WAFFL CONSTRAINTS

WAFFL.Conferences

* PRIMARY KEY: ConferenceID = NOT NULL
* ConferenceName = NOT NULL, UNIQUE

WAFFL.Teams

* PRIMARY KEY: TeamID = NOT NULL
* TeamName = NOT NULL, UNIQUE
* FOREIGN KEY: ConferenceID = NOT NULL
* Wins = NOT NULL, CHECKS >= 0
* Loses = NOT NULL, CHECKS >= 0

WAFFL.Players

* PRIMARY KEY: PlayerID = NOT NULL
* FirstName = NOT NULL
* LastName = NOT NULL
* Age = NOT NULL, CHECKS >= 13 & <=17
* FOREIGN KEY: TeamID = NOT NULL

WAFFL.Referees

* PRIMARY KEY: RefereeID = NOT NULL
* FirstName = NOT NULL
* LastName = NOT NULL

WAFFL.Locations

* PRIMARY KEY: LocationID = NOT NULL
* ParkName = NOT NULL
* LocationAddress = NOT NULL

WAFFL.Games

* PRIMARY KEY: GameID = NOT NULL
* GameDate = NOT NULL
* FOREIGN KEY: LocationID = NOT NULL
* FOREIGN KEY: TeamID\_1 = NOT NULL
* FOREIGN KEY: TeamID\_2 = NOT NULL
* FOREIGN KEY: RefeereeID = NOT NULL

WAFFL.Results

* PRIMARY KEY: ResultID = NOT NULL
* FOREIGN KEY: GameID = NOT NULL
* FOREIGN KEY: TeamID\_Winner = NOT NULL
* Team1\_Score = NOT NULL, CHECKS >=0
* Team2\_Score = NOT NULL, CHECKS >=0

Relationship at this time look like this:

Conference can have many teams.

Teams can only have one Conference.

Teams can have many Players.

Players can have only one team.

Location can have host many games.

A Game can only have one location.

Games will only have one result.

And Refs can officiate many games.

Stored Procedures

* WAFFL.PlayerByTeam Retrieves players in a given team based on TeamID.
* WAFFL.TeamByConference Retrieves teams that belong to a conference.
* WAFFL.PlayersByPosition Retrieves players based on their position.
* WAFFL.GameResults: Retrieves game results.
* WAFFL.GameRefs: Retrieves games along with their referees.
* WAFFL.GameRefsADV: Retrieves games and referee names more detailed.

Triggers

* WAFFL.tr\_Teams\_RecordAfterUpdate: Trigger that updates a team win/loss record, ensuring wins and loses are not decreased and updates history.
* WAFFL.tr\_Players\_AfterDelete: Trigger that performs actions after a player is deleted, checks for player existence, and updates history.