Development of a Better System of Stat Collection for the AMCC Conference

**CIST 0209 Web Databases Final Project**



4/23/2013

University of Pittsburgh at Bradford

Adebimbola Ogunyale

Table of Contents

**Table of Contents------------------------------------------------------------Page 1**

**Introduction-------------------------------------------------------------------Page 2**

* **Enterprise Overview---------------------------------------Page 2**
* **Business Problems to Solve--------------------------Page 2-3**
* **Project Proposal---------------------------------------------Page 3**

**Conceptual Data Model--------------------------------------------------Page 4**

**Physical Data Model--------------------------------------------------Page 5-11**

**Analysis------------------------------------------------------------------------Page 12**

**Appendix----------------------------------------------------------------------Page 13**

**References---------------------------------------------------------------Page 13**

**Enterprise Overview**

As a member of the Pitt-Bradford Men’s soccer team, I participate in games set up by the AMCC. AMCC stands for Allegheny Mountain Collegiate Conference. This conference began as a 501(c)(3), otherwise known as a non-profit organization during the 1997-98 academic school year. Dr. Richard McDowell, then the president of the University of Pittsburgh at Bradford, took the lead in forming the AMCC conference. The AMCC conference competes at the NCAA Division III level. The inaugural year had 6 charter member schools that began it all. Today, the conference has expanded to 10 total schools. The schools stretch across 3 states, Pennsylvania, Ohio and New York. These ten schools compete in 9 different sports in both men and women competition. The winner in each of these sports except cross country and men’s volleyball all receive automatic bids to the NCAA national tournament.

Being a conference at the NCAA Division III level, strong emphasis is placed on student performance both in the classroom, in competition and in the community. The AMCC promotes clean, fair play at all sporting events to the fans and players with the motto, Be loud, Be Proud and Be Positive. In addition to this, weekly acts of community service are promoted on the conference’s website amcconf.org. The NCAA promotes academics at Division III institutions and the AMCC takes it a step further by awarding students for their academic achievement. Finally, the AMCC is an athletic conference that strives to produce the best student-athletes they can by announcing weekly top performers as well as end of the season all-conference teams.

**Business Problems to solve**

The problem I will investigate is that of the AMCC athletic conference. The University of Pittsburgh at Bradford athletic teams are members of this conference. One major problem is a consistent collection of conference stats weekly. These stats can also come from non-conference competition play. The other problem is the website itself. The conference website amcconf.org is rather basic compared to other major collegiate athletic conference websites. This contributes to the issue of a consistent and well organized system of stat keepings.

This problem becomes apparent when trying to compare weekly stats of players in the conference. For example, player of the week honors are awarded on a weekly basis throughout the season. However, there is no currently no way to see the stats of all of the top performers of the week. This raises the issue, how does one know who deserves player of the week honors without being able to track stats?

**Goals and Objectives**

The goal of this project was to potentially create a system in which the AMCC can use to better track their stats. The project will provide student-athletes and fans the opportunity to see the weekly performers based on stats. When broken down further, these top performers can be grouped by class, position and school.

**Description of the Project**

In order to complete this project, I will first identify the different entities and attributes to use to sort the data that will be collected on a weekly basis. I will then sort the data using SQL commands and tables in a database. This database will hold and continue to acquire weekly stats that will be sorted and presented to users in a way that is easily navigational on the conference website amcconf.org.

Within this database, I will have several entities and attributes. A few of the entities will include player names, schools, Week, position (goals, assists) stats. These are just four of main entities that will be used to organize this data. I will also use two join tables between names and positions and player, week and stats.

**Timeline**

Jan 13- Topic Due

Jan 28- Conceptual Model

Feb 4- Logical Model

Feb 11- Table Creation Script

Feb 18- Constraints and Index Script

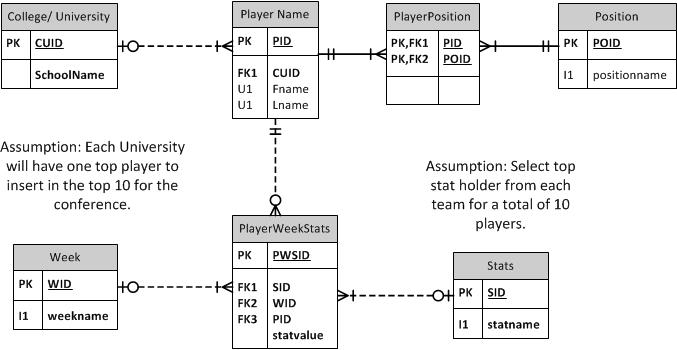
Feb 25- Sample Data Script

Mar 18- Sample Reports

Apr 1- Constraints and Index Script

Apr 8- Physical Database Design

Apr 22- Final Project Completed

**ER Diagram** 

**Physical Data Model Script**

use master;

go

drop database AMCCStats;

go

create database AMCCStats;

go

use AMCCStats;

go

create table University(

CUID int identity primary key,

SchoolName varchar(50) not null);

create table Player\_Name(

PID int identity primary key,

CUID int references University(CUID),

Fname varchar(30) not null,

Lname varchar(30) not null);

create table Position(

POID int identity primary key,

positionname varchar(25));

create table Week(

WID int identity primary key,

weekname varchar(25));

create table Statistic(

SID int identity primary key,

statname varchar(25) not null);

create table PlayerPosition(

PID int references Player\_Name(PID) not null,

POID int references Position(POID) not null

primary key (PID,POID));

create table PlayerWeekStats(

PWSID int identity primary key,

SID int references Statistic(SID) not null,

WID int references Week(WID) not null,

PID int references Player\_Name(PID) not null,

statvalue int not null);

create unique index Lname\_idx on Player\_Name(Lname,Fname);

create index week\_idx on Week(weekname);

create index stat\_idx on Statistic(statname);

create index position\_idx on Position(positionname);

-- Insert Statements for University

insert into University (SchoolName)

values ('DYouville');

insert into University (SchoolName)

values ('Hilbert');

insert into University (SchoolName)

values ('Medaille');

insert into University (SchoolName)

values ('Mt. Aloysius');

insert into University (SchoolName)

values ('Penn State Behrend');

insert into University (SchoolName)

values ('Penn State Altoona');

insert into University (SchoolName)

values ('Pitt Bradford');

insert into University (SchoolName)

values ('Pitt Greensburg');

-- Insert Statements for Week

insert into Week (weekname)

values (1);

insert into Week (weekname)

values (2);

insert into Week (weekname)

values (3);

insert into Week (weekname)

values (4);

insert into Week (weekname)

values (5);

insert into Week (weekname)

values (6);

insert into Week (weekname)

values (7);

insert into Week (weekname)

values (8);

insert into Week (weekname)

values (9);

insert into Week (weekname)

values (10);

insert into Week (weekname)

values (11);

insert into Week (weekname)

values (12);

-- Insert Statements for Position

insert into Position (positionname)

values ('Goalkeeper');

insert into Position (positionname)

values ('Defender');

insert into Position (positionname)

values ('Midfield');

insert into Position (positionname)

values ('Forward');

-- Insert Statements for Player\_Name

insert into Player\_Name (Lname, Fname)

values ('Tor', 'George');

insert into Player\_Name (Lname, Fname)

values ('Gamble', 'Jake');

insert into Player\_Name (Lname, Fname)

values ('Williams', 'Mario');

insert into Player\_Name (Lname, Fname)

values ('Forster', 'Adam');

insert into Player\_Name (Lname, Fname)

values ('Kraska', 'Jeremy');

insert into Player\_Name (Lname, Fname)

values ('Greenwood', 'Jamie');

-- Insert Statements for Statistic

insert into Statistic (statname)

values ('Shots');

insert into Statistic (statname)

values ('Goals');

insert into Statistic (statname)

values ('Assists');

insert into Statistic (statname)

values ('Saves');

insert into Statistic (statname)

values ('Shutouts');

-- Insert Statements for PlayerPosition

insert into PlayerPosition (PID, POID)

values (1,4);

insert into PlayerPosition (PID, POID)

values (2,3);

insert into PlayerPosition (PID, POID)

values (3,3);

insert into PlayerPosition (PID, POID)

values (4,1);

insert into PlayerPosition (PID, POID)

values (5,1);

insert into PlayerPosition (PID, POID)

values (6,1);

update Player\_Name

set CUID = 3

where Lname = 'tor';

update Player\_Name

set CUID = 5

where Lname = 'Gamble';

update Player\_Name

set CUID = 2

where Lname = 'Williams';

update Player\_Name

set CUID = 7

where Lname = 'Forster';

update Player\_Name

set CUID = 8

where Lname = 'Kraska';

update Player\_Name

set CUID = 4

where Lname = 'Greenwood';

-- Insert Statements for PlayerWeekStats

insert into PlayerWeekStats (SID, WID, PID, statvalue)

values (3, 1, 1, 2);

insert into PlayerWeekStats (SID, WID, PID, statvalue)

values (2, 1, 1, 1);

insert into PlayerWeekStats (SID, WID, PID, statvalue)

values (3, 1, 2, 1);

insert into PlayerWeekStats (SID, WID, PID, statvalue)

values (2, 1, 2, 3);

insert into PlayerWeekStats (SID, WID, PID, statvalue)

values (1, 1, 3, 4);

insert into PlayerWeekStats (SID, WID, PID, statvalue)

values (4, 1, 4, 12);

insert into PlayerWeekStats (SID, WID, PID, statvalue)

values (5, 1, 4, 2);

insert into PlayerWeekStats (SID, WID, PID, statvalue)

values (4, 1, 5, 8);

insert into PlayerWeekStats (SID, WID, PID, statvalue)

values (5, 1, 5, 1);

insert into PlayerWeekStats (SID, WID, PID, statvalue)

values (4, 1, 6, 9);

insert into PlayerWeekStats (SID, WID, PID, statvalue)

values (5, 1, 6, 0);

go

-- View Statement for Goals of Week 1

CREATE VIEW Goals\_for\_week\_1 AS

select pn.Fname + ' ' + pn.Lname as 'Player Name', s.statname, pws.statvalue

from PlayerWeekStats pws join Player\_Name pn on pws.PID = pn.PID join Statistic s on pws.SID = s.SID

where s.statname = 'goals';

go

-- View Statement for Saves of Week 1

CREATE VIEW Saves\_for\_week\_1 AS

select pn.Fname + ' ' + pn.Lname as 'Player Name', s.statname, pws.statvalue

from PlayerWeekStats pws join Player\_Name pn on pws.PID = pn.PID join Statistic s on pws.SID = s.SID

where s.statname = 'saves';

go

-- View Statement for Total Stats for a Week

CREATE VIEW Total\_week\_Stats AS

select PID, SUM(statvalue) as 'Total Week Stats'

from PlayerWeekStats

group by PID;

go

select \* from Goals\_for\_week\_1;

select \* from Saves\_for\_week\_1;

select \* from Total\_week\_Stats;

go

-- 1) A query based stored procedure that accepts input and produces results based on the input.

drop procedure loadplayer;

go

create procedure loadplayer

@CUID int,

@Fname varchar(30),

@Lname varchar(30)

AS

BEGIN

insert into Player\_Name(CUID, Fname, Lname)

values (@CUID, @Fname, @Lname)

END;

exec loadplayer 7,'Adebimbola', 'Ogunyale';

-- 2) A stored procedure that loads data into a join table and the two tables it joins.

drop procedure loadPlayer\_Position;

go

create procedure loadPlayer\_Position

@fname varchar(30),

@lname varchar(30),

@CUID int,

@positionname varchar(25)

AS

Begin

declare @pid int, @poid int

insert into Player\_Name(fname,lname,CUID) values (@fname, @lname, @CUID);

set @pid=(select SCOPE\_IDENTITY());

insert into Position(positionname) values (@positionname);

set @poid=(select SCOPE\_IDENTITY());

insert into PlayerPosition(PID, POID) values (@pid, @poid);

END;

go

exec loadPlayer\_Position 'Ramu','Davis', 7, 'Midfield';

-- 3) A stored procedure that updates data in your database based on input.

drop procedure updatePlayerName;

go

create procedure updatePlayerName

@PID int,

@CUID int,

@Fname varchar(30),

@Lname varchar(30)

AS

BEGIN

update Player\_Name

set CUID = @CUID

where PID = @PID and Fname=@Fname and Lname=@Lname

END;

-- Obtain the Player Name and School of a Player who had any stats

drop procedure ANYStat;

go

create procedure ANYStat

@PID int,

@Fname varchar(30) output,

@Lname varchar(30) output,

@SchoolName varchar(50)output,

@statvalue int output,

@statname varchar(25) output

AS

BEGIN

select @Fname=Fname, @Lname=Lname, @SchoolName=SchoolName, @statvalue=statvalue, @statname=statname

from Player\_Name PN join PlayerWeekStats PWS on PN.PID=PWS.PID join University U on PN.CUID=U.CUID join Statistic S on PWS.SID=S.SID

where @PID=PN.PID

END;

go

declare @A\_Fname varchar(30), @A\_Lname varchar(30), @A\_SchoolName varchar(50), @A\_statvalue int, @A\_statname varchar(25);

exec AnyStat 2, @A\_Fname output, @A\_Lname output, @A\_SchoolName output, @A\_statvalue output, @A\_statname output;

select @A\_Fname, @A\_Lname, @A\_SchoolName, @A\_statvalue, @A\_statname;

go

**Analysis**

Since the original proposal, I had to make minimal changes. The first was the amount of players I would track using this database. My original plan was to gather all of the players in the AMCC and collect their stats within my database. However, I realized that this would be a daunting task and as a result, I minimized the numbers of players used to seven and the stats to a substancially smaller sample. This was done simply for timesake. Although, now I have created procedures to load data faster than before. The other change that was made was with the entities. In my original proposal, I only had five entities which were player names, school, week, position and stat. Since then, I have created two more entities which are join tables between two entities that have a many to many relationship.

I learned a lot from this project. I learned the entire concept of relational databases from the basic ER diagram to a complex sql query creating and populating a database. I also learned what goes into creating databases. One major thing that went into creating databases was time. Something as small as this was exceptionally time consuming and I can only imagine the work required to manage databases within large corporations. Another thing I learned from this project is patience. Throughout the creation of this project, I ran into a few small syntax errors in SQL management studio. These errors can become tedious to fix as the length of the query increases in size. This brings me to the next thing I learned which is precision. When dealing with databases, I learned quickly that everything has to be precise or else the proper results will not be produced. When dealing with larger database projects with more important information, no room for error will be allowed. With this being said, this project has taught me how to develop and effectively manage a database with proper precision and patience in the allotted time given.

**Appendix**

AMCC Conference. Men’s Soccer Page. 2013. 17 April 2013 <http://www.amcconf.org/msoc.htm>

NCCA Conference. Men’s Soccer Page. 2013. 17 April 2013 <http://www.ncaa.com/sports/soccer-men/d1>

**References**

“Oracle Database Admin Guide” docs.oracle.com/cd/B19306\_01/server.102/b14231/tables.htm

Last, Pratt. “A Guide to SQL 8e.” Boston: Course Technology, 2009. Print.

AMCC Conference. About Page. 2013. 17 April 2013 <http://www.amcconf.org/about.htm>