MARIST

A Marist Swimming & Diving Team Database Proposal

~~~~~~

By: Harris Nagle
~~~~~~

Table Of Contents

Executive Summary ~

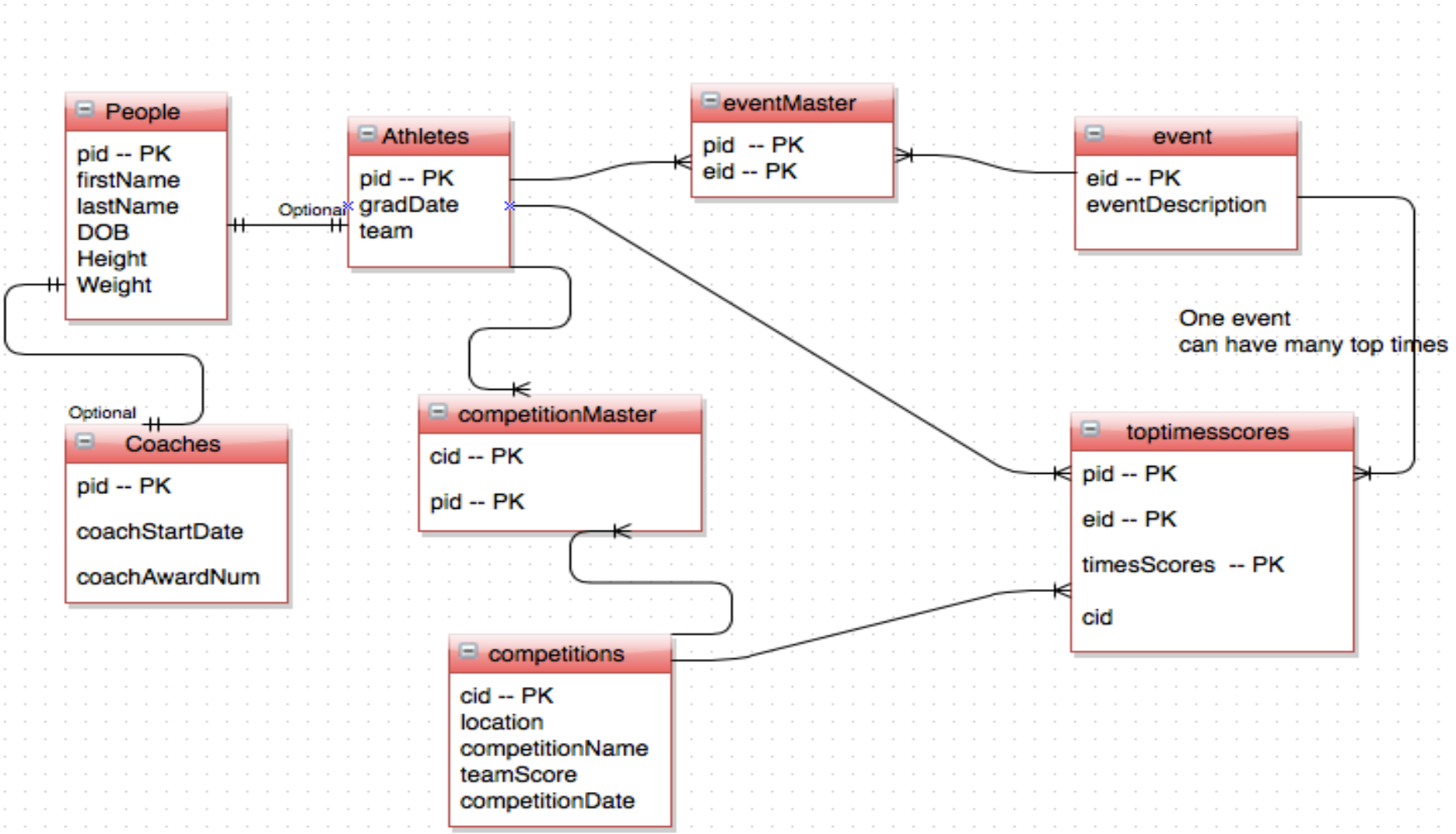
The Marist College Swimming & Dive Team is one of the most highly accomplished teams that Marist College has to offer. Collectively the men and women's program has acquired 28 of the schools 103 Metro Atlantic Athletic Conference championship titles, more than any other athletic program. The swim & dive team has massive amounts of data and records that must be kept stored in an orderly manner. This data also must be accessible so that the coaching staff may access it for creating competition entries without any inaccuracies within the data

The current issue with this process is that the swim & dive team uses a paper-pencil procedure and has a reliance on Red Fox Aquatic Club's Hy-Tek Sports Software. This software helps create paper print outs that the coaching staff utilizes when deciding entries. This makes the workload for swim coach VanWagner and diving coach Bolstad monumental, especially when selecting championship entries for all 45 athletes. The coaching staff needs a better way to gather, organize, and distribute its data in a more efficient manner.

The Marist Swim-NET is a system designed to organize times and scores for swimmers and divers. Overall this system was designed in order to ease the amount of work for the coaching staff, and allow the athletes themselves to see exactly where they stand on the team based on their score or time. This new system is designed to make the process of creating programs for competitions, and keeping tack of scores and times completely run/managed by the coaching staff. This system will help sophisticate and separate Marist College Swimming & Diving from any other team at Marist in terms of organization. This database was made on PostGres MySQL.

Entity Relationship Diagram ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

This ER-Diagram was made with Draw.io ***



People Table ~~~~~~

The People table is designed to include everyone on the roster and coaching staff of the Marist Swimming and diving team. It also contains the stats of all the people apart of the coaching staff and team. It shows their first name, last name, date of birth (D.O.B), sex, height, and weight. This is essential to have in the Marist Swim-NET so that the coaching staff can easily look up an athlete using the people ID (pid), and see their stats. This will make it more efficient in the recruiting process because the coach can quickly look up where a potential athlete would fit on the team.

FUNCTIONAL DEPENDENCIES ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

PRIMARY KEY → pid (peopleid)

CREATE STATEMENT~~~~~~

```
CREATE TABLE people
(
pid          varchar(3),     NOT NULL
firstname    varchar(255),   NOT NULL
lastname     varchar(255),   NOT NULL
dob          varchar(255),   NOT NULL
height       varchar(255),   NOT NULL
weight       varchar(255),   NOT NULL
primary key (pid),
)
```

SAMPLE DATA FOR PEOPLE TABLE ~~~~~~~~~~~

Data Output | Explain | Messages | History

	pid character(255)	firstname character(255)	lastname character(255)	dob character varying(255)	sex character(255)	height character(255)	weight character(255)
1	01	Jacob	Baker	Janurary 17th 1993	M	5-10	152
2	02	Caleb	Camp	April 11th 1994	M	6-2	176
3	03	Alexander	Cassel	August 3rd 1994	M	5-8	161
4	04	Dylan	Cummings	March 4th 1993	M	6-2	165
5	05	Matt	Darcy	August 25th 1995	M	5-11	161
6	06	Josh	Dodway	March 7th 1994	M	5-8	148
7	07	Billy	Drennan	Febuary 8th 1993	M	6-5	201
8	08	Peter	Gallino	July 22nd 1996	M	5-7	143
9	09	Spencer	Kelly	June 26th 1993	M	6-3	183
10	10	Brendan	McCarthy	May 3rd 1993	M	5-10	158
11	11	Ryan	Murphy	April 17th 1995	M	6-1	170
12	12	Harris	Nagle	April 5th 1995	M	5-6	155
13	13	Christopher	Penteck	May 7th 1995	M	6-0	170
14	14	Smauel	Presgraves	June 22nd 1996	M	5-6	160
15	15	Jason	Ruddy	January 23rd 1993	M	6-1	176
16	16	John	Spitzer	June 16th 1993	M	6-3	188
17	17	Sean	Sullivan	March 1st 1993	M	5-8	160
18	18	Nicholas	Vandermolen	April 10th 1993	M	6-2	188
19	19	Jackson	Wahl	August 30th 1995	M	5-11	170
20	20	Larry	VanWagner	April 18th 1956	M	5-10	178
21	21	Melanie	Bolstad	July 23rd 1957	F	5-5	134
22	22	Jim	Billesimo	May 6th 1964	M	5-9	164
23	23	Kyle	O Neil	June 9th 1970	M	6-1	205
24	24	Mike	Burud	May 20th 1992	M	5-11	171

Coaches Table

The Coaches table is designed to keep track of how long coaches have been coaching the marist swimming and diving team by storing their start date through the coachStartDate attribute. It also helps keep track of how many awards they have acquired throughtout their coaching career through the coachAwardNum. It is also useful for potential recruits to look up and see how much expeirience they have coaching the team, because with more expierience means more wisdom and that is attractive to potential new athletes.

FUNCTIONAL DEPENDENCIES

Primary Key → people id (pid)

CREATE STATEMENT

```
CREATE TABLE coaches
(
pid             varchar(255),   NOT NULL
coachAwardNum   varchar(255),
coachStartDate  varchar(255),
primary key (pid),
)
```

SAMPLE DATA FOR COACHES TABLE~~~~~~~~~~~~

Data Output | Explain | Messages | History

	pid character(255)	coachstartdate character(255)	coachawardnum character(255)
1	20	1976	153
2	21	1989	83
3	22	1990	43
4	23	1987	57
5	24	2014	2

Athlete Table

The Athlete table is designed to organize and separate all the athletes out of the people table and away from the coaches. They athletes are organized by their people id (pid), gradDate, and team. The gradDate attribute is important because it will help the coaching staff see what class in school they are in, meaning if the athlete is currently a freshman they will have 2018 next to their pid. The team attribute is important because coaches call up a call for all the swimmers on the team instead of all the swimmers and divers together in their result.

FUNCTIONAL DEPENDENCIES

Primary Key → people id (pid)

CREATE STATEMENT

```
CREATE TABLE athletes
(
pid             varchar(255),   NOT NULL
gradDate        varchar(255),
team            varchar(255),   NOT NULL
primary key (pid),
)
```

SAMPLE DATA FOR ATHLETES TABLE ~~~~~~~~~~

Data Output | Explain | Messages | History

	pid character(255)	graddate character(255)	team character(255)
1	01	2015	Diving
2	02	2016	Swimming
3	03	2017	Swimming
4	04	2015	Swimming
5	05	2017	Swimming
6	06	2016	Diving
7	07	2015	Swimming
8	08	2018	Diving
9	09	2015	Swimming
10	10	2015	Swimming
11	11	2017	Swimming
12	12	2017	Swimming
13	13	2017	Swimming
14	14	2018	Swimming
15	15	2015	Swimming
16	16	2015	Swimming
17	17	2015	Swimming
18	18	2015	Swimming
19	19	2018	Swimming

Event Master Table

The eventMaster table is designed to compile all the athletes based on their people.id and the events that they have been entered in throughout multiple competitions during the season. This table is useful to the coaching staff because it will help them figure out what athletes were in what events throughout the season. This feature comes in handy when they have to decide what events an athlete is going to do at one of our championship competitions.

FUNCTIONAL DEPENDENCIES

Primary Key → event.id (eid – comes from event table), people.id (pid)

CREATE STATEMENT

```
CREATE TABLE eventmaster
(
pid           varchar(255),   NOT NULL
eid           varchar(255),   NOT NULL
primary key (pid, eid),
)
```

SAMPLE DATA FOR EVENTSMASTER TABLE ~~~~~

Data Output | Explain | Messages

	pid character(255)	eid character(255)
1	17	015
2	17	008
3	17	013
4	10	003
5	10	004
6	10	011
7	01	017
8	01	016
9	06	017
10	06	016
11	19	003
12	19	011
13	19	010

Event Table ~~~~~~

The event table is designed to be the archive of event id's (eid) and people id's (pid) provided by the athletes' table. This table is helpful to the coaching staff because it contains all the individual events available to enter one of their athletes into. The table contains and event id for each event and an eventDescription to go with the eid which shows the title/name of the event.

FUNCTIONAL DEPENDENCIES ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Primary Key → event id (eid)

CREATE STATEMENT ~~~~~~

```
CREATE TABLE event
(
eid                 varchar(255),   NOT NULL
eventDescription    varchar(255),
primary key (eid)
)
```

SAMPLE DATA FOR EVENT TABLE ~~~~~~~~~~~~~~~~

Data Output | Explain | Messages | History

	eid character(255)	eventdescription character(255)
1	001	50 Free
2	002	100 FREE
3	003	200 Free
4	004	500 Free
5	005	1000 Free
6	006	1650 Free
7	007	100 Fly
8	008	200 Fly
9	009	50 Back
10	010	100 Back
11	011	200 Back
12	012	100 Breast
13	013	200 Breast
14	014	200 Individual Medley
15	015	400 Individual Medley
16	016	1 Meter Diving
17	017	3 Meter Diving

Competition Master Table ~~~~~~~~~~~~~~~~~~~~~~~~~

The competition master table is designed to keep hold of what athletes are going to what specific competitions they have already competed in. Competitions are organized through a competition id (cid), and they are linked with a people id (pid). This is useful for the coaching staff when they have to decide a championship roster, which is limited to 18. This will table will help them decide if an athlete has had enough racing this season to even be considered being entered into a championship competition and being one of the select 18.

FUNCTIONAL DEPENDENCIES ~~~~~~

Primary Key → competition id (cid), people id (pid)

CREATE STATEMENT ~~~~~~

```
CREATE TABLE competitionMaster
(
cid      varchar(255),    NOT NULL
pid      varchar(255),    NOT NULL
primary key (cid, pid),
)
```

SAMPLE DATA FOR COMPETITION MASTER TABLE ~~~~~~

Data Output | Explain | Messages

	cid character(255)	pid character(255)
1	A05	17
2	A05	10
3	A05	01
4	A05	06
5	A05	19
6	A05	12
7	A05	14
8	A02	17
9	A02	10
10	A02	01
11	A02	06
12	A02	19
13	A02	12
14	A02	14
15	H04	17
16	H04	10
17	H04	01
18	H04	06
19	H04	19
20	H04	12
21	H04	14
22	H01	17

Competitions Table ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

The Competition table was created to store the actual competitions and their data. The data that corresponds with a competition would its competition id (cid), location, competitionName, teamScore, and competitionDate. All these attributes clarify when the competition happened, where it happened, whether we won or lost, and who the competition was against. This is useful when the staff is preparing the schedule for next season and they would like to reflect back on the previous season so they can maybe pick the same weekends to have competitions.

FUNCTIONAL DEPENDENCIES ~~~~~~

Primary Key → competition id (cid)

CREATE STATEMENT ~~~~~~

```
CREATE TABLE competitions
(
cid             varchar(255),   NOT NULL
location        varchar(255),
competitionName varchar(255),
teamScore       varchar(255),
competitionDate varchar(255),
primary key (cid)
)
```

SAMPLE DATA FOR COMPETITIONS ~~~~~~~~~~~~

Data Output | Explain | Messages

	cid character(255)	pid character(255)
1	A05	17
2	A05	10
3	A05	01
4	A05	06
5	A05	19
6	A05	12
7	A05	14
8	A02	17
9	A02	10
10	A02	01
11	A02	06
12	A02	19
13	A02	12
14	A02	14
15	H04	17
16	H04	10
17	H04	01
18	H04	06
19	H04	19
20	H04	12
21	H04	14
22	H01	17

Toptimesscores Table ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

The toptimesscores table is designed to store all of the top times swum or dove by athletes at any competition through out the season. It is organized by people id(pid), event id(eid), timesScores, and competition id (cid). The main attribute in this table that makes it unique is the timesScores column. This specific column contains all the fastest times swum in and event during the season. This will be useful for the coaching staff because they can visually see where they have plenty of fast men in events compared to events where the team is weaker in and will need to recruit from.

FUNCTIONAL DEPENDENCIES ~~~~~~

Primary Key → people id (pid), event id (eid), timesScores

CREATE STATEMENT ~~~~~~

```
CREATE TABLE toptimesscores
(
pid          varchar(255),    NOT NULL
eid          varchar(255),    NOT NULL
timesScores varchar(255),    NOT NULL
cid          varchar(255),
primary key (pid, eid, timesScores)
)
```

SAMPLE DATA FOR TOPTIMESSCORES ~~~~~~~~~~

Data Output | Explain | Messages | History

	pid character(255)	eid character(255)	timesscores character(255)	cid character(255)
1	07	003	1:40:96	A05
2	12	003	1:42:80	A05
3	12	012	57:77	A06
4	14	012	59:42	A06
5	10	003	1:45:76	A06
6	10	004	4:36:21	A05
7	14	004	4:50:09	A02
8	19	011	1:51:88	A05
9	19	010	51:42	A06
10	07	004	4:32:11	A05
11	10	011	1:52:11	A06
12	10	010	51:90	A02
13	01	016	298.87	H02
14	01	017	312.56	H02
15	06	016	302.67	H04
16	06	017	324.32	A05

Views ~~~~~~

This view, once run, will display a specific athlete and their best times from any events they swim. This is useful for the coaches because they will be able to see where a specific athlete compares to incoming and current team members.

```
CREATE VIEW athleteTimes (pid, timesScores, eid) AS
        SELECT p.pid, t.timesScores, e.eventDescription
        FROM people p, toptimesscores t, event e
        WHERE p.pid = '06'
```

This specific version of the view calls upon the pid of ‘06’. When run it will return this…

	pid character(255)	eid character(255)	timesscores character(255)
1	06	016	302.67
2	06	017	324.32

Stored Procedures ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

The getAthletesInfo stored procedure is designed to call upon a specific athlete using their pid. This is useful when coaches are wanting to look up and athlete and don't want to write a full query. This stored procedure takes less time and less keystrokes for our coaching staff to do.

```
CREATE OR REPLACE FUNCTION getAthletesInfo (varchar (255), REFCURSOR) RETURNS refcursor AS
$$
DECLARE
        athlete varchar(255)    := $1;
        resultset       REFCURSOR       := $2;
BEGIN
        OPEN resultset for
                select pid, people.firstname, people.lastname, people.dob, people.height, people.weight
                from people
                where pid = athlete;
        RETURN resultset;
END;
$$
language plpgsql;

select getAthletesInfo ('05', 'results');
fetch all from results;
```

Data Output | Explain | Messages | History

	pid character(255)	firstname character(255)	lastname character(255)	dob character varying(255)	height character(255)	weight character(255)
1	05	Matt	Darcy	August 25th 1995	5-11	161

Reports

Competitions Attended by Athletes

This will display all of the competitions and which athletes have participated in them. It will do this because when I call upon the competition master table there are pid's and cid's where the cid's correspond with the competitions and they will match up with an athlete if they have participated in.

This is the general query to get all the information from the competitionmaster table.

```
SELECT * FROM competitionmaster
```

The results here show all of the competitions that any athlete attended during the season. Here we can see the for the MAAC Championships we had 7 athletes attend. We can see that because there are 7 A05's listed in the cid column.

Data Output | Explain | Message

	cid character(255)	pid character(255)
1	A05	17
2	A05	10
3	A05	01
4	A05	06
5	A05	19
6	A05	12
7	A05	14
8	A02	17
9	A02	10
10	A02	01
11	A02	06

Reports

Competitions Attended by Athletes

This is a more specific query than the previous one. This report will help coaches see exactly what competitions a specific athlete attended during the season.

```
SELECT * FROM competitionmaster
        WHERE cid ='A05'
        AND pid ='06'
```

This specific athlete ‘06’ is Josh Dodway a Diver on the team. And we are checking to see if he attend the MAAC championships in Buffalo because the cid =’A05’

Data Output | Explain | Messages

	cid character(255)	pid character(255)
1	A05	06

The results show that Josh Dodway did in fact attend the MAAC championships held in Buffalo NY. The coaching staff can use this query for when they need to look any competition information for any athlete on the team.

Security

There are two different types of users for this database as specified below by the following Grant commands

1. The admin who can change, update, and maintain the database through its life cycle.

```
CREATE ROLE ADMIN
GRANT SELECT, INSERT, UPDATE, ALTER
ON ALL TABLES IN SCHEMA PUBLIC
TO ADMIN
```

2. The public user who can see and utilize the database by performing queries on it to get their desired information.

```
CREATE ROLE public
GRANT SELECT
ON ALL TABLES IN SCHEMA PUBLIC
TO public
```

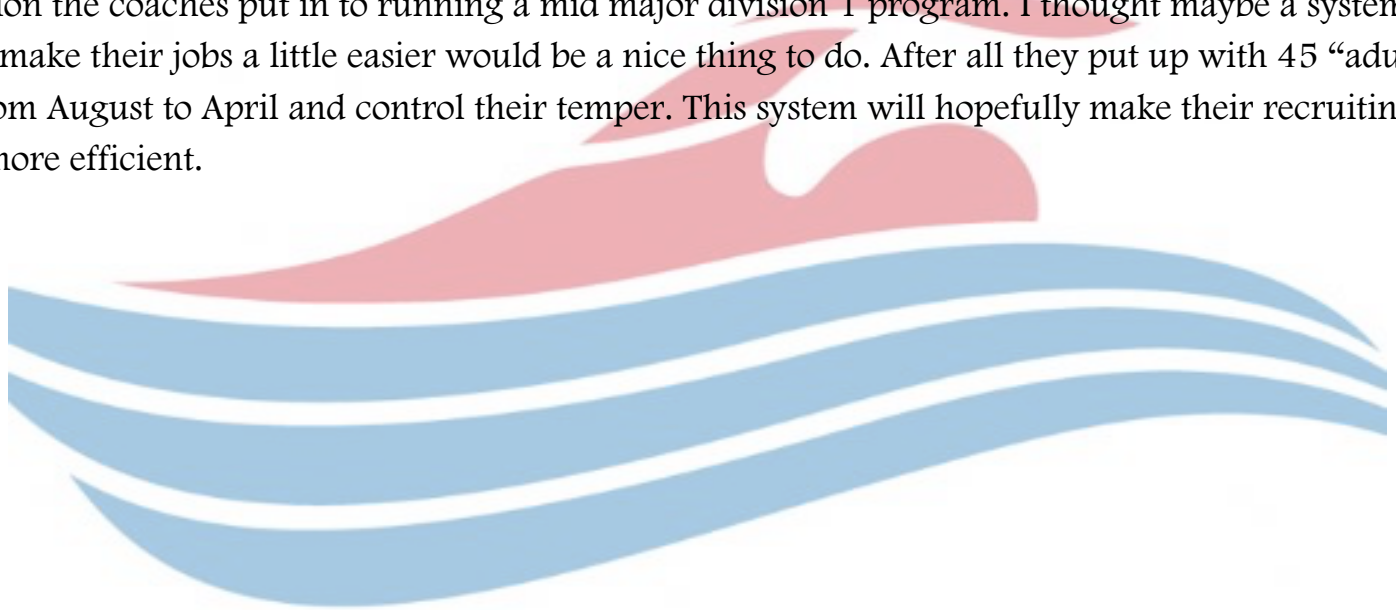
Future Enhancements~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

One of the possible future enhancements would be to include the women's entire roster. The women's swimming and diving team is a very big part of our team and we are proud to swim along side them in practice. Having the women's team included in the Marist Swim~NET would make it much more useful to the coaching staff. If they were added to the database if would make the database more useful for recruiting not only men but also other potential women athletes. This will maximize the coach's time when they are looking up where we have depth deficiencies and need to recruit new potential athletes. This will make our recruiting procedure become faster and more efficient making us able to reach out to faster prospects. This will ultimately make the Marist Swimming & Diving Team become even more competitive in the MAAC conference and an overall faster team.

Another future enhancements would be the possibility of creating a login, username, and password table. That way athletes can look at their own best times in all of their events. Having an account would help personalize the database to its people who

Implementation Notes~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

The implementation process for the Marist Swim-NET was very complex considering I was foolish and didn't save my .sql file in more than one location. Other than that, I had a great time looking up people's times and where they performed at their best. Swimming is a passion of mine and I really appreciate the hard work and dedication the coaches put in to running a mid major division 1 program. I thought maybe a system designed to make their jobs a little easier would be a nice thing to do. After all they put up with 45 "adults" everyday from August to April and control their temper. This system will hopefully make their recruiting process easier and more efficient.



Known Problems ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

A known problem with this database is that Initially I intended it to be used by the coaches and athletes. I did not address the athletes being able to use this system. They would need a username and password to access the database and its information. Another known problem is that the coach wont be able to find a list of specific athletes and their best times in all the events they've been entered in. This would be a key feature for coaches so that they could see new events where they could enter an athlete into; shake up the original line up so to speak.