE TABLE EliteFastestKm (  
    contact VARCHAR(20) PRIMARY KEY,  
    fastestKM TIME,  
    FOREIGN KEY (contact) REFERENCES EliteContact(contact) ON  
UPDATE CASCADE  
);
```

24. RookieContact

```
CREATE TABLE RookieContact (  
    runnerID INTEGER PRIMARY KEY,  
    contact VARCHAR(20)  
);
```

25. RookieFirstName

```
CREATE TABLE RookieFirstName (  
    contact VARCHAR(20) PRIMARY KEY,  
    firstName VARCHAR(20),  
    FOREIGN KEY (contact) REFERENCES RookieContact(contact) ON  
UPDATE CASCADE  
);
```

26. RookieLastName

```
CREATE TABLE RookieLastName (  
    contact VARCHAR(20) PRIMARY KEY,  
    lastName VARCHAR(20),  
    FOREIGN KEY (contact) REFERENCES RookieContact(contact) ON  
UPDATE CASCADE  
);
```

27. RookieGender

```
CREATE TABLE RookieGender (  
    contact VARCHAR(20) PRIMARY KEY,  
    gender VARCHAR(20),  
    FOREIGN KEY (contact) REFERENCES RookieContact(contact) ON  
UPDATE CASCADE  
);
```

28. RookieAge

```
CREATE TABLE RookieAge (  
    contact VARCHAR(20) PRIMARY KEY,  
    age INTEGER,  
    FOREIGN KEY (contact) REFERENCES RookieContact(contact) ON  
UPDATE CASCADE  
);
```

29. RookieEstimateTime

```
CREATE TABLE RookieEstimateTime (  
    contact VARCHAR(20) PRIMARY KEY,  
    estimatedTime TIME,  
    FOREIGN KEY (contact) REFERENCES RookieContact(contact) ON  
UPDATE CASCADE  
);
```

30. Result

```
CREATE TABLE Result (  
    runnerID INTEGER PRIMARY KEY,  
    rank INTEGER,  
    finishTime TIME  
);
```

31. Volunteers

```
CREATE TABLE Volunteers (  
    volunteerID INTEGER,  
    marathonName VARCHAR(20),  
    marathonDate DATE,  
    PRIMARY KEY (volunteerID, marathonName, marathonDate)  
    FOREIGN KEY (volunteerID) REFERENCES Volunteer(volunteerID) ON  
UPDATE CASCADE  
    FOREIGN KEY (marathonName, marathonDate) REFERENCES  
MarathonCity(marathonName, marathonDate) ON UPDATE CASCADE  
);
```

32. MarathonEventRaceCategory

```
CREATE TABLE MarathonEventRaceCategory (  
    categoryDistance INTEGER,  
    marathonName VARCHAR(20),  
    marathonDate DATE,  
    PRIMARY KEY (categoryDistance, marathonName, marathonDate)  
    FOREIGN KEY (marathonName, marathonDate) REFERENCES  
MarathonCity(marathonName, marathonDate) ON UPDATE CASCADE  
    FOREIGN KEY (categoryDistance) REFERENCES  
RaceCategory(categoryDistance) ON UPDATE CASCADE  
);
```

33. Sponsors

```
CREATE TABLE Sponsors (  
    marathonName VARCHAR(20),  
    marathonDate DATE,  
    sponsorName VARCHAR(20),  
    PRIMARY KEY (marathonName, marathonDate, sponsorName)  
    FOREIGN KEY (marathonName, marathonDate) REFERENCES  
MarathonCity(marathonName, marathonDate) ON UPDATE CASCADE  
    FOREIGN KEY (sponsorName) REFERENCES Sponsor(marathonName,  
sponsorName) ON UPDATE CASCADE  
);
```

34. Vendor

```
CREATE TABLE Vendor (  
    name VARCHAR(50) PRIMARY KEY,  
    contact VARCHAR(20) NOT NULL,  
    stallNo INTEGER,  
    type VARCHAR(20) NOT NULL  
);
```

35. Booths

```
CREATE TABLE Booths (  
    vendorName VARCHAR(50),  
    marathonDate DATE,  
    marathonName VARCHAR(50),  
    PRIMARY KEY (vendorName, marathonDate, marathonName)  
    FOREIGN KEY (marathonName, marathonDate) REFERENCES  
MarathonCity(marathonName, marathonDate) ON UPDATE CASCADE  
    FOREIGN KEY (vendorName) REFERENCES Vendor(vendorName) ON  
UPDATE CASCADE  
);
```

Populate Tables:

1. VolunteerContact

```
INSERT INTO VolunteerContact (volunteerID, volunteerContact)  
VALUES (1, 'john@gmail.com');  
INSERT INTO VolunteerContact (volunteerID, volunteerContact)  
VALUES (2, 'jane@gmail.com');  
INSERT INTO VolunteerContact (volunteerID, volunteerContact)  
VALUES (3, 'bob@gmail.com');  
INSERT INTO VolunteerContact (volunteerID, volunteerContact)  
VALUES (4, 'susan@gmail.com');  
INSERT INTO VolunteerContact (volunteerID, volunteerContact)  
VALUES (5, 'david@gmail.com');
```

2. VolunteerFirstName

```
INSERT INTO VolunteerFirstName (volunteerContact, firstName)  
VALUES ('john@gmail.com', 'John');  
INSERT INTO VolunteerFirstName (volunteerContact, firstName)  
VALUES ('jane@gmail.com', 'Jane');  
INSERT INTO VolunteerFirstName (volunteerContact, firstName)  
VALUES ('bob@gmail.com', 'Bob');  
INSERT INTO VolunteerFirstName (volunteerContact, firstName)  
VALUES ('susan@gmail.com', 'Susan');  
INSERT INTO VolunteerFirstName (volunteerContact, firstName)  
VALUES ('david@gmail.com', 'David');
```

3. VolunteerLastName

```
INSERT INTO VolunteerLastName (volunteerContact, lastName) VALUES
('john@gmail.com', 'Doe');
INSERT INTO VolunteerLastName (volunteerContact, lastName) VALUES
('jane@gmail.com', 'Smith');
INSERT INTO VolunteerLastName (volunteerContact, lastName) VALUES
('bob@gmail.com', 'Johnson');
INSERT INTO VolunteerLastName (volunteerContact, lastName) VALUES
('susan@gmail.com', 'Williams');
INSERT INTO VolunteerLastName (volunteerContact, lastName) VALUES
('david@gmail.com', 'Jones');
```

4. VolunteerRole

```
INSERT INTO VolunteerRole (volunteerContact, volunteerRole) VALUES
('john@gmail.com', 'CleanUp');
INSERT INTO VolunteerRole (volunteerContact, volunteerRole) VALUES
('jane@gmail.com', 'Pacer');
INSERT INTO VolunteerRole (volunteerContact, volunteerRole) VALUES
('bob@gmail.com', 'Coordinator');
INSERT INTO VolunteerRole (volunteerContact, volunteerRole) VALUES
('susan@gmail.com', 'Registration');
INSERT INTO VolunteerRole (volunteerContact, volunteerRole) VALUES
('david@gmail.com', 'Medic');
```

5. MarathonCity

```
INSERT INTO MarathonCity (marathonName, marathonDate, city) VALUES
('BMO Marathon', '2023-11-15', 'Vancouver'),
('First Half', '2023-12-10', 'Vancouver'),
('Boston Marathon', '2024-01-20', 'Boston'),
('Chicago Marathon', '2024-03-05', 'Chicago'),
('Mumbai Marathon', '2024-04-15', 'Mumbai');
```

6. MarathonWeather

```
INSERT INTO MarathonWeather (city, marathonDate, weatherConditions)
VALUES
('Vancouver', '2023-11-15', 'Sunny'),
('Vancouver', '2023-12-10', 'Cloudy'),
('Boston', '2024-01-20', 'Rainy'),
('Chicago', '2024-03-05', 'Partly Cloudy'),
('Mumbai', '2024-04-15', 'Windy');
```

7. Sponsor

```
INSERT INTO Sponsor (name, contribution) VALUES
('BMO', 100000),
('BlueShore', 75000),
('TCS', 120000),
('Goldman Sachs', 50000),
('Reliance', 80000);
```

8. RaceCategory

```
INSERT INTO RaceCategory (categoryDistance, fee, startTime) VALUES
(5, 25, '2023-11-15 08:00:00'),
(10, 35, '2023-12-10 08:30:00'),
(15, 40, '2024-01-20 09:00:00'),
(21, 55, '2024-03-05 09:30:00'),
(42, 80, '2024-04-15 10:00:00');
```

9. CourseNameDistance

```
INSERT INTO CourseNameDistance (courseDistance, courseName) VALUES
(5, 'BMO Marathon');
INSERT INTO CourseNameDistance (courseDistance, courseName) VALUES
(10, 'First Half');
INSERT INTO CourseNameDistance (courseDistance, courseName) VALUES
(15, 'Boston Marathon');
INSERT INTO CourseNameDistance (courseDistance, courseName) VALUES
(21, 'Chicago Marathon');
INSERT INTO CourseNameDistance (courseDistance, courseName) VALUES
(42, 'Mumbai Marathon');
```

10. CourseTerrain

```
INSERT INTO CourseTerrain (courseName, terrainType) VALUES ('BMO
Marathon', 'City');
INSERT INTO CourseTerrain (courseName, terrainType) VALUES ('First
Half', NULL);
INSERT INTO CourseTerrain (courseName, terrainType) VALUES
('Boston Marathon', 'City');
INSERT INTO CourseTerrain (courseName, terrainType) VALUES
('Chicago Marathon', 'City');
INSERT INTO CourseTerrain (courseName, terrainType) VALUES
('Mumbai Marathon', 'Coastal');
```

11. CourseStartPoint

```
INSERT INTO CourseStartPoint (courseName, courseDistance, stP)
VALUES ('BMO Marathon', 5, 'StartA');
INSERT INTO CourseStartPoint (courseName, courseDistance, stP)
VALUES ('First Half', 10, 'StartB');
INSERT INTO CourseStartPoint (courseName, courseDistance, stP)
VALUES ('Boston Marathon', 15, 'StartC');
INSERT INTO CourseStartPoint (courseName, courseDistance, stP)
VALUES ('Chicago Marathon', 21, 'StartD');
INSERT INTO CourseStartPoint (courseName, courseDistance, stP)
VALUES ('Mumbai Marathon', 42, 'StartE');
```

12. CourseEndPoint

```
INSERT INTO CourseEndPoint (courseName, courseDistance, endP)
VALUES ('BMO Marathon', 5, 'EndA');
INSERT INTO CourseEndPoint (courseName, courseDistance, endP)
VALUES ('First Half', 10, 'EndB');
INSERT INTO CourseEndPoint (courseName, courseDistance, endP)
VALUES ('Boston Marathon', 15, 'EndC');
INSERT INTO CourseEndPoint (courseName, courseDistance, endP)
VALUES ('Chicago Marathon', 21, 'EndD');
INSERT INTO CourseEndPoint (courseName, courseDistance, endP)
VALUES ('Mumbai Marathon', 42, 'EndE');
```

13. CourseCategoryDistance

```
INSERT INTO CourseCategoryDistance (courseDistance,
raceCategoryDistance) VALUES (5, 5);
INSERT INTO CourseCategoryDistance (courseDistance,
raceCategoryDistance) VALUES (10, 10);
INSERT INTO CourseCategoryDistance (courseDistance,
raceCategoryDistance) VALUES (15, 15);
INSERT INTO CourseCategoryDistance (courseDistance,
raceCategoryDistance) VALUES (21, 21);
INSERT INTO CourseCategoryDistance (courseDistance,
raceCategoryDistance) VALUES (42, 42);
```

14. CourseDistance

```
INSERT INTO CourseDistance (stP, endP, courseDistance) VALUES
('StartA', 'EndA', 5);
INSERT INTO CourseDistance (stP, endP, courseDistance) VALUES
('StartB', 'EndB', 10);
INSERT INTO CourseDistance (stP, endP, courseDistance) VALUES
('StartC', 'EndC', 15);
INSERT INTO CourseDistance (stP, endP, courseDistance) VALUES
('StartD', 'EndD', 21);
INSERT INTO CourseDistance (stP, endP, courseDistance) VALUES
('StartE', 'EndE', 42);
```

15. Charity

```
INSERT INTO Charity (charityName, total) VALUES
('Red Cross', 10000),
('Malala Fund', 7500),
('GreenPeace', 12000),
('TataFoundation', 5000),
('RelianceFoundation', 8000);
```

16. Donation

```
INSERT INTO Donation (runnerID, charityName, amount) VALUES
(1, 'Red Cross', 100),
(2, 'Malala Fund', 50),
(3, 'GreenPeace', 150),
(4, 'Red Cross', 75),
(5, 'TataFoundation', 200);
```

17. Registration

```
INSERT INTO Registration (confirmationID, registrationDate,
runnerID, distance, marathonName, marathonDate) VALUES
(1, '2023-10-01', 1, 10, 'BMO Marathon', '2023-11-15'),
(2, '2023-10-02', 2, 15, 'First Half', '2023-12-10'),
(3, '2023-10-03', 3, 21, 'Boston Marathon', '2024-01-20'),
(4, '2023-10-04', 4, 5, 'Chicago Marathon', '2024-03-05'),
(5, '2023-10-05', 5, 42, 'Mumbai Marathon', '2024-04-15');
```


18. EliteContact

```
INSERT INTO EliteContact (runnerID, contact) VALUES
(1, 'johns@gmail.com'),
(2, 'doej@gmail.com'),
(3, 'mickey@gmail.com'),
(4, 'emilyyy@gmail.com'),
(5, 'brownied@gmail.com');
```

19. EliteFirstName

```
INSERT INTO EliteFirstName (contact, firstName) VALUES
('johns@gmail.com', 'John'),
('doej@gmail.com', 'Jane'),
('mickey@gmail.com', 'Michael'),
('emilyyy@gmail.com', 'Emily'),
('brownied@gmail.com', 'David');
```

20. EliteLastName

```
INSERT INTO EliteLastName (contact, lastName) VALUES
('johns@gmail.com', 'Smith'),
('doej@gmail.com', 'Doe'),
('mickey@gmail.com', 'Johnson'),
('emilyyy@gmail.com', 'Wilson'),
('brownied@gmail.com', 'Brown');
```

21. EliteGender

```
INSERT INTO EliteGender (contact, gender) VALUES
('johns@gmail.com', 'Male'),
('doej@gmail.com', 'Female'),
('mickey@gmail.com', 'Male'),
('emilyyy@gmail.com', 'Female'),
('brownied@gmail.com', 'Male');
```

22. EliteAge

```
INSERT INTO EliteAge (contact, age) VALUES
('johns@gmail.com', 30),
('doej@gmail.com', 25),
('mickey@gmail.com', 35),
('emilyyy@gmail.com', 28),
('brownied@gmail.com', 32);
```

23. EliteFastestKm

```
INSERT INTO EliteFastestKm (contact, fastestKM) VALUES
('johns@gmail.com', '04:30:00'),
('doej@gmail.com', '04:15:00'),
('mickey@gmail.com', '03:45:00'),
('emilyyy@gmail.com', '04:00:20'),
('brownied@gmail.com', '04:15:13');
```

24. RookieContact

```
INSERT INTO RookieContact (runnerID, contact) VALUES
(11, 'alexjohn@gmail.com'),
(12, 'sophie@gmail.com'),
(13, 'ethanthegreat@gmail.com'),
(14, 'oliiviia@gmail.com'),
(15, 'liamw@gmail.com');
```

25. RookieFirstName

```
INSERT INTO RookieFirstName (contact, firstName) VALUES
('alexjohn@gmail.com', 'Alex'),
('sophie@gmail.com', 'Sophia'),
('ethanthegreat@gmail.com', 'Ethan'),
('oliiviia@gmail.com', 'Olivia'),
('liamw@gmail.com', 'Liam');
```

26. RookieLastName

```
INSERT INTO RookieLastName (contact, lastName) VALUES
('alexjohn@gmail.com', 'Johnson'),
('sophie@gmail.com', 'Smith'),
('ethanthegreat@gmail.com', 'Brown'),
('oliiviia@gmail.com', 'Doe'),
('liamw@gmail.com', 'Wilson');
```

27. RookieGender

```
INSERT INTO RookieGender (contact, gender) VALUES
('alexjohn@gmail.com', 'Male'),
('sophie@gmail.com', 'Female'),
('ethanthegreat@gmail.com', 'Male'),
('oliiviia@gmail.com', 'Female'),
('liamw@gmail.com', 'Male');
```

28. RookieAge

```
INSERT INTO RookieAge (contact, age) VALUES
('alexjohn@gmail.com', 28),
('sophie@gmail.com', 23),
('ethanthegreat@gmail.com', 29),
('oliiviia@gmail.com', 26),
('liamw@gmail.com', 30);
```

29. RookieEstimateTime

```
INSERT INTO RookieEstimateTime (contact, estimatedTime) VALUES
('alexjohn@gmail.com', '10:15:00'),
('sophie@gmail.com', '07:30:00'),
('ethanthegreat@gmail.com', '06:20:00'),
('oliiviia@gmail.com', '08:45:00'),
('liamw@gmail.com', '05:10:00');
```

30. Result

```
INSERT INTO Result (runnerID, rank, finishTime) VALUES
(1, 1, '02:30:00'),
(2, 2, '02:35:00'),
(3, 3, '02:40:00'),
(4, 4, '02:45:00'),
(5, 5, '02:50:00');
```

31. Volunteers

```
INSERT INTO Volunteers (volunteerID, marathonName, marathonDate)
VALUES
(1, 'BMO Marathon', '2023-11-15'),
(2, 'First Half', '2023-12-10'),
(3, 'Boston Marathon', '2024-01-20'),
(4, 'Chicago Marathon', '2024-03-05'),
(5, 'Mumbai Marathon', '2024-04-15');
```

32. MarathonEventRaceCategory

```
INSERT INTO MarathonEventRaceCategory (categoryDistance,
marathonName, marathonDate) VALUES
(5, 'BMO Marathon', '2023-11-15'),
(10, 'First Half', '2023-12-10'),
(15, 'Boston Marathon', '2024-01-20'),
(20, 'Chicago Marathon', '2024-03-05'),
(25, 'Mumbai Marathon', '2024-04-15');
```

33. Sponsors

```
INSERT INTO Sponsors (marathonName, marathonDate, sponsorName)
VALUES
('BMO Marathon', '2023-11-15', 'BMO'),
('First Half', '2023-12-10', 'BlueShore'),
('Boston Marathon', '2024-01-20', 'TCS'),
('Chicago Marathon', '2024-03-05', 'Goldman Sachs'),
('Mumbai Marathon', '2024-04-15', 'Reliance');
```

34. Vendor

```
INSERT INTO Vendor (name, contact, stallNo, type) VALUES  
( 'Forerunners', 'forerunners@org.com', 101, 'Equipment'),  
( 'BodyEnergyClub', 'becvan@gmail.com', 102, 'Food'),  
( 'XACTNutrition', 'xactnutrition@gmail.com', 103, 'Food'),  
( 'Gatorade', 'gatorade@gmail.com', 104, 'Food'),  
( 'Lululemon', 'lulu@gmail.com', 105, 'Equipment');
```

35. Booths

```
INSERT INTO Booths (vendorName, marathonDate, marathonName) VALUES  
( 'Forerunners', '2023-11-15', 'BMO Marathon'),  
( 'BodyEnergyClub', '2023-12-10', 'First Half'),  
( 'XACTNutrition', '2024-01-20', 'Boston Marathon'),  
( 'Gatorade', '2024-03-05', 'Chicago Marathon'),  
( 'Lululemon', '2024-04-15', 'Mumbai Marathon');
```