**Stored Procedures**

1. Display the home team, stadium name, city, state, capacity, surface, and year opened for the user’s choice of team. **DONE**

delimiter $$

drop procedure if exists getHomeStadium;

create procedure getHomeStadium(IN teamName VARCHAR(255), INOUT stadiumDetails VARCHAR(4000))

begin

declare isDone integer default 0;

declare homeTeam varchar(255) default “”;

declare stadiumName varchar(255) default “”;

declare cityName varchar(255) default “”;

declare stateName varchar(255) default “”;

declare capacity bigint default 0;

declare surface varchar(255) default “”;

declare yearOpened bigint default 0;

declare stadiumCursor cursor for

**SELECT** **DISTINCT** t.name, s.name, s.city, s.state, s.capacity, s.surface, s.year\_opened

**FROM** Stadium s, Game g, Team t

**WHERE** s.stadium\_code = g.stadium\_code **AND** g.home\_team\_code = t.team\_code **AND** t.name = teamName;

declare continue handler

for not found set isDone = 1;

open stadiumCursor;

getList: loop

fetch stadiumCursor into homeTeam, stadiumName, cityName, stateName, capacity, surface, yearOpened;

if isDone = 1 then

leave getList;

end if;

set stadiumDetails = concat(“”, “\n”, stadiumDetails);

set stadiumDetails = concat(yearOpened, “\t”, stadiumDetails);

set stadiumDetails = concat(surface, “\t”, stadiumDetails);

set stadiumDetails = concat(capacity, “\t”, stadiumDetails);

set stadiumDetails = concat(stateName, “\t”, stadiumDetails);

set stadiumDetails = concat(cityName, “\t”, stadiumDetails);

set stadiumDetails = concat(stadiumName, “\t”, stadiumDetails);

set stadiumDetails = concat(homeTeam, “\t\t”, stadiumDetails);

end loop getList;

close stadiumCursor;

end $$

delimiter;

1. Display the player for a specified team with the most touchdowns (Rec TD + Rush TD + Pass TD) in each game. List the game date, player with most TDs, the amount of touchdowns, and the amount of both receiving and rushing yards **DONE**

delimiter $$

drop procedure if exists getPlayerWithMostTouchdowns;

create procedure getPlayerWithMostTouchdowns(IN teamName VARCHAR(255), INOUT playerDetails VARCHAR(4000))

begin

declare isDone integer default 0;

declare gameDate date default null;

declare firstName varchar(255) default “”;

declare lastName varchar(255) default “”;

declare touchdowns bigint default 0;

declare rushYards bigint default 0;

declare recYards bigint default 0;

declare playerCursor cursor for

**SELECT** g.date, p.first\_name, p.last\_name, **MAX**(ps.rush\_td + ps.rec\_td + ps.pass\_td), ps.rush\_yard, ps.rec\_yards

**FROM** Team t, Player p, Player\_Statistics ps, Stadium s, Game g

**WHERE** t.name = teamName **AND** ((t.team\_code = g.home\_team\_code) **OR** (t.team\_code = g.visit\_team\_code)) **AND** g.game\_code = ps.game\_code **AND** ps.player\_code = p.player\_code

**GROUP BY** g.date;

declare continue handler

for not found set isDone = 1;

open playerCursor;

getList: loop

fetch playerCursor into gameDate, firstName, lastName, touchdowns, rushYards, recYards;

if isDone = 1 then

leave getList;

end if;

set playerDetails = concat(“”, “\n”, playerDetails);

set playerDetails = concat(recYards, “\t”, playerDetails);

set playerDetails = concat(rushYards, “\t”, playerDetails);

set playerDetails = concat(touchdowns, “\t”, playerDetails);

set playerDetails = concat(lastName, “\t”, playerDetails);

set playerDetails = concat(firstName, “\t”, playerDetails);

set playerDetails = concat(gameDate, “\t\t”, playerDetails);

end loop getList;

close playerCursor;

end $$

delimiter;

1. Display the total season stats for the player of our choice. **DONE**

delimiter $$

drop procedure if exists getTotalSeasonStats;

create procedure getTotalSeasonStats(IN firstName VARCHAR(255), lastName VARCHAR(255), INOUT seasonStats VARCHAR(4000))

begin

declare isDone integer default 0;

declare playerFirstname varchar(255) default “”;

declare playerLastname varchar(255) default “”;

declare totalPassAttempts bigint default 0;

declare totalPassCompletions bigint default 0;

declare totalPassYards bigint default 0;

declare totalPassTouchdowns bigint default 0;

declare totalRushTouchdowns bigint default 0;

declare totalRushYards bigint default 0;

declare totalRecTouchdowns bigint default 0;

declare totalRecYards bigint default 0;

declare statsCursor cursor for

**SELECT** p.first\_name, p.last\_name, **SUM**(ps.pass\_attempt), **SUM**(ps.pass\_completion), **SUM**(ps.pass\_yard), **SUM**(ps.pass\_td), **SUM**(ps.rush\_td), **SUM**(ps.rush\_yard), **SUM**(ps.rec\_td), **SUM**(ps.rec\_yards)

**FROM** Player p, Player\_Statistics ps

**WHERE** p.first\_name = firstName **AND** p.last\_name = lastName **AND** p.player\_code = ps.player\_code;

declare continue handler

for not found set isDone = 1;

open statsCursor;

getList: loop

fetch statsCursor into playerFirstName, playerLastName, totalPassAttempts, totalPassCompletions, totalPassYards, totalPassTouchdowns, totalRushTouchdowns, totalRushYards, totalRecTouchdowns, totalRecYards;

if isDone = 1 then

leave getList;

end if;

set seasonStats = concat(“”, “\n”, seasonStats);

set seasonStats = concat(totalRecYards, “\t”, seasonStats);

set seasonStats = concat(totalRecTouchdowns, “\t”, seasonStats);

set seasonStats = concat(totalRushYards, “\t”, seasonStats);

set seasonStats = concat(totalRushTouchdowns, “\t”, seasonStats);

set seasonStats = concat(totalPassTouchdowns, “\t”, seasonStats);

set seasonStats = concat(totalPassYards, “\t”, seasonStats);

set seasonStats = concat(totalPassCompletions, “\t”, seasonStats);

set seasonStats = concat(totalPassAttempts, “\t”, seasonStats);

set seasonStats = concat(playerLastName, “\t”, seasonStats);

set seasonStats = concat(playerFirstName, “\t\t”, seasonStats);

end loop getList;

close statsCursor;

end $$

delimiter;

1. Display the participating teams, date, attendance, and location of the game with the highest attendance between two specified dates. **DONE**

delimiter $$

drop procedure if exists getHighestAttendanceGame;

create procedure getHighestAttendanceGame(IN startDate DATE, endDate DATE, INOUT gameDetails VARCHAR(4000))

begin

declare isDone integer default 0;

declare homeTeam varchar(255) default “”;

declare visitTeam varchar(255) default “”;

declare gameDate date default null;

declare maxAttendance bigint default 0;

declare stadiumName varchar(255) default “”;

declare cityName varchar(255) default “”;

declare stateName varchar(255) default “”;

declare gameCursor cursor for

**SELECT** t1.name, t2.name, g.date, **MAX**(gs.attendance), s.name, s.city, s.state

**FROM** Team t1, Team t2, Game g, Game\_Statistics gs, Stadium s

**WHERE** g.home\_team\_code = t1.team\_code **AND** g.visit\_team\_code = t2.team\_code **AND** gs.game\_code = g.game\_code **AND** g.stadium\_code = s.stadium\_code **AND**

g.date >= startDate **AND** g.date <= endDate;

declare continue handler

for not found set isDone = 1;

open gameCursor;

getList: loop

fetch gameCursor into homeTeam, visitTeam, gameDate, maxAttendance, stadiumName, cityName, stateName;

if isDone = 1 then

leave getList;

end if;

set gameDetails = concat(“”, “\n”, gameDetails);

set gameDetails = concat(stateName, “\t”, gameDetails);

set gameDetails = concat(cityName, “\t”, gameDetails);

set gameDetails = concat(stadiumName, “\t”, gameDetails);

set gameDetails = concat(maxAttendance, “\t”, gameDetails);

set gameDetails = concat(gameDate, “\t”, gameDetails);

set gameDetails = concat(visitTeam, “\t”, gameDetails);

set gameDetails = concat(homeTeam, “\t\t”, gameDetails);

end loop getList;

close gameCursor;

end $$

delimiter;

1. Display the home team, visit team, date, attendance, and duration of all the games that took place in a user specified stadium. **DONE**

delimiter $$

drop procedure if exists getGameList;

create procedure getGameList(IN stadiumName VARCHAR(255), cityName VARCHAR(255), stateName VARCHAR(255), INOUT gameList VARCHAR(4000))

begin

declare isDone integer default 0;

declare currentHomeTeam varchar(255) default “”;

declare currentVisitTeam varchar(255) default “”;

declare currentDate date default null;

declare currentAttendance bigint default 0;

declare currentDuration bigint default 0;

declare gameCursor cursor for

**SELECT** t1.name, t2.name, g.date, gs.attendance, gs.duration

**FROM** Team t1, Team t2, Game g, Game\_Statistics gs, Stadium s

**WHERE** t1.team\_code = g.home\_team\_code **AND** t2.team\_code = g.visit\_team\_code **AND** gs.game\_code = g.game\_code **AND** g.stadium\_code = s.stadium\_code **AND** s.name = stadiumName **AND** s.city = cityName **AND** s.state = stateName;

declare continue handler

for not found set isDone = 1;

open gameCursor;

getList: loop

fetch gameCursor into currentHomeTeam, currentVisitTeam, currentDate, currentAttendance, currentDuration;

if isDone = 1 then

leave getList;

end if;

set gameList = concat(“”, “\n”, gameList);

set gameList = concat(currentDuration, “\t”, gameList);

set gameList = concat(currentAttendance, “\t”, gameList);

set gameList = concat(currentDate, “\t”, gameList);

set gameList = concat(currentVisitTeam, “\t”, gameList);

set gameList = concat(currentHomeTeam, “\t\t”, gameList);

end loop getList;

close gameCursor;

end $$

delimiter;

1. Display the stats of a user specified player for a game between the user’s choice of two teams. **DONE**

delimiter $$

drop procedure if exists getPlayerStats;

create procedure getPlayerStats(IN firstName VARCHAR(255), lastName VARCHAR(255), team1 VARCHAR(255), team2 VARCHAR(255), INOUT playerStats VARCHAR(255))

begin

declare isDone integer default 0;

declare playerFirstName varchar(255) default “”;

declare playerLastName varchar(255) default “”;

declare passAttempts bigint default 0;

declare passCompletions bigint default 0;

declare passYards bigint default 0;

declare passTouchdowns bigint default 0;

declare rushTouchdowns bigint default 0;

declare rushYards bigint default 0;

declare recTouchdowns bigint default 0;

declare recYards bigint default 0;

declare statsCursor cursor for

**SELECT** p.first\_name, p.last\_name, ps.pass\_attempt, ps.pass\_completion, ps.pass\_yard, ps.pass\_td, ps.rush\_td, ps.rush\_yard, ps.rec\_td, ps.rec\_yards

**FROM** Player p, Player\_Statistics ps, Team t1, Team t2, Game g

**WHERE** t1.name = team1 **AND** t2.name = team2 **AND** ((t1.team\_code = g.home\_team\_code **AND** t2.team\_code = g.visit\_team\_code) **OR** (t1.team\_code = g.visit\_team\_code **AND** t2.team\_code = g.home\_team\_code)) **AND** g.game\_code = ps.game\_code **AND** p.player\_code = ps.player\_code **AND** p.first\_name = firstName **AND** p.last\_name = lastName;

declare continue handler

for not found set isDone = 1;

open statsCursor;

getList: loop

fetch statsCursor into playerFirstName, playerLastName, passAttempts, passCompletions, passYards, passTouchdowns, rushTouchdowns, rushYards, recTouchdowns, recYards;

if isDone = 1 then

leave getList;

end if;

set playerStats = concat(“”, “\n”, playerStats);

set playerStats = concat(recYards, “\t”, playerStats);

set playerStats = concat(recTouchdowns, “\t”, playerStats);

set playerStats = concat(rushYards, “\t”, playerStats);

set playerStats = concat(rushTouchdowns, “\t”, playerStats);

set playerStats = concat(passTouchdowns, “\t”, playerStats);

set playerStats = concat(passYards, “\t”, playerStats);

set playerStats = concat(passCompletions, “\t”, playerStats);

set playerStats = concat(passAttempts, “\t”, playerStats);

set playerStats = concat(playerLastName, “\t”, playerStats);

set playerStats = concat(playerFirstName, “\t\t”, playerStats);

end loop getList;

close statsCursor;

end $$

delimiter;

1. Display the name, position, class, # of touchdowns, and uniform number of the top 5 players from a specific team for that particular season. **DONE**

delimiter $$

drop procedure if exists getTopFivePlayers;

create procedure getTopFivePlayers(IN teamName VARCHAR(255), INOUT playerList VARCHAR(4000))

begin

declare isDone integer default 0;

declare currentFirstName varchar(255) default “”;

declare currentLastName varchar(255) default “”;

declare currentClass varchar(255) default “”;

declare currentTotalTouchdowns bigint default 0;

declare currentUniformNumber bigint default 0;

declare playerCursor cursor for

**SELECT** p.first\_name, p.last\_name, p.class, **SUM**(ps.rush\_TD + ps.rec\_TD + ps.pass\_TD) **AS** total\_touchdowns, p.uniform\_number

**FROM** Player p, Player\_Statistics ps, Team t, Game g

**WHERE** p.player\_code = ps.player\_code **AND** ps.game\_code = g.game\_code **AND** t.name = teamName **AND** t.team\_code = p.team\_code

**GROUP BY** p.player\_code

**ORDER BY** total\_touchdowns **DESC** limit 5;

declare continue handler

for not found set isDone = 1;

open playerCursor;

getList: loop

fetch playerCursor into currentFirstName, currentLastName, currentClass, currentTotalTouchdowns, currentUniformNumber;

if isDone = 1 then

leave getList;

end if;

set playerList = concat(“”, “\n”, playerList);

set playerList = concat(currentUniformNumber, “\t”, playerList);

set playerList = concat(currentTotalTouchdowns, “\t”, playerList);

set playerList = concat(currentClass, “\t”, playerList);

set playerList = concat(currentLastName, “\t”, playerList);

set playerList = concat(currentFirstName, “\t\t”, playerList);

end loop getList;

close playerCursor;

end $$

delimiter;

1. Display the total season stats for each player playing the user specified position. **DONE**

delimiter $$

drop procedure if exists getPositonSpecificTotalSeasonStats;

create procedure getPositionSpecificTotalSeasonStats(IN positionName CHAR(3),  INOUT playerList LONGTEXT)

begin

declare isDone integer default 0;

declare currentFirstName varchar(255) default “”;

declare currentLastName varchar(255) default “”;

declare currentTotalPassAttempts bigint default 0;

declare currentTotalPassCompletions bigint default 0;

declare currentTotalPassYards bigint default 0;

declare currentTotalPassTouchdowns bigint default 0;

declare currentTotalRushTouchdowns bigint default 0;

declare currentTotalRushYards bigint default 0;

declare currentTotalRecTouchdowns bigint default 0;

declare currentTotalRecYards bigint default 0;

declare playerCursor cursor for

**SELECT** p.first\_name, p.last\_name, **SUM**(ps.pass\_attempt), **SUM**(ps.pass\_completion), **SUM**(ps.pass\_yard), **SUM**(ps.pass\_td), **SUM**(ps.rush\_td), **SUM**(ps.rush\_yard), **SUM**(ps.rec\_td), **SUM**(ps.rec\_yards)

**FROM** Player p, Player\_Statistics ps

**WHERE** p.player\_code = ps.player\_code **AND** p.position = positionName

**GROUP BY** p.player\_code;

declare continue handler

for not found set isDone = 1;

open playerCursor;

getList: loop

fetch playerCursor into currentFirstName, currentLastName, currentTotalPassAttempts, currentTotalPassCompletions, currentTotalPassYards, currentTotalPassTouchdowns, currentTotalRushTouchdowns, currentTotalRushYards, currentTotalRecTouchdowns, currentTotalRecYards;

if isDone = 1 then

leave getList;

end if;

set playerList = concat(“”, “\n”, playerList);

set playerList = concat(currentTotalRecYards, “\t”, playerList);

set playerList = concat(currentTotalRecTouchdowns, “\t”, playerList);

set playerList = concat(currentTotalRushYards, “\t”, playerList);

set playerList = concat(currentTotalRushTouchdowns, “\t”, playerList);

set playerList = concat(currentTotalPassTouchdowns, “\t”, playerList);

set playerList = concat(currentTotalPassYards, “\t”, playerList);

set playerList = concat(currentTotalPassCompletions, “\t”, playerList);

set playerList = concat(currentTotalPassAttempts, “\t”, playerList);

set playerList = concat(currentLastName, “\t”, playerList);

set playerList = concat(currentFirstName, “\t\t”, playerList);

end loop getList;

close playerCursor;

end $$

delimiter;

1. Display the complete list of the players playing the user specified position including all of their full names, class, position, and jersey numbers for a team of the user’s choice. **DONE**

delimiter $$

drop procedure if exists getPlayerListFromTeam;

create procedure getPlayerListFromTeam(IN positionName CHAR(3), IN teamName VARCHAR(255), INOUT playerList VARCHAR(4000))

begin

declare isDone integer default 0;

declare currentFirstName varchar(255) default “”;

declare currentLastName varchar(255) default “”;

declare currentClass char(2) default “”;

declare currentPosition char(3) default “”;

declare currentUniformNumber bigint default 0;

declare playerCursor cursor for

**SELECT** p.first\_name, p.last\_name, p.class, p.position, p.uniform\_number

**FROM** Player p, Team t

**WHERE** t.team\_code = p.team\_code **AND** p.position = positionName **AND** t.name = teamName;

declare continue handler

for not found set isDone = 1;

open playerCursor;

getList: loop

fetch playerCursor into currentFirstName, currentLastName, currentClass, currentPosition, currentUniformNumber;

if isDone = 1 then

leave getList;

end if;

set playerList = concat(“”, “\n”, playerList);

set playerList = concat(currentUniformNumber, “\t”, playerList);

set playerList = concat(currentPosition, “\t”, playerList);

set playerList = concat(currentClass, “\t”, playerList);

set playerList = concat(currentLastName, “\t”, playerList);

set playerList = concat(currentFirstName, “\t\t”, playerList);

end loop getList;

close playerCursor;

end $$

delimiter;