University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: 2024-10-15

Group Number: 16

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Nazia Edroos	20010476	d3e6q	edroos.nazia@gmail.com
Prajna Nayak	78725462	j8s1q	prajnapn36@gmail.com
Rachel Wang	71451769	i6a7p	rachelwang0432@gmail.com

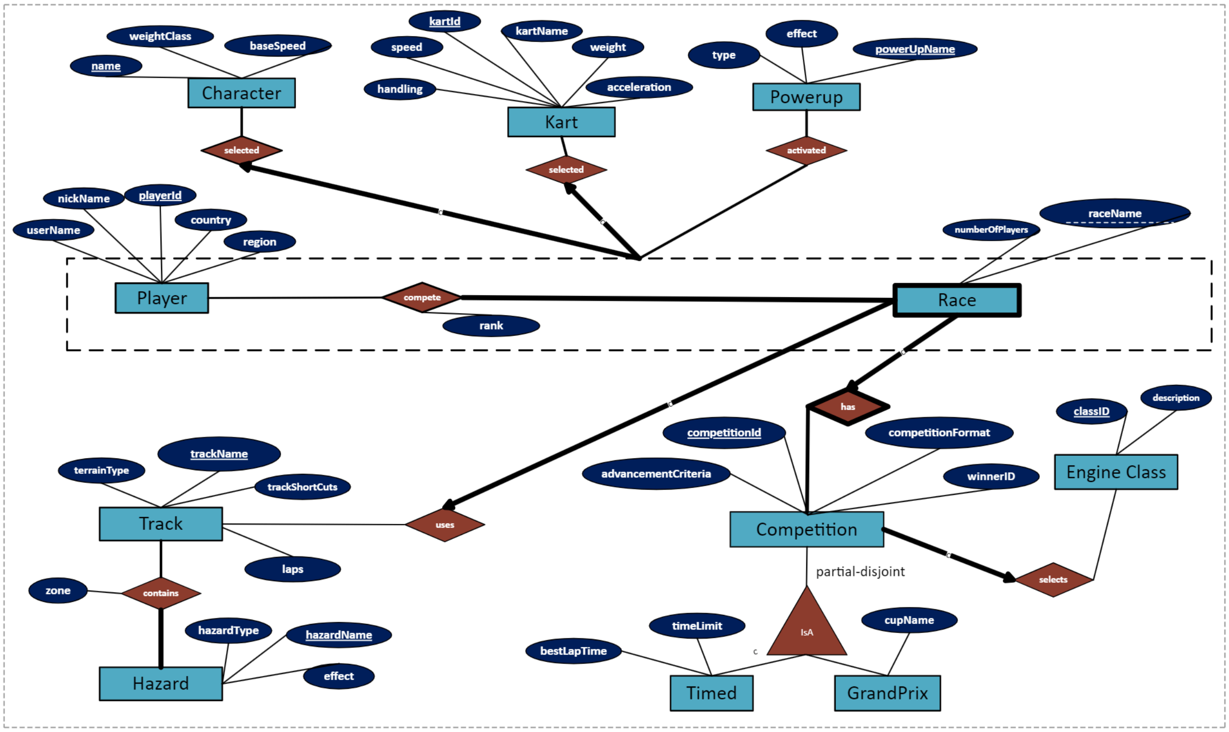
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your email address,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

The Database for Mario Kart competitions is designed to collect, organize, and analyze player performance data, offering valuable insights for both casual players and professional eSports competitors. By evaluating factors such as character and kart combinations, track performance, and power-up usage, the system empowers users to refine their strategies, enhance their competitive gameplay, and make informed decisions to optimize their performance in future races.

ER Diagram



Changes in ER Diagram

Player

- Reworded name attribute to nickName for clarity
- Added userName to add more candidate keys
- Added email attribute to add more candidate keys
- Added region attribute to allow for FDs that aren't identified by a PK or CK

Character

- We added the baseSpeed attribute to allow for FDs that aren't identified by a PK or CK

Kart

- Added KartName to add more candidate keys

Track

- Changed lap attribute to terrainType since Race already has the laps attribute

- Added trackShortCuts (description of known shortcuts on the track) to include meaningful non-key attributes

Powerup

- Added type attribute (one of Offense, Defense, Speed) to include meaningful non-key attributes

Competition

- Added competitionFormat (e.g. elimination bracket, group stage advancement, standard grand prix)to allow for FD that aren't identified by a PK or CK
- Added advancement criteria (e.g. winner advances, top 2 players advance, highest point scorer advances) to allow for FD that aren't identified by a PK or CK

TimedCompetition

- Added bestLapTime to include meaningful non-key attributes

Race

- Added numberOfPlayers so race includes meaningful non-key attributes

Hazard

- Added hazard type(obstacle, enemy) to include meaningful non-key attributes

TrackContainsHazard

- Added a zone attribute (which means where the hazard occurs on the track) for the relationship.

Aggregation Change

- We have changed the aggregation to be with Player and Race instead of Competition. This is because we want to be able to see what players competed in each race. And we will be able to know which competition they participated in because Race is weak to Competition.

Many-to-One Relationship

- We have changed EngineClass to have a many-to-one relationship with competition instead of aggregation. This is because we need to control the competition so that it can only have one engine class.

TA Suggestions

We clarified that our ISA relationship was a partial disjoint. We also changed the line thickness between race and track to clarify that it is not total participation for track (e.g. a track doesn't have to be used by a race).

Schema

- Player(playerID: INT, email: VARCHAR2(100) UNIQUE , userName: VARCHAR2(100) UNIQUE, nickName: VARCHAR2(100), country: CHAR(3), region: VARCHAR2(3))
- EngineClass(classID: VARCHAR2(100), description: VARCHAR2(100) UNIQUE)
- Kart(kartID: INT, kartName: VARCHAR2(100), handling NUMBER(10, 2), speed NUMBER(10, 2), weight NUMBER(10, 2), acceleration NUMBER(10, 2))
- PowerUp(powerUpName: VARCHAR2(100), effect: VARCHAR2(100), type: VARCHAR2(100))
- Character(characterName: VARCHAR2(100), weightClass: VARCHAR2(100), baseSpeed: INT)
- Competition(competitionID: INT, **winnerID**: INT, competitionFormat: VARCHAR2(100), advancementCriteria: VARCHAR2(100))
- TimedCompetition(**competitionId** : INTEGER, timeLimit : INTEGER, bestLapTime: INTEGER)
- GrandPrixCompetition(**competitionId** : INTEGER, cupName : VARCHAR2(100))
- RaceCompetition(raceId: INT, **competitionID**: INT, laps: INT, numberOfPlayers: INT)
- PlayerRace(**playerID**: INT, **raceId**: INT, **competitionID**:INT, rank: INT, **characterName**: VARCHAR2(100), **kartID**: INT)

- CompetitionSelectEngineClass(**competitionId**: INT, **classID**: VARCHAR2(100))
- PlayerRaceActivatePowerUp(**playerId**: INT, **raceId**: INT,**competitionId**: INT, **powerUpName**: VARCHAR2(100))
- Track(trackName: VARCHAR2(500), terrainType: VARCHAR2(100), trackShortCuts: VARCHAR2(100))
- Hazard(hazardName: VARCHAR2(100), effect: VARCHAR2(100), hazardType: VARCHAR2(100))
- TrackContainsHazard(**trackName**: VARCHAR2(100), **hazardName**: VARCHAR2(100), zone: VARCHAR2(100))
- RaceUsesTrack(**raceID**: VARCHAR2(100), **competitionID**: INT, **trackName**: VARCHAR2(100)),

Primary Keys

- Player.playerID
- EngineClass.classID
- Kart.kartName
- PowerUp.powerUpName
- Character.characterName
- Competition.competitionID
- GrandPrixCompetition.competitionID
- TimedCompetition.competitionID
- RaceCompetition.competitionID, RaceCompetition.raceID
- PlayerRace.playerID, PlayerRace.raceName, PlayerCompetition.competitionID
- CompetitionSelectEngineClass.competitionID, CompetitionSelectEngineClass.classID
- PlayerRaceActivatePowerUp.playerID, PlayerCompetitionActivatePowerUp.raceName PlayerRaceActivatePowerUp.competitionID, PlayerCompetitionActivatePowerUp.powerUpName
- Track.trackName
- Hazard.hazardName
- RaceUsesTrack.raceId, RaceUsesTrack.competitionId, RaceUsesTrack.trackName
- TrackContainsHazard.trackName, TrackContainsHazard.hazardName

Candidate Keys

- Player.email
- Player.userName
- Kart.kartID
- EngineClass.description

Foreign Keys

- GrandPrixCompetition.competitionID
- TimedCompetition.competitionID
- Competition.winnerID
- PlayerRace.playerID, PlayerRace.competitionID, PlayerRace.characterName, PlayerRace.kartID
- CompetitionSelectEngineClass.competitionID, CompetitionSelectEngineClass.classID
- PlayerRaceActivatePowerUp.playerID, PlayerRaceActivatePowerUp.competitionID, PlayerRaceActivatePowerUp.powerUpName
- RaceCompetition.competitionID

- RaceUsesTrack.raceId, RaceUsesTrack.competitionId, RaceUsesTrack.trackName
- TrackContainsHazard.trackName, TrackContainsHazard.hazardName

Not Null

- PlayerRace.characterName, PlayerRace.kartID
- RaceUsesTrack.trackName
- CompetitionSelectEngineClass.classID

Unique

- Player.email
- Player.userName
- EngineClass.description
- Kart.kartName
- PlayerRace.competitionId, PlayerRace.characterName
- CompetitionEngineClass.competitionID
- RaceUsesTrack.raceName, RaceUsesTrack.competitionID

Functional Dependencies

Player

- playerID → email, country, region, nickname, userName
- Email → playerID, nickname, country, region, userName
- UserName → playerID, email, nickname, country, region
- Country → Region

EngineClass

- classID → description
- Description → classID

Kart

- kartName → kartID, handling, speed, weight, acceleration
- kartID → kartName, handling, speed, weight, acceleration

Powerup

- powerUpName → effect, type

Character

- characterName → weightClass
- weightClass → baseSpeed

Competition

- competitionID → winnerID, competitionFormat, advancementCriteiria
- competitionFormat → advancementCriteria

TimedCompetition

- competitionID → timeLimit, bestLapTime

GrandPrixCompetition

- competitionID → cupName

RaceCompetition

- raceId, competition → laps, numberOfPlayers

PlayerRace

- playerId, raceID, competitionId → rank, characterName, kartID

CompetitionSelectEngineClass

- *No non-trivial dependencies*

PlayerCompetitionActivatePowerUp

- *No non-trivial dependencies*

Track

- trackName → terrainType, trackShortCuts

Hazard

- hazardName → effect, hazardType

TrackContainsHazard

- trackName, hazardName → zone

RaceUsesTrack

- *No non-trivial dependencies*

Normalization: *3NF Lossless Join Method*

Player Entity* was decomposed down to *PlayerAccount* and *PlayerRegion

Player(playerID: INT, email: VARCHAR2(100) UNIQUE, userName: VARCHAR2(100) UNIQUE, nickName: VARCHAR2(100), country: CHAR(3), region: VARCHAR2(3))

From the FDs:

- playerID → email, country, region, nickname, userName
- Email → playerID, nickname, country, region, userName
- UserName → playerID, email, nickname, country, region
- Country → Region

We can see that Country → Region violates 3NF because Country is not a superkey.

Our minimal covers are:

- playerID → email
- playerID → country
- playerID → nickName
- playerID → userName
- Country → region

PlayerAccount(playerID: INT, email: VARCHAR2(100), userName: VARCHAR2(100), nickName: VARCHAR(100)(100), **country**: CHAR(3))
Location(country CHAR(3), region: VARCHAR2(4))

Competition Entity* was decomposed down to *CompetitionInformation* and *CompetitionFormat

Competition(competitionID: INT, winnerID: INT, competitionFormat: VARCHAR2100), advancementCriteria: VARCHAR2(100))

From the FDs:

- competitionID → winnerID, competitionFormat, advancementCriteira

- competitionFormat → advancementCriteria

We can see that competitionFormat → advancementCriteria violates 3NF because competitionFormat is not a superkey
Our minimal covers are:

- competitionID → winnerID
- competitionID → competitionFormat
- competitionFormat → advancementCriteria

CompetitionInformation(competitionID: INT, winnerID: INT, **competitionFormat**: VARCHAR(100)(100))
CompetitionFormat(competitionFormat: VARCHAR(100)(100), advancementCriteria: VARCHAR(100)(100))

Character Entity **was decomposed down to** ***CharacterInformation*** **and** ***WeightClassInformation***
Character(characterName: VARCHAR2(100), weightClass: VARCHAR2(100), baseSpeed: INT)
From the FDs:

- characterName → weightClass, baseSpeed
- weightClass → baseSpeed

We can see that weightClass→ baseSpeed violates 3NF because weightClass is not a superkey
Our minimal covers:

- characterName → weightClass
- weightClass → baseSpeed

CharacterInformation(characterName: VARCHAR2(100), **weightClass**: VARCHAR2(100))
WeightClassInformation(weightClass: VARCHAR2(100), baseSpeed: INT)

Below are all the table's schemas (updated tables are in purple)

New Schemas

- PlayerAccount(playerID: INT, email: VARCHAR2(100) UNIQUE, userName: VARCHAR2(100) UNIQUE, nickName: VARCHAR2(100), **country**: CHAR(3))
- Location(country CHAR(3), region: VARCHAR2(4))
- EngineClass(classID: VARCHAR2(100), description: VARCHAR2(100) UNIQUE)
- Kart(kartID: INT UNIQUE, kartName: VARCHAR2(100), handling NUMBER(10, 2), speed NUMBER(10, 2), weight NUMBER(10, 2), acceleration NUMBER(10, 2))
- PowerUp(powerUpName: VARCHAR2(100), effect: VARCHA2R(100), type: VARCHAR2(100))
- CharacterInformation(characterName: VARCHAR2(100), **weightClass**: VARCHAR2(100))
- WeightClassInformation(weightClass: VARCHAR2(100), baseSpeed: INT)
- CompetitionInformation(competitionID: INT, **winnerID**: INT, **competitionFormat**: VARCHAR2(100))
- CompetitionFormat(competitionFormat: VARCHAR2(100) advancementCriteria: VARCHAR2(100))
- TimedCompetition(**competitionId** : INTEGER, timeLimit : INTEGER, bestLapTime: INTEGER)
- GrandPrixCompetition(**competitionId** : INTEGER, cupName : VARCHAR2(100))
- RaceCompetition(raceId: INT, **competitionID**: INT, laps: INT, numberOfPlayers: INT)
- PlayerRace(**playerID**: INT, **raceId**: INT, **competitionID**:INT, rank: INT, **characterName**: VARCHAR2(100), **kartID**: INT)
- CompetitionSelectEngineClass(**competitionId**: INT, **classID**: VARCHAR2(100))
- PlayerRaceActivatePowerUp(**playerId**: INT, **raceId**: INT,**competitionId**: INT, **powerUpName**: VARCHAR2(100))
- Track(trackName: VARCHAR2(500), terrainType: VARCHAR2(100), trackShortCuts: VARCHAR2(100))

- Hazard(hazardName: VARCHAR2(100), effect: VARCHAR2(100), hazardType: VARCHAR2(100))
- TrackContainsHazard(**trackName**: VARCHAR2(100), **hazardName**: VARCHAR2(100), zone: VARCHAR2(100))
- RaceUsesTrack(**raceID**: VARCHAR2(100), **competitionID**: INT, **trackName**: VARCHAR2(100)),

Primary Keys

- PlayerAccount.playerID
- Country.country
- EngineClass.classID
- Kart.kartName
- PowerUp.powerUpName
- CharacterInformation.characterName
- WeightClassInformation.weightClass
- CompetitionInformation.competitionID
- CompetitionFormat.competitionFormat
- GrandPrixCompetition.competitionID
- TimedCompetition.competitionID
- RaceCompetition.competitionID, RaceCompetition.raceID
- PlayerRace.playerID, PlayerRace.raceName, PlayerCompetition.competitionID
- CompetitionSelectEngineClass.competitionID, CompetitionSelectEngineClass.classID
- PlayerRaceActivatePowerUp.playerID, PlayerCompetitionActivatePowerUp.raceName PlayerRaceActivatePowerUp.competitionID, PlayerCompetitionActivatePowerUp.powerUpName
- Track.trackName
- Hazard.hazardName
- RaceUsesTrack.raceId, RaceUsesTrack.competitionId, RaceUsesTrack.trackName
- TrackContainsHazard.trackName, TrackContainsHazard.hazardName

Candidate Keys no new CKs

- Player.email
- Player.userName
- Kart.kartID
- EngineClass.description

Foreign Keys

- PlayerAccount.country
- CompetitionFormat.winnerID
- CompetitionInformation.competitionFormat
- CharacterInformation.weightClass
- GrandPrixCompetition.competitionID
- TimedCompetition.competitionID
- Competition.winnerID

- PlayerRace.playerID, PlayerRace.competitionID, PlayerRace.characterName, PlayerRace.kartID
- CompetitionSelectEngineClass.competitionID, CompetitionSelectEngineClass.classID
- PlayerRaceActivatePowerUp.playerID, PlayerRaceActivatePowerUp.competitionID, PlayerRaceActivatePowerUp.powerUpName
- RaceCompetition.competitionID
- RaceUsesTrack.raceId, RaceUsesTrack.competitionId, RaceUsesTrack.trackName
- TrackContainsHazard.trackName, TrackContainsHazard.hazardName

Not Null no new not nulls

- PlayerRace.characterName, PlayerRace.kartID
- RaceUsesTrack.trackName
- CompetitionSelectEngineClass.classID

Unique no new unique

- Player.email
- Player.userName
- EngineClass.description
- Kart.kartName
- PlayerRace.raceName, PlayerRace.characterName
- CompetitionEngineClass.competitionID
- RaceUsesTrack.raceName, RaceUsesTrack.competitionID

SQL DDL Statements

```
CREATE TABLE Country (
  Country CHAR(3) PRIMARY KEY,
  Region VARCHAR2(4)
);


CREATE TABLE PlayerAccount (
  PlayerID INTEGER PRIMARY KEY,
  email VARCHAR2(100) UNIQUE,
  userName VARCHAR2(100) UNIQUE,
  nickName VARCHAR2(100),
  Country CHAR(3),
  FOREIGN KEY (Country) REFERENCES Country(Country)
  ON DELETE CASCADE
);

CREATE TABLE Kart (
  kartID INT PRIMARY KEY,
  kartName VARCHAR2(100) UNIQUE,
  handling NUMBER(10, 2),
  speed NUMBER(10, 2),
  weight NUMBER(10, 2),
  acceleration NUMBER(10, 2)
 );

  CREATE TABLE PowerUp (
  powerUpName VARCHAR2(100)PRIMARY KEY,
  effect VARCHAR2(100),
  type VARCHAR2(100)
 );

 CREATE TABLE WeightClassInformation(
    weightClass VARCHAR2(100) PRIMARY KEY,
    baseSpeed INTEGER
   );


  CREATE TABLE CharacterInformation (
    characterName VARCHAR2(100) PRIMARY KEY,
    weightClass VARCHAR2(100),
    FOREIGN KEY (weightClass) REFERENCES WeightClassInformation (weightClass)
 );


 CREATE TABLE EngineClass (
  classID VARCHAR2(100) PRIMARY KEY,
  description VARCHAR2(100)
);

   CREATE TABLE CompetitionFormat (
  competitionFormat VARCHAR2(100)PRIMARY KEY,
  advancementCriteria VARCHAR2(100)
);
  CREATE TABLE CompetitionInformation (
  competitionID INTEGER PRIMARY KEY,
  winnerID INTEGER,
  competitionFormat VARCHAR2(100),
  FOREIGN KEY (competitionFormat) REFERENCES CompetitionFormat(competitionFormat)
   ON DELETE CASCADE,
  FOREIGN KEY (winnerID) REFERENCES PlayerAccount(PlayerID)
  ON DELETE CASCADE
 );

 CREATE TABLE TimedCompetition (
    competitionId INTEGER PRIMARY KEY,
    timeLimit INTEGER,
    bestLapTime INTEGER,
    FOREIGN KEY (competitionId) REFERENCES CompetitionInformation(competitionID)
            ON DELETE CASCADE
);
CREATE TABLE GrandPrixCompetition (
    competitionId INTEGER PRIMARY KEY,
    cupName VARCHAR2(100),
    FOREIGN KEY (competitionId) REFERENCES CompetitionInformation(competitionID)
            ON DELETE CASCADE
);

CREATE TABLE RaceCompetition(
  raceName VARCHAR2(100),
  competitionID INTEGER,
  laps INTEGER,
  numberOfPlayers INTEGER,
  PRIMARY KEY (raceName, competitionID),
  FOREIGN KEY (competitionID) REFERENCES CompetitionInformation (competitionID) ON
DELETE CASCADE
);

 CREATE TABLE PlayerRace (
   playerID INTEGER,
   raceName VARCHAR2(100),
   competitionID INTEGER,
   rank INTEGER,
   characterName VARCHAR2(100) NOT NULL,
   kartID INTEGER NOT NULL,
   PRIMARY KEY (playerID, raceName, competitionID),
   FOREIGN KEY (playerID) REFERENCES PlayerAccount(playerID) ON DELETE CASCADE,
   FOREIGN KEY (competitionID) REFERENCES CompetitionInformation(competitionID) ON
DELETE CASCADE,
    FOREIGN KEY (characterName) REFERENCES Character(characterName) ON DELETE
CASCADE,
    FOREIGN KEY (kartID) REFERENCES KART (kartID) ON DELETE CASCADE,
    UNIQUE (raceName, characterName)
);

CREATE TABLE PlayerRaceActivatePowerUp(
  playerID INTEGER,
  raceName VARCHAR2(100),
  competitionID INTEGER,
  powerUpName VARCHAR2(100),
  PRIMARY KEY (playerID, raceName, competitionID, powerUpName),
  FOREIGN KEY (playerID, raceName, competitionID) REFERENCES PlayerRace(playerID, raceName,
competitionID) ON DELETE CASCADE,
  FOREIGN KEY (powerUpName) REFERENCES PowerUp (powerUpName)
);

CREATE TABLE CompetitionEngineClass (
  competitionID INTEGER,
  classID VARCHAR2(100) NOT NULL,
  UNIQUE(competitionID),
  PRIMARY KEY (competitionID, classID),
  FOREIGN KEY (classID) REFERENCES EngineClass(classID) ON DELETE CASCADE,
  FOREIGN KEY (competitionID) REFERENCES CompetitionInformation ON DELETE CASCADE
);

CREATE TABLE Track(
  trackName VARCHAR2(100) PRIMARY KEY,
  terrainType VARCHAR2(100),
  trackShortCuts VARCHAR2(500)
);

CREATE TABLE Hazard(
  hazardName VARCHAR2(100) PRIMARY KEY,
  hazardType VARCHAR2(100),
  effect VARCHAR2(100)
);


CREATE TABLE TrackContainsHazard(
  trackName VARCHAR2(100),
  hazardName VARCHAR2(100),
  zone VARCHAR2(100),
  PRIMARY KEY (trackName, hazardName),
  FOREIGN KEY (trackName) REFERENCES Track (trackName)ON DELETE CASCADE,
  FOREIGN KEY (hazardName) REFERENCES Hazard (hazardName)ON DELETE CASCADE
);

CREATE TABLE RaceUsesTrack(
  raceName VARCHAR(200),
  competitionID INTEGER,
  trackName VARCHAR2(100),
  UNIQUE (raceName, competitionID),
  PRIMARY KEY (raceName, competitionID, trackName),
  FOREIGN KEY (raceName, competitionID) REFERENCES RaceCompetition(raceName,
competitionID) ON DELETE CASCADE,
  FOREIGN KEY (trackName) REFERENCES Track (trackName) ON DELETE CASCADE
);
```

Oracle does not support update cascade. Our foreign keys are referencing our primary keys (it's a bad database design to change primary keys) **Note that we cannot enforce the participation constraints in the many-to-many relationship between player and competition and that we will enforce it at a later point (with assertions).**

INSERT Statements

-- Insert WeightClassInformation

```
INSERT INTO WeightClassInformation(weightClass, baseSpeed) VALUES ('Medium Heavy', 10);
INSERT INTO WeightClassInformation(weightClass, baseSpeed) VALUES ('Very Heavy', 20);
INSERT INTO WeightClassInformation(weightClass, baseSpeed) VALUES ('Very Light', 3);
INSERT INTO WeightClassInformation(weightClass, baseSpeed) VALUES ('Heavy', 15);
INSERT INTO WeightClassInformation(weightClass, baseSpeed) VALUES ('Light', 5);
```

-- Insert CharacterInformation

```
INSERT INTO CharacterInformation(characterName, weightClass) VALUES ('Mario', 'Medium Heavy');
INSERT INTO CharacterInformation(characterName, weightClass) VALUES ('Baby Mario', 'Very
Light');
INSERT INTO CharacterInformation(characterName, weightClass) VALUES ('Bowser', 'Very Heavy');
INSERT INTO CharacterInformation(characterName, weightClass) VALUES ('Donkey Kong', 'Heavy');
INSERT INTO CharacterInformation(characterName, weightClass) VALUES ('Toad', 'Light');
```

-- Insert Kart

```
INSERT INTO KART(kartId, kartName, handling, speed, weight, acceleration) VALUES (1, 'Standard
Kart', 0, 0, 0, 0);
INSERT INTO KART(kartId, kartName, handling, speed, weight, acceleration) VALUES (2, 'Pipe
Frame', 0.5, 0, -0.25, 0.25);
INSERT INTO KART(kartId, kartName, handling, speed, weight, acceleration) VALUES (3, 'Mach 8', 0,
0.5, 0.25, -0.25);
```

```
INSERT INTO KART(kartId, kartName, handling, speed, weight, acceleration) VALUES (4, 'Steel Driver', -0.5, 0, 0.5, -0.5);
INSERT INTO KART(kartId, kartName, handling, speed, weight, acceleration) VALUES (5, 'Cat Cruiser', 0, 0, 0, 0);

-- Insert Country
INSERT INTO Country(Country, Region) VALUES ('CAD', 'NA');
INSERT INTO Country(Country, Region) VALUES ('CHN', 'AS');
INSERT INTO Country(Country, Region) VALUES ('LON', 'EU');
INSERT INTO Country(Country, Region) VALUES ('EGY', 'AF');
INSERT INTO Country (Country, Region) VALUES ('NZL', 'OC'); [-- New Zealand]

-- Insert player accounts
INSERT INTO PlayerAccount(PlayerID, email, userName, nickName, Country)
VALUES (1, 'rachel@gmail.com', 'rach01', 'Rachel', 'CAD');

INSERT INTO PlayerAccount(PlayerID, email, userName, nickName, Country)
VALUES (2, 'nazia@gmail.com', 'nazia02', 'Nazia', 'CHN');

INSERT INTO PlayerAccount(PlayerID, email, userName, nickName, Country)
VALUES (3, 'prajna@gmail.com', 'prajna03', 'Prajna', 'LON');

INSERT INTO PlayerAccount(PlayerID, email, userName, nickName, Country)
VALUES (4, 'victoria@gmail.com', 'vic04', 'Victoria', 'EGY');

INSERT INTO PlayerAccount(PlayerID, email, userName, nickName, Country)
VALUES (5, 'val@gmail.com', 'val05', 'Val', 'LON');


 -- INSERT CompetitionFormat
 INSERT INTO CompetitionFormat(competitionFormat, advancementCriteria) VALUES('elimination bracket', 'winner advances');
 INSERT INTO CompetitionFormat(competitionFormat, advancementCriteria) VALUES('group stage advancement', 'top 2 advances');
 INSERT INTO CompetitionFormat(competitionFormat, advancementCriteria) VALUES('standard grandprix', 'highest point scorer advances');
 INSERT INTO CompetitionFormat(competitionFormat, advancementCriteria) VALUES('Round Robin','players with the best win-loss ratio advances to finals');
 INSERT INTO CompetitionFormat(competitionFormat, advancementCriteria) VALUES('Teams', 'teams w/ the best score advance');
 -- INSERT CompetitionInformation
INSERT INTO CompetitionInformation(competitionID, winnerID, competitionFormat) VALUES(1, 1, 'elimination bracket');
INSERT INTO CompetitionInformation(competitionID, winnerID, competitionFormat) VALUES(2, 1, 'group stage advancement');
INSERT INTO CompetitionInformation(competitionID, winnerID, competitionFormat) VALUES(3, 2, 'standard grandprix');
INSERT INTO CompetitionInformation(competitionID, winnerID, competitionFormat) VALUES(4, 3, 'Round Robin');
INSERT INTO CompetitionInformation(competitionID, winnerID, competitionFormat) VALUES(5, 3, 'Teams');
INSERT INTO CompetitionInformation(competitionID, winnerID, competitionFormat) VALUES(6, 4, 'elimination bracket');
INSERT INTO CompetitionInformation(competitionID, winnerID, competitionFormat) VALUES(7, 4, 'group stage advancement');
INSERT INTO CompetitionInformation(competitionID, winnerID, competitionFormat) VALUES(8, 5, 'standard grandprix');
INSERT INTO CompetitionInformation(competitionID, winnerID, competitionFormat) VALUES(9, 5, 'Round Robin');
INSERT INTO CompetitionInformation(competitionID, winnerID, competitionFormat) VALUES(10, 5, 'Teams');

-- Insert TimedCompetition
INSERT INTO TimedCompetition (competitionId, timeLimit, bestLapTime) VALUES (1, 300, 120);
INSERT INTO TimedCompetition (competitionId, timeLimit, bestLapTime) VALUES (2, 300, 120);
INSERT INTO TimedCompetition (competitionId, timeLimit, bestLapTime) VALUES (3, 300, 120);
INSERT INTO TimedCompetition (competitionId, timeLimit, bestLapTime) VALUES (4, 300, 120);
INSERT INTO TimedCompetition (competitionId, timeLimit, bestLapTime) VALUES (5, 300, 120);

-- Insert GrandPrixCompetition
INSERT INTO GrandPrixCompetition (competitionId, cupName) VALUES (6, 'Mushroom Cup');
INSERT INTO GrandPrixCompetition (competitionId, cupName) VALUES (7, 'Flower Cup');
INSERT INTO GrandPrixCompetition (competitionId, cupName) VALUES (8, 'Star Cup');
INSERT INTO GrandPrixCompetition (competitionId, cupName) VALUES (9, 'Special Cup');
INSERT INTO GrandPrixCompetition (competitionId, cupName) VALUES (10, 'Lightening Cup');

-- Insert RaceCompetition
INSERT INTO RaceCompetition(raceName, competitionID, laps, numberOfPlayers) VALUES ('RACE 1', 1, 3, 8);
INSERT INTO RaceCompetition(raceName, competitionID, laps, numberOfPlayers) VALUES ('RACE 1', 2, 4, 10);
INSERT INTO RaceCompetition(raceName, competitionID, laps, numberOfPlayers) VALUES ('RACE 2', 1, 3, 8);
INSERT INTO RaceCompetition(raceName, competitionID, laps, numberOfPlayers) VALUES ('RACE 2', 2, 4, 10);
INSERT INTO RaceCompetition(raceName, competitionID, laps, numberOfPlayers) VALUES ('RACE 3', 3, 3, 8);


 -- Insert PlayerRace
INSERT INTO PlayerRace(playerId, raceName, competitionID, rank, characterName, kartID) VALUES (1, 'RACE 1', 1, 1, 'Mario', 1);
INSERT INTO PlayerRace(playerId, raceName, competitionID, rank, characterName, kartID) VALUES (1, 'RACE 1', 2, 1, 'Toad', 1);
INSERT INTO PlayerRace(playerId, raceName, competitionID, rank, characterName, kartID) VALUES (2, 'RACE 2', 1, 1, 'Bowser', 2);
INSERT INTO PlayerRace(playerId, raceName, competitionID, rank, characterName, kartID) VALUES (3, 'RACE 2', 2, 3, 'Baby Mario', 3);
INSERT INTO PlayerRace(playerId, raceName, competitionID, rank, characterName, kartID) VALUES (1, 'RACE 3', 3, 1, 'Mario', 1);
-- Insert EngineClass
INSERT INTO EngineClass(classID, description) VALUES ('50cc', 'Easy');
INSERT INTO EngineClass(classID, description) VALUES ('100cc', 'Medium');
INSERT INTO EngineClass(classID, description) VALUES ('150cc', 'Hard');
INSERT INTO EngineClass(classID, description) VALUES ('200cc', 'Expert');
INSERT INTO EngineClass(classID, description) VALUES ('Mirror Mode', 'tracks horizontally flipped');

 -- Insert Powerup
INSERT INTO PowerUp(powerUpName, effect,type) VALUES('Banana', 'make players slip', 'OFFENCE/DEFENSE');
INSERT INTO PowerUp(powerUpName, effect,type) VALUES('Lightening', 'all players except user lose items + spin', 'OFFENCE');
INSERT INTO PowerUp(powerUpName, effect,type) VALUES('Green Shell', 'throw at opponent', 'OFFENCE');
INSERT INTO PowerUp(powerUpName, effect,type) VALUES('Mushroom', 'accelerate user', 'SPEED');
INSERT INTO PowerUp(powerUpName, effect,type) VALUES('Starman', 'turn user invincible + increase speed by 1/3', 'SPEED/DEFENSE');

-- InsertPlayerRaceActivatePowerUp
INSERT INTO PlayerRaceActivatePowerUp(playerId, raceName, competitionID, powerUpName) VALUES (1, 'RACE 1', 1, 'Banana');
INSERT INTO PlayerRaceActivatePowerUp(playerId, raceName, competitionID, powerUpName) VALUES (1, 'RACE 1', 2, 'Mushroom');
INSERT INTO PlayerRaceActivatePowerUp(playerId, raceName, competitionID, powerUpName) VALUES (2, 'RACE 2', 1, 'Banana');
INSERT INTO PlayerRaceActivatePowerUp(playerId, raceName, competitionID, powerUpName) VALUES (3, 'RACE 2', 2, 'Banana');
INSERT INTO PlayerRaceActivatePowerUp(playerId, raceName, competitionID, powerUpName) VALUES (1, 'RACE 3', 3, 'Banana');

-- Insert CompetitionSelectEngineClass
INSERT INTO CompetitionEngineClass(competitionID, classID) VALUES (1, '100cc');
INSERT INTO CompetitionEngineClass(competitionID, classID) VALUES (2, '100cc');
INSERT INTO CompetitionEngineClass(competitionID, classID) VALUES (3, '150cc');
INSERT INTO CompetitionEngineClass(competitionID, classID) VALUES (4, '200cc');
INSERT INTO CompetitionEngineClass(competitionID, classID) VALUES (5, '50cc');

-- Insert Hazard
INSERT INTO Hazard(hazardName, hazardType, effect) VALUES ('Goomba', 'Enemy', 'spin players');
INSERT INTO Hazard(hazardName, hazardType, effect) VALUES ('Lava Boulder', 'Volcanic', 'spin players and lose time');
INSERT INTO Hazard(hazardName, hazardType, effect) VALUES ('Car', 'Traffic', 'send you into the air');
INSERT INTO Hazard(hazardName, hazardType, effect) VALUES ('Piranha Plants', 'Plant', 'knocked over');
INSERT INTO Hazard(hazardName, hazardType, effect) VALUES ('Thwomps', 'Obstacle', 'block or squish user');

-- Insert Track
INSERT INTO Track(trackName, terrainType, trackShortCuts) VALUES ('Rainbow Road', 'Rainbow', 'In the space station with the rotating floors, use a Mushroom on the inside of the area to cut through most of this section.');
INSERT INTO Track(trackName, terrainType, trackShortCuts) VALUES ('Mario Circuit', 'Standard', 'After the first turn, use a Mushroom to cut through the grass with the two pipes. Do the same for the next turn as well.');
INSERT INTO Track(trackName, terrainType, trackShortCuts) VALUES ('Bowser`s Castle', 'Lava', 'At the end of the track are several tight turns. Use a Mushroom to get past any of the three rough areas.');
INSERT INTO Track(trackName, terrainType, trackShortCuts) VALUES ('Mount Wario', 'Ice', 'Before you enter the icy cave, jump off the small ramp on your right just before the tight turn.');
INSERT INTO Track(trackName, terrainType, trackShortCuts) VALUES ('Toad`s Turnpike', 'City', 'Keep an eye out for a truck with a Gliding ramp just before going under the highway. Glide high to pass right over the entire highway');

-- Insert TrackContainsHazard
INSERT INTO TrackContainsHazard (trackName, hazardName, zone) VALUES ('Mario Circuit', 'Goomba', 'Mid');
INSERT INTO TrackContainsHazard (trackName, hazardName, zone) VALUES ('Mario Circuit', 'Piranha Plants', 'Start and Mid');
INSERT INTO TrackContainsHazard (trackName, hazardName, zone) VALUES ('Bowser`s Castle', 'Lava Boulder', 'End');
INSERT INTO TrackContainsHazard (trackName, hazardName, zone) VALUES ('Rainbow Road', 'Thwomps', 'End');
INSERT INTO TrackContainsHazard (trackName, hazardName, zone) VALUES ('Toad`s Turnpike', 'Car', 'Mid');

-- PlayerRaceUsesTrack
INSERT INTO RaceUsesTrack(raceName, competitionID, trackName) VALUES ('RACE 1', 1,'Mario Circuit');
INSERT INTO RaceUsesTrack(raceName, competitionID, trackName) VALUES ('RACE 1', 2, 'Bowser`s Castle');
INSERT INTO RaceUsesTrack(raceName, competitionID, trackName) VALUES ('RACE 2', 1, 'Rainbow Road');
INSERT INTO RaceUsesTrack(raceName, competitionID, trackName) VALUES ('RACE 2', 2, 'Rainbow Road');
INSERT INTO RaceUsesTrack(raceName, competitionID, trackName) VALUES ('RACE 3', 3, 'Toad`s Turnpike');
```