```
CREATE TABLE CricketPerson
        (ID SERIAL PRIMARY KEY,
        Name VARCHAR(50) NOT NULL,
        DoB DATE NOT NULL);

CREATE TABLE Coach
        (ID INTEGER REFERENCES CricketPerson(ID),
        PRIMARY KEY(ID));

CREATE TABLE Player
        (ID SERIAL REFERENCES CricketPerson(ID),
        BattingStyle VARCHAR(30),
        BowlingStyle VARCHAR(30),
        PRIMARY KEY (ID));

CREATE TABLE Umpire
        (ID INTEGER REFERENCES CricketPerson(ID),
        NumODIs INTEGER NOT NULL,
        NumTests INTEGER NOT NULL,
        NumT20s INTEGER NOT NULL,
        PRIMARY KEY (ID));

CREATE TABLE ODIPlayer
        (ID INTEGER REFERENCES Player(ID),
        NumMatches INTEGER NOT NULL,
        NumCatches INTEGER NOT NULL,
        PRIMARY KEY (ID),
        CHECK (10* NumCatches <= NumMatches));

CREATE TABLE TestPlayer
        (ID INTEGER REFERENCES Player(ID),
        NumMatches INTEGER NOT NULL,
        NumCatches INTEGER NOT NULL,
        PRIMARY KEY (ID),
        CHECK (20* NumCatches <= NumMatches));

CREATE TABLE T20Player
        (ID INTEGER REFERENCES Player(ID),
        NumMatches INTEGER NOT NULL,
        NumCatches INTEGER NOT NULL,
        PRIMARY KEY (ID),
        CHECK (10* NumCatches <= NumMatches));
```

```
CREATE TABLE ODIBatsman
        (ID INTEGER REFERENCES ODIPlayer(ID),
        NumInnings INTEGER NOT NULL,
        NumNotOuts INTEGER NOT NULL,
        Runs INTEGER NOT NULL,
        HighScore INTEGER NOT NULL,
        Average REAL NOT NULL,
        StrikeRate REAL NOT NULL,
        Fifties INTEGER NOT NULL,
        Hundreds INTEGER NOT NULL,
        Fours INTEGER NOT NULL,
        Sixes INTEGER NOT NULL,
        PRIMARY KEY(ID),
        CHECK (Runs >= 4 * Fours + 6 * Sixes),
        CHECK (NumNotOuts <= NumInnings)
        );

        CHECK (NumInnings <= (SELECT NumMatches FROM ODIPlayer AS E WHERE ID =
    E.ID)),


CREATE TABLE TestBatsman
        (ID INTEGER REFERENCES TestPlayer(ID),
        NumInnings INTEGER NOT NULL,
        NumNotOuts INTEGER NOT NULL,
        Runs INTEGER NOT NULL,
        HighScore INTEGER NOT NULL,
        Average REAL NOT NULL,
        StrikeRate REAL NOT NULL,
        Fifties INTEGER NOT NULL,
        Hundreds INTEGER NOT NULL,
        Fours INTEGER NOT NULL,
        Sixes INTEGER NOT NULL,
        PRIMARY KEY(ID),
        CHECK(Runs >= 4 * Fours + 6 * Sixes),
        CHECK (NumNotOuts <= NumInnings)
        );

CHECK (NumInnings <= (SELECT NumMatches FROM TestPlayer AS E WHERE ID = E.ID)),
```

```
CREATE TABLE T20Batsman
        (ID INTEGER REFERENCES T20Player(ID),
        NumInnings INTEGER NOT NULL,
        NumNotOuts INTEGER NOT NULL,
        Runs INTEGER NOT NULL,
        HighScore INTEGER NOT NULL,
        Average REAL NOT NULL,
        StrikeRate REAL NOT NULL,
        Fifties INTEGER NOT NULL,
        Hundreds INTEGER NOT NULL,
        Fours INTEGER NOT NULL,
        Sixes INTEGER NOT NULL,
        PRIMARY KEY(ID),
        CHECK(Runs >= 4 * Fours + 6 * Sixes),
        CHECK (NumNotOuts <= NumInnings)
        );

    CHECK (NumInnings <= (SELECT NumMatches FROM T20Player AS E WHERE ID = E.ID)),


CREATE TABLE ODIBowler
        (ID INTEGER REFERENCES ODIPlayer(ID),
        NumInnings  INTEGER NOT NULL,
        NumBalls INTEGER NOT NULL,
        Runs INTEGER NOT NULL,
        Wickets INTEGER NOT NULL,
        BestPerformanceWickets INTEGER NOT NULL,
        BestPerformanceRuns INTEGER NOT NULL,
        Average REAL NOT NULL,
        Economy REAL NOT NULL,
        StrikeRate REAL,
        Num5W INTEGER NOT NULL,
        PRIMARY KEY (ID),
        CHECK (BestPerformanceWickets < 11)
        );
        ,
        CHECK (Wickets <=10 * (SELECT NumMatches FROM ODIPlayer AS E WHERE ID =
        E.ID))

CREATE TABLE TestBowler
        (ID INTEGER REFERENCES TestPlayer(ID),
        NumInnings  INTEGER NOT NULL,
        NumBalls INTEGER NOT NULL,
```

```
        Runs INTEGER NOT NULL,
        Wickets INTEGER NOT NULL,
        BestInningsPerformanceWickets INTEGER NOT NULL,
        BestInningsPerformanceRuns INTEGER NOT NULL,
        BestMatchPerformanceWickets INTEGER NOT NULL,
        BestMatchPerformanceRuns INTEGER NOT NULL,
        Average REAL NOT NULL,
        Economy REAL NOT NULL,
        StrikeRate REAL,
        Num5W INTEGER NOT NULL,
        PRIMARY KEY (ID),
        CHECK (BestInningsPerformanceWickets < 11),
        CHECK (BestMatchPerformanceWickets < 21)
        );

        ,
        CHECK (Wickets <=20 * (SELECT NumMatches FROM TestPlayer AS E WHERE ID =
        E.ID))

CREATE TABLE T20Bowler
        (ID INTEGER REFERENCES T20Player(ID),
        NumInnings  INTEGER NOT NULL,
        NumBalls INTEGER NOT NULL,
        Runs INTEGER NOT NULL,
        Wickets INTEGER NOT NULL,
        BestPerformanceWickets INTEGER NOT NULL,
        BestPerformanceRuns INTEGER NOT NULL,
        Average REAL NOT NULL,
        Economy REAL NOT NULL,
        StrikeRate REAL,
        Num5W INTEGER NOT NULL,
        PRIMARY KEY (ID),
        CHECK (BestPerformanceWickets < 11)
        );

        ,
        CHECK (Wickets <=10 * (SELECT NumMatches FROM T20Player AS E WHERE ID =
        E.ID))
        ODIWicketKeeper(ID, Stumpings)

CREATE TABLE ODIWicketKeeper
        (ID INTEGER REFERENCES ODIPlayer(ID),
        Stumpings INTEGER NOT NULL,
```

```
        PRIMARY KEY (ID));

        ,
        CHECK (10 * Stumpings <= (SELECT NumMatches FROM ODIPlayer AS E WHERE
    E.ID = ID))

CREATE TABLE TestWicketKeeper
        (ID INTEGER REFERENCES TestPlayer(ID),
        Stumpings INTEGER NOT NULL,
        PRIMARY KEY (ID));

        ,
        CHECK (20 * Stumpings <= (SELECT NumMatches FROM TestPlayer AS E WHERE
    E.ID = ID))

CREATE TABLE T20WicketKeeper
        (ID INTEGER REFERENCES T20Player(ID),
        Stumpings INTEGER NOT NULL,
        PRIMARY KEY (ID));

        ,
        CHECK (10 * Stumpings <= (SELECT NumMatches FROM T20Player AS E WHERE
    E.ID = ID))
```

- ➔ IsInTeam(Country, PlayerID)
  - Country references CountryName from Team
  - PlayerID references ID from Player

```
CREATE TABLE IsInTeam
        (Country VARCHAR(20) REFERENCES Team(CountryName),
        ID INTEGER REFERENCES Player(ID),
        PRIMARY KEY(Country, ID));
```

- ➔ ODIBatsmanRanking(PlayerID, Rank, Points)
  - PlayerID references ID from ODIBatsman

```
CREATE TABLE ODIBatsmanRanking(
        ID INTEGER REFERENCES ODIBatsman(ID),
        Rank INTEGER,
        Points INTEGER,
        PRIMARY KEY(ID));
```

- ➔ ODIBowlerRanking(PlayerID, Rank, Points)
  - PlayerID references ID from ODIPlayer

```
CREATE TABLE ODIBowlerRanking(
        ID INTEGER REFERENCES ODIBowler(ID),
        Rank INTEGER,
        Points INTEGER,
        PRIMARY KEY(ID));
```

➔ ODIAllRounderRanking(PlayerID, Rank, Points)

- PlayerID references ID from ODIPlayer

```
CREATE TABLE ODIAllRounderRanking(
        ID INTEGER,
        Rank INTEGER,
        Points INTEGER,
        PRIMARY KEY(ID));
,
        CHECK (EXISTS (SELECT * FROM ODIBatsman AS E1 WHERE E1.ID = ID)
                AND EXISTS (SELECT * FROM ODIBowler AS E1 WHERE E1.ID = ID)));
```

➔ TestBatsmanRanking(PlayerID, Rank, Points)

- PlayerID references ID from TestPlayer

```
CREATE TABLE TestBatsmanRanking(
        ID INTEGER REFERENCES TestBatsman(ID),
        Rank INTEGER,
        Points INTEGER,
        PRIMARY KEY(ID));
```

➔ TestBowlerRanking(PlayerID, Rank, Points)

- PlayerID references ID from TestPlayer

```
CREATE TABLE TestBowlerRanking(
        ID INTEGER REFERENCES TestBowler(ID),
        Rank INTEGER,
        Points INTEGER,
        PRIMARY KEY(ID));
```

➔ TestAllRounderRanking(PlayerID, Rank, Points)

- PlayerID references ID from TestPlayer

```
CREATE TABLE TestAllRounderRanking(
        ID INTEGER,
```

```
        Rank INTEGER,
        Points INTEGER,
        PRIMARY KEY(ID));
,
        CHECK (EXISTS (SELECT * FROM TestBatsman AS E1 WHERE E1.ID = ID)
                AND EXISTS (SELECT * FROM TestBowler AS E1 WHERE E1.ID = ID)));
```

➔ T20BatsmanRanking(PlayerID, Rank, Points)

- PlayerID references ID from T20Player

```
CREATE TABLE T20BatsmanRanking(
        ID INTEGER REFERENCES T20Batsman(ID),
        Rank INTEGER,
        Points INTEGER,
        PRIMARY KEY(ID));
```

➔ T20BowlerRanking(PlayerID, Rank, Points)

- PlayerID references ID from T20Player

```
CREATE TABLE T20BowlerRanking(
        ID INTEGER REFERENCES T20Bowler(ID),
        Rank INTEGER,
        Points INTEGER,
        PRIMARY KEY(ID));
```

➔ T20AllRounderRanking(PlayerID, Rank, Points)

- PlayerID references ID from T20Player

```
CREATE TABLE T20AllRounderRanking(
        ID INTEGER,
        Rank INTEGER,
        Points INTEGER,
        PRIMARY KEY(ID));
,
        CHECK (EXISTS (SELECT * FROM T20Batsman AS E1 WHERE E1.ID = ID)
                AND EXISTS (SELECT * FROM T20Bowler AS E1 WHERE E1.ID = ID)));
```

```
CREATE TABLE ODIInnings
        (MatchID INTEGER REFERENCES ODIMatch(ID),
        InningNum INTEGER NOT NULL,
        Score INTEGER NOT NULL,
        Wickets INTEGER NOT NULL,
        Overs INTEGER NOT NULL,
        BattingTeamName VARCHAR(30) REFERENCES ODITeam(CountryName),
        CHECK (Wickets < 11),
        CHECK (InningNum < 3 AND InningNum > 0),
        CHECK (Overs <= 50),
        PRIMARY KEY (MatchID, InningNum));

CREATE TABLE TestInnings
        (MatchID INTEGER REFERENCES TestMatch(ID),
        InningNum INTEGER NOT NULL,
        Score INTEGER NOT NULL,
        Wickets INTEGER NOT NULL,
        Overs INTEGER NOT NULL,
        BattingTeamName VARCHAR(30) REFERENCES TestTeam(CountryName),
        CHECK (Wickets < 11),
        CHECK (InningNum < 5 AND InningNum > 0),
        PRIMARY KEY (MatchID, InningNum));

CREATE TABLE T20Innings
        (MatchID INTEGER REFERENCES T20Match(ID),
        InningNum INTEGER NOT NULL,
        Score INTEGER NOT NULL,
        Wickets INTEGER NOT NULL,
        Overs INTEGER NOT NULL,
        BattingTeamName VARCHAR(20) REFERENCES T20Team(CountryName),
        CHECK (Wickets < 11),
        CHECK (InningNum < 3 AND InningNum > 0),
        CHECK (Overs <= 20),
        PRIMARY KEY (MatchID, InningNum));



CREATE TABLE TestBattingScoreCard
    (MatchID INTEGER,
```

```
    InningNum INTEGER,
    BatsmanID INTEGER REFERENCES TestBatsman(ID),
    RunsScored INTEGER NOT NULL,
    BallsFaced INTEGER NOT NULL,
    Fours INTEGER NOT NULL,
    Sixes INTEGER NOT NULL,
    StrikeRate REAL,
    BattingPosition INTEGER NOT NULL,
    PRIMARY KEY (MatchID, InningNum, BatsmanID),
    FOREIGN KEY (MatchID, InningNum) REFERENCES TestInnings(MatchID, InningNum),
CHECK (BattingPosition < 12 AND BattingPosition > 0));

CHECK (EXISTS (SELECT * FROM PlayedInTest AS E WHERE E.MatchID = MatchID AND
    E.PlayerID = BatsmanID)));

CREATE TABLE ODIBattingScoreCard
    (MatchID INTEGER,
    InningNum INTEGER,
    BatsmanID INTEGER REFERENCES ODIBatsman(ID),
    RunsScored INTEGER NOT NULL,
    BallsFaced INTEGER NOT NULL,
    Fours INTEGER NOT NULL,
    Sixes INTEGER NOT NULL,
    StrikeRate REAL,
    BattingPosition INTEGER NOT NULL,
    PRIMARY KEY (MatchID, InningNum, BatsmanID),
    FOREIGN KEY (MatchID, InningNum) REFERENCES ODIInnings(MatchID, InningNum),
CHECK (BattingPosition < 12 AND BattingPosition > 0));

CHECK (EXISTS (SELECT * FROM PlayedInODI AS E WHERE E.MatchID = MatchID AND
    E.PlayerID = BatsmanID)));

CREATE TABLE T20BattingScoreCard
    (MatchID INTEGER,
    InningNum INTEGER,
    BatsmanID INTEGER REFERENCES T20Batsman(ID),
    RunsScored INTEGER NOT NULL,
    BallsFaced INTEGER NOT NULL,
    Fours INTEGER NOT NULL,
    Sixes INTEGER NOT NULL,
    StrikeRate REAL,
    BattingPosition INTEGER NOT NULL,
    PRIMARY KEY (MatchID, InningNum, BatsmanID),
```

```
    FOREIGN KEY (MatchID, InningNum) REFERENCES T20Innings(MatchID, InningNum),
CHECK (BattingPosition < 12 AND BattingPosition > 0));
,
CHECK (EXISTS (SELECT * FROM PlayedInT20 AS E WHERE E.MatchID = MatchID AND
    E.PlayerID = BatsmanID)));


CREATE TABLE ODIBowlingScoreCard
(MatchID INTEGER,
InningNum INTEGER,
BowlerID INTEGER REFERENCES ODIBowler(ID),
OversBowled REAL NOT NULL,
MaidenOvers INTEGER NOT NULL,
Runs INTEGER NOT NULL,
Wickets INTEGER NOT NULL,
EconomyRate REAL NOT NULL,
PRIMARY KEY (MatchID, InningNum, BowlerID),
FOREIGN KEY (MatchID, InningNum) REFERENCES ODIInnings(MatchID, InningNum),
CHECK (Wickets < 11));

,
CHECK (EXISTS (SELECT * FROM PlayedInODI AS E WHERE E.MatchID = MatchID AND
    E.PlayerID = BowlerID)));


CREATE TABLE TestBowlingScoreCard
(MatchID INTEGER,
InningNum INTEGER,
BowlerID INTEGER REFERENCES TestBowler(ID),
OversBowled REAL NOT NULL,
MaidenOvers INTEGER NOT NULL,
Runs INTEGER NOT NULL,
Wickets INTEGER NOT NULL,
EconomyRate REAL NOT NULL,
PRIMARY KEY (MatchID, InningNum, BowlerID),
FOREIGN KEY (MatchID, InningNum) REFERENCES TestInnings(MatchID, InningNum),
CHECK (Wickets < 11));
,
CHECK (EXISTS (SELECT * FROM PlayedInTest AS E WHERE E.MatchID = MatchID AND
    E.PlayerID = BowlerID)));


CREATE TABLE T20BowlingScoreCard
```

```
(MatchID INTEGER,
InningNum INTEGER,
BowlerID INTEGER REFERENCES T20Bowler(ID),
OversBowled REAL NOT NULL,
MaidenOvers INTEGER NOT NULL,
Runs INTEGER NOT NULL,
Wickets INTEGER NOT NULL,
EconomyRate REAL NOT NULL,
PRIMARY KEY (MatchID, InningNum, BowlerID),
FOREIGN KEY (MatchID, InningNum) REFERENCES T20Innings(MatchID, InningNum),
CHECK (Wickets < 11));
,
CHECK (EXISTS (SELECT * FROM PlayedInT20 AS E WHERE E.MatchID = MatchID AND
    E.PlayerID = BowlerID)));


CREATE TABLE ODIPartnershipScoreCard
            (MatchID INTEGER,
            InningNum INTEGER,
            Batsman1ID INTEGER REFERENCES ODIBatsman(ID),
            Batsman2ID INTEGER REFERENCES ODIBatsman(ID),
            WicketNumber INTEGER NOT NULL,
            IncomingRuns INTEGER NOT NULL,
            PartnershipRuns INTEGER NOT NULL,
            PRIMARY KEY (MatchID, InningNum, WicketNumber),
            CHECK (Batsman1ID <> Batsman2ID),
            CHECK (WicketNumber < 11 AND WicketNumber > 0),
            FOREIGN KEY (MatchID, InningNum) REFERENCES ODIInnings(MatchID, InningNum));

            ,
CHECK ((EXISTS (SELECT * FROM PlayedInODI AS E WHERE E.MatchID = MatchID AND
    E.PlayerID = Batsman1ID)) AND
(EXISTS (SELECT * FROM PlayedInODI AS E WHERE E.MatchID = MatchID AND E.PlayerID =
    Batsman2ID)));

CREATE TABLE TestPartnershipScoreCard
            (MatchID INTEGER,
            InningNum INTEGER,
            Batsman1ID INTEGER REFERENCES TestBatsman(ID),
            Batsman2ID INTEGER REFERENCES TestBatsman(ID),
            WicketNumber INTEGER NOT NULL,
            IncomingRuns INTEGER NOT NULL,
            PartnershipRuns INTEGER NOT NULL,
```

```
            PRIMARY KEY (MatchID, InningNum, WicketNumber),
            CHECK (Batsman1ID <> Batsman2ID),
            CHECK (WicketNumber < 11 AND WicketNumber > 0),
            FOREIGN KEY (MatchID, InningNum) REFERENCES TestInnings(MatchID,
    InningNum));
            ,
CHECK ((EXISTS (SELECT * FROM PlayedInTest AS E WHERE E.MatchID = MatchID AND
    E.PlayerID = Batsman1ID)) AND
(EXISTS (SELECT * FROM PlayedInTest AS E WHERE E.MatchID = MatchID AND E.PlayerID =
    Batsman2ID)));


CREATE TABLE T20PartnershipScoreCard
            (MatchID INTEGER,
            InningNum INTEGER,
            Batsman1ID INTEGER REFERENCES T20Batsman(ID),
            Batsman2ID INTEGER REFERENCES T20Batsman(ID),
            WicketNumber INTEGER NOT NULL,
            IncomingRuns INTEGER NOT NULL,
            PartnershipRuns INTEGER NOT NULL,
            PRIMARY KEY (MatchID, InningNum, WicketNumber),
            CHECK (Batsman1ID <> Batsman2ID),
            CHECK (WicketNumber < 11 AND WicketNumber > 0),
            FOREIGN KEY (MatchID, InningNum) REFERENCES T20Innings(MatchID, InningNum));
            ,
CHECK ((EXISTS (SELECT * FROM PlayedInT20 AS E WHERE E.MatchID = MatchID AND
    E.PlayerID = Batsman1ID)) AND
(EXISTS (SELECT * FROM PlayedInT20 AS E WHERE E.MatchID = MatchID AND E.PlayerID =
    Batsman2ID)));



CREATE TABLE Team
        (CountryName VARCHAR(20) NOT NULL PRIMARY KEY);

CREATE TABLE ODITeam
        (CountryName VARCHAR(20) REFERENCES Team(CountryName),
        MatchesPlayed INTEGER,
        MatchesWon INTEGER,
        MatchesDrawn INTEGER,
        PRIMARY KEY (CountryName),
        CHECK (MatchesPlayed >= MatchesWon + MatchesDrawn));
```

```
CREATE TABLE TestTeam
        (CountryName VARCHAR(20) REFERENCES Team(CountryName),
        MatchesPlayed INTEGER,
        MatchesWon INTEGER,
        MatchesDrawn INTEGER,
        CHECK (MatchesPlayed >= MatchesWon + MatchesDrawn),
        PRIMARY KEY (CountryName));

CREATE TABLE T20Team
        (CountryName VARCHAR(20) REFERENCES Team(CountryName),
        MatchesPlayed INTEGER,
        MatchesWon INTEGER,
        MatchesDrawn INTEGER,
        CHECK (MatchesPlayed >= MatchesWon + MatchesDrawn),
        PRIMARY KEY (CountryName));

CREATE TABLE ODICoach
        (CountryName VARCHAR(20) REFERENCES ODITeam(CountryName),
        CoachID INTEGER REFERENCES Coach(ID),
        Role VARCHAR(20) NOT NULL,
        PRIMARY KEY (CoachID));

CREATE TABLE TestCoach
        (CountryName VARCHAR(20) REFERENCES TestTeam(CountryName),
        CoachID INTEGER REFERENCES Coach(ID),
        Role VARCHAR(20) NOT NULL,
        PRIMARY KEY (CoachID));

CREATE TABLE T20Coach
        (CountryName VARCHAR(20) REFERENCES T20Team(CountryName),
        CoachID INTEGER REFERENCES Coach(ID),
        Role VARCHAR(20) NOT NULL,
        PRIMARY KEY (CoachID));
```

➔ ODIRanking(CountryName, Rank, Points)

- CountryName references CountryName from ODITeam

```
CREATE TABLE ODIRanking
    (CountryName VARCHAR(20) REFERENCES ODITeam(CountryName),
    Rank INTEGER NOT NULL,
    Points INTEGER NOT NULL,
    PRIMARY KEY (CountryName));
```

```
CREATE TABLE TestRanking
    (CountryName VARCHAR(20) REFERENCES TestTeam(CountryName),
    Rank INTEGER NOT NULL,
    Points INTEGER NOT NULL,
    PRIMARY KEY (CountryName));

CREATE TABLE T20Ranking
    (CountryName VARCHAR(20) REFERENCES T20Team(CountryName),
    Rank INTEGER NOT NULL,
    Points INTEGER NOT NULL,
    PRIMARY KEY (CountryName));


CREATE TABLE Venue
            (ID SERIAL,
            Name VARCHAR(30) NOT NULL,
            Capacity INTEGER NOT NULL,
            City VARCHAR(20) NOT NULL,
            Country VARCHAR(20) NOT NULL,
            PRIMARY KEY (ID));


CREATE TYPE RESULTTYPE AS ENUM ('TEAMA', 'TEAMB',  'DRAW', 'TIE', 'NORESULT' );


CREATE TABLE Match(
            ID SERIAL,
            Date DATE NOT NULL,
            PRIMARY KEY (ID)
            );

CREATE TYPE DAYTYPE AS ENUM('DAY', 'DAYNIGHT');

CREATE TABLE ODIMatch
            (ID INTEGER REFERENCES Match(ID),
            Date DATE NOT NULL,
            Result RESULTTYPE,
            DayVersusDayNight DAYTYPE,
            Umpire1 INTEGER REFERENCES Umpire(ID),
            Umpire2 INTEGER REFERENCES Umpire(ID),
            Umpire3 INTEGER REFERENCES Umpire(ID),
            TossWonBy VARCHAR(20) REFERENCES ODITeam(CountryName),
            TeamA VARCHAR(20) REFERENCES ODITeam(CountryName),
```

```
        TeamB VARCHAR(20) REFERENCES ODITeam(CountryName),
        VenueID INTEGER REFERENCES Venue(ID),
        TeamACaptainID INTEGER REFERENCES ODIPlayer(ID),
        TeamBCaptainID INTEGER REFERENCES ODIPlayer(ID),
        TeamAWKID INTEGER REFERENCES ODIPlayer(ID),
        TeamBWKID INTEGER  REFERENCES ODIPlayer(ID),
        PRIMARY KEY (ID),
        CHECK (TeamA <> TeamB AND (TossWonBy = TeamA OR TossWonBy = TeamB)),
        CHECK (Umpire1 <> Umpire2 AND Umpire2 <> Umpire3 AND Umpire3 <> Umpire1)
        );

CREATE TABLE T20Match
        (ID INTEGER REFERENCES Match(ID),
        Date DATE NOT NULL,
        Result RESULTTYPE,
        DayVersusDayNight DAYTYPE,
        Umpire1 INTEGER REFERENCES Umpire(ID),
        Umpire2 INTEGER REFERENCES Umpire(ID),
        Umpire3 INTEGER REFERENCES Umpire(ID),
        TossWonBy VARCHAR(20) REFERENCES T20Team(CountryName),
        TeamA VARCHAR(20) REFERENCES T20Team(CountryName),
        TeamB VARCHAR(20) REFERENCES T20Team(CountryName),
        VenueID INTEGER REFERENCES Venue(ID),
        TeamACaptainID INTEGER REFERENCES T20Player(ID),
        TeamBCaptainID INTEGER REFERENCES T20Player(ID),
        TeamAWID INTEGER  REFERENCES T20Player(ID),
        TeamBWID INTEGER  REFERENCES T20Player(ID),
        PRIMARY KEY (ID),
        CHECK (TeamA <> TeamB AND (TossWonBy = TeamA OR TossWonBy = TeamB)),
        CHECK (Umpire1 <> Umpire2 AND Umpire2 <> Umpire3 AND Umpire3 <> Umpire1)
        );

CREATE TABLE TestMatch
        (ID INTEGER REFERENCES Match(ID),
        Date DATE NOT NULL,
        Result RESULTTYPE,
        Umpire1 INTEGER REFERENCES Umpire(ID),
        Umpire2 INTEGER REFERENCES Umpire(ID),
        Umpire3 INTEGER REFERENCES Umpire(ID),
        TossWonBy VARCHAR(20) REFERENCES TestTeam(CountryName),
        TeamA VARCHAR(20) REFERENCES TestTeam(CountryName),
        TeamB VARCHAR(20) REFERENCES TestTeam(CountryName),
        VenueID INTEGER REFERENCES Venue(ID),
```

```
    TeamACaptainID INTEGER REFERENCES TestPlayer(ID),
    TeamBCaptainID INTEGER REFERENCES TestPlayer(ID),
    TeamAWID INTEGER  REFERENCES TestPlayer(ID),
    TeamBWID INTEGER  REFERENCES TestPlayer(ID),
    PRIMARY KEY (ID),
    CHECK (TeamA <> TeamB AND (TossWonBy = TeamA OR TossWonBy = TeamB)),
    CHECK (Umpire1 <> Umpire2 AND Umpire2 <> Umpire3 AND Umpire3 <> Umpire1)
    );
```

➔ MatchAward(MatchID, PlayerID, AwardName)

- MatchID references ID from Match
- ID references ID from Player

```
CREATE TABLE MatchAward(
    MatchID INTEGER REFERENCES Match(ID),
    PlayerID INTEGER REFERENCES Player(ID),
    AwardName VARCHAR(30),
    PRIMARY KEY(MatchID, PlayerID));
```

```
    CHECK (EXISTS (SELECT * FROM PlayedInODI AS E1 WHERE E1.MatchID=MatchID
AND E1.PlayerID=PlayerID) OR
        EXISTS (SELECT * FROM PlayedInTest AS E1 WHERE E1.MatchID =
    MatchID AND E1.PlayerID=PlayerID OR
        EXISTS (SELECT * FROM PlayedInT20 AS E1 WHERE E1.MatchID = MatchID
    AND E1.PlayerID=PlayerID));
```

➔ InODITournament(MatchID, TournamentName, Year, Stage)

- MatchID references ID from ODIMatch
- (TournamentName, Year) references (Name, Year) from ODITournament

```
CREATE TABLE InODITournament(
    MatchID INTEGER REFERENCES ODIMatch(ID),
    TournamentName  VARCHAR(30),
    Year INTEGER,
    Stage VARCHAR(20),
    FOREIGN KEY(TournamentName, Year) REFERENCES ODITournament(Name,Year),
    PRIMARY KEY(MatchID, TournamentName, Year));
```

➔ PlayedInODI(MatchID, PlayerID)

- MatchID references ID from ODIMatch
- PlayerID references ID from ODIPlayer

```
CREATE TABLE PlayedInODI(
    MatchID INTEGER REFERENCES ODIMatch(ID),
```

```
    PlayerID INTEGER REFERENCES ODIPlayer(ID),
    PRIMARY KEY(MatchID, PlayerID));
```

➔ InTestTournament(<u>MatchID</u>, <u>TournamentName</u>, <u>Year</u>, Stage)

- MatchID references ID from TestMatch
- (TournamentName, Year) references (Name, Year) from TestTournament

```
CREATE TABLE InTestTournament(
    MatchID INTEGER REFERENCES TestMatch(ID),
    TournamentName  VARCHAR(30),
    Year INTEGER,
    Stage VARCHAR(20),
    FOREIGN KEY(TournamentName, Year) REFERENCES TestTournament(Name,Year),
    PRIMARY KEY(MatchID, TournamentName, Year));
```

➔ PlayedInTest(<u>MatchID</u>, <u>PlayerID</u>)

- MatchID references ID from TestMatch
- PlayerID references ID from TestPlayer

```
CREATE TABLE PlayedInTest(
    MatchID INTEGER REFERENCES TestMatch(ID),
    PlayerID INTEGER REFERENCES TestPlayer(ID),
    PRIMARY KEY(MatchID, PlayerID));
```

➔ InT20Tournament(<u>MatchID</u>, <u>TournamentName</u>, <u>Year</u>, Stage)

- MatchID references ID from T20Match
- (TournamentName, Year) references (Name, Year) from T20Tournament

```
CREATE TABLE InT20Tournament(
    MatchID INTEGER REFERENCES T20Match(ID),
    TournamentName  VARCHAR(30),
    Year INTEGER,
    Stage VARCHAR(20),
    FOREIGN KEY(TournamentName, Year) REFERENCES T20Tournament(Name,Year),
    PRIMARY KEY(MatchID, TournamentName, Year));
```

➔ PlayedInTest(<u>MatchID</u>, <u>PlayerID</u>)

- MatchID references ID from TestMatch
- PlayerID references ID from TestPlayer

```
CREATE TABLE PlayedInT20(
    MatchID INTEGER REFERENCES T20Match(ID),
    PlayerID INTEGER REFERENCES T20Player(ID),
```

```
    PRIMARY KEY(MatchID, PlayerID));
```

➔ Tournament(<u>Name</u>, <u>Year</u>)

```
CREATE TABLE Tournament(
    Name VARCHAR(30),
    Year INTEGER,
    PRIMARY KEY(Name, Year));
```

➔ ODITournament(<u>Name</u>, <u>Year</u>, Result)

```
CREATE TABLE ODITournament(
    Name VARCHAR(30),
    Year INTEGER,
    Result VARCHAR(30),
    PRIMARY KEY(Name, Year),
    FOREIGN KEY (Name, Year) REFERENCES Tournament(Name, Year));
```

- (Name, Year) references (Name, Year) from Tournament

➔ TestTournament(<u>Name</u>, <u>Year</u>, Result)

```
CREATE TABLE TestTournament(
    Name VARCHAR(30),
    Year INTEGER,
    Result  VARCHAR(30),
    PRIMARY KEY(Name, Year),
    FOREIGN KEY (Name, Year) REFERENCES Tournament(Name, Year));
```

- (Name, Year) references (Name, Year) from Tournament

➔ T20Tournament(<u>Name</u>, <u>Year</u>, Result)

```
CREATE TABLE T20Tournament(
    Name VARCHAR(30),
    Year INTEGER,
    Result  VARCHAR(30),
    PRIMARY KEY(Name, Year),
    FOREIGN KEY (Name, Year) REFERENCES Tournament(Name, Year));
```

- (Name, Year) references (Name, Year) from Tournament

➔ ODITournamentWinner(<u>Name</u>, <u>Year</u>, <u>CountryName</u>)

- (Name, Year) references (Name, Year) from ODITournament
- CountryName references CountryName from ODITeam

```
CREATE TABLE ODITournamentWinner(
        Name VARCHAR(30),
        Year INTEGER,
        CountryName VARCHAR(20) REFERENCES ODITeam(CountryName),
        PRIMARY KEY (Name, Year, CountryName),
        FOREIGN KEY (Name, Year) REFERENCES ODITournament(Name, Year)
        );


CREATE TABLE TestTournamentWinner(
        Name VARCHAR(30),
        Year INTEGER,
        CountryName VARCHAR(20) REFERENCES TestTeam(CountryName),
        PRIMARY KEY (Name, Year, CountryName),
        FOREIGN KEY (Name, Year) REFERENCES TestTournament(Name, Year)
        );


CREATE TABLE T20TournamentWinner(
        Name VARCHAR(30),
        Year INTEGER,
        CountryName VARCHAR(20) REFERENCES T20Team(CountryName),
        PRIMARY KEY (Name, Year, CountryName),
        FOREIGN KEY (Name, Year) REFERENCES T20Tournament(Name, Year)
        );
```

➔ ODITournamentAward(Name, Year, PlayerID, AwardName)

- (Name, Year) references (Name, Year) from ODITournament
- PlayerIDreferences ID from ODIPlayer

```
CREATE TABLE ODITournamentAward(
        Name VARCHAR(30),
        Year INTEGER,
        PlayerID INTEGER REFERENCES ODIPlayer(ID),
        AwardName VARCHAR(30),
        FOREIGN KEY (Name, Year) REFERENCES ODITournament (Name, Year),
        PRIMARY KEY (Name, Year, PlayerID));
,
        CHECK (EXISTS (SELECT * FROM InODITournament AS E1
                        WHERE E1.TournamentName = Name AND E1.Year = Year AND
                                EXISTS (SELECT * FROM PlayedInODI AS E2 WHERE
                                        PlayerID = E2.PlayerID AND
                                        E1.MatchID = E2.MatchID)))
```

```
);


CREATE TABLE TestTournamentAward(
        Name VARCHAR(30),
        Year INTEGER,
        PlayerID INTEGER REFERENCES TestPlayer(ID),
        AwardName VARCHAR(30),
        FOREIGN KEY (Name, Year) REFERENCES TestTournament (Name, Year),
        PRIMARY KEY (Name, Year, PlayerID));
,
        CHECK (EXISTS (SELECT * FROM InTestTournament AS E1
                        WHERE E1.TournamentName = Name AND E1.Year = Year AND
                                EXISTS (SELECT * FROM PlayedInTest AS E2 WHERE
                                        PlayerID = E2.PlayerID AND
                                        E1.MatchID = E2.MatchID)))
);


CREATE TABLE T20TournamentAward(
        Name VARCHAR(30),
        Year INTEGER,
        PlayerID INTEGER REFERENCES T20Player(ID),
        AwardName VARCHAR(30),
        FOREIGN KEY (Name, Year) REFERENCES T20Tournament (Name, Year),
        PRIMARY KEY (Name, Year, PlayerID));
,
        CHECK (EXISTS (SELECT * FROM InT20Tournament AS E1
                        WHERE E1.TournamentName = Name AND E1.Year = Year AND
                                EXISTS (SELECT * FROM PlayedInT20 AS E2 WHERE
                                        PlayerID = E2.PlayerID AND
                                        E1.MatchID = E2.MatchID)))
);



CREATE ASSERTION RankingsCorrect
CHECK ((NOT EXISTS (SELECT * FROM ODIRankings AS E1 WHERE EXISTS (SELECT *
    FROM ODIRankings AS E2 WHERE E1.Rank < E2.Rank AND E1.Points < E2.Points)))
AND (NOT EXISTS (SELECT * FROM TestRankings AS E1 WHERE EXISTS (SELECT * FROM
    TestRankings AS E2 WHERE E1.Rank < E2.Rank AND E1.Points < E2.Points)))
AND (NOT EXISTS (SELECT * FROM T20Rankings AS E1 WHERE EXISTS (SELECT * FROM
    T20Rankings AS E2 WHERE E1.Rank < E2.Rank AND E1.Points < E2.Points))));
```

```
CREATE ASSERTION ValidNumberOfInnings
CHECK ((NOT EXISTS (SELECT * FROM ODIInnings GROUP BY MatchID HAVING COUNT(*) >
    2)) AND
(NOT EXISTS (SELECT * FROM T20Innings GROUP BY MatchID HAVING COUNT(*) > 2)) AND
    (NOT EXISTS (SELECT * FROM TestInnings GROUP BY MatchID HAVING COUNT(*) > 4)));


CREATE ASSERTION CorrectPlayersFrom
CHECK ((NOT EXISTS
(SELECT * FROM ODIMatch AS E1 WHERE EXISTS
        (SELECT * FROM PlayedInODI AS E2 WHERE E2.MatchID = E1.ID AND
        (SELECT Country FROM IsInTeam WHERE PlayerID = E2.PlayeriD) <> E1.TeamA AND
        (SELECT Country FROM IsInTeam WHERE PlayerID = E2.PlayeriD) <> E1.TeamB)))
        AND
(NOT EXISTS
(SELECT * FROM TestMatch AS E1 WHERE EXISTS
        (SELECT * FROM PlayedInTest AS E2 WHERE E2.MatchID = E1.ID AND
        (SELECT Country FROM IsInTeam WHERE PlayerID = E2.PlayeriD) <> E1.TeamA AND
        (SELECT Country FROM IsInTeam WHERE PlayerID = E2.PlayeriD) <> E1.TeamB)))
        AND
(NOT EXISTS
(SELECT * FROM T20Match AS E1 WHERE EXISTS
        (SELECT * FROM PlayedInT20 AS E2 WHERE E2.MatchID = E1.ID AND
        (SELECT Country FROM IsInTeam WHERE PlayerID = E2.PlayeriD) <> E1.TeamA AND
        (SELECT Country FROM IsInTeam WHERE PlayerID = E2.PlayeriD) <> E1.TeamB))));

CREATE ASSERTION TestBowlerPerformance CHECK(
    NOT EXISTS(
            SELECT * FROM TestBowler
            WHERE
            (BestMatchPerformanceWickets < BestInningsPerformanceWickets)));
```