Database Specifications for:

Marcus Anton Zimmermann Memorial Geological History Museum Collective

"He always loved rocks, probably…*crying sounds*"

-Marcus' grieving mother

December 8th, 2016

Jeffrey Lupia

Table of Contents

1 .....Title Page
2 .....Table of Contents
3-4.....Executive Summary
3-4.....*Overview*
4.....*Objectives*
5.....ER Diagram
6-7.....People
8.....TeamMembers
9.....Teams
10.....Museums
11.....DateSpecs
12.....Layers
13.....Fossils
14.....Views
15-16.....Roles
16.....Known Problems
17.....Future Enhancements

Executive Summary

Overview:

The *Marcus Anton Zimmerman Memorial Geological History Museum Collective* is a non-profit organization created in memory of Marcus Anton Zimmerman, who, 25 years ago, died tragically after forgetting how to breathe. After his death, his family wished to honor his memory in a way that Marcus would have loved. Sadly, Marcus didn't actually have any hobbies, or aspirations in life, so his parents decided to set up the *Marcus Anton Zimmerman Memorial Geological History Museum Collective* in the off-chance that Marcus secretly liked rocks. Over the years this Organization has grown to a national level, with 12 different museums across the United States.

In honor of the 25th anniversary of his untimely demise, the Organization has planned an expedition to the *Hell's Creek* rock formation in Montana with the purpose of retrieving fossils to create a new paleontological exhibit in each of the museums. A large area of land in this formation has been donated to the museum. The organization has commissioned each of the museums to

hire a team of five professionals, specializing in the fields of Paleontology and/or Geology to facilitate this dig. The organization has asked that these expert have at least a Master's degree in their respective subjects. Additionally, each team is to have one supervising member to facilitate the dig, and the remainder of the team is to be a mix of both paleontologists, and geologist, in order to identify fossils while on the dig-site, and to identify the different layers of rock. This identification will be used to approximate the age of the fossils.

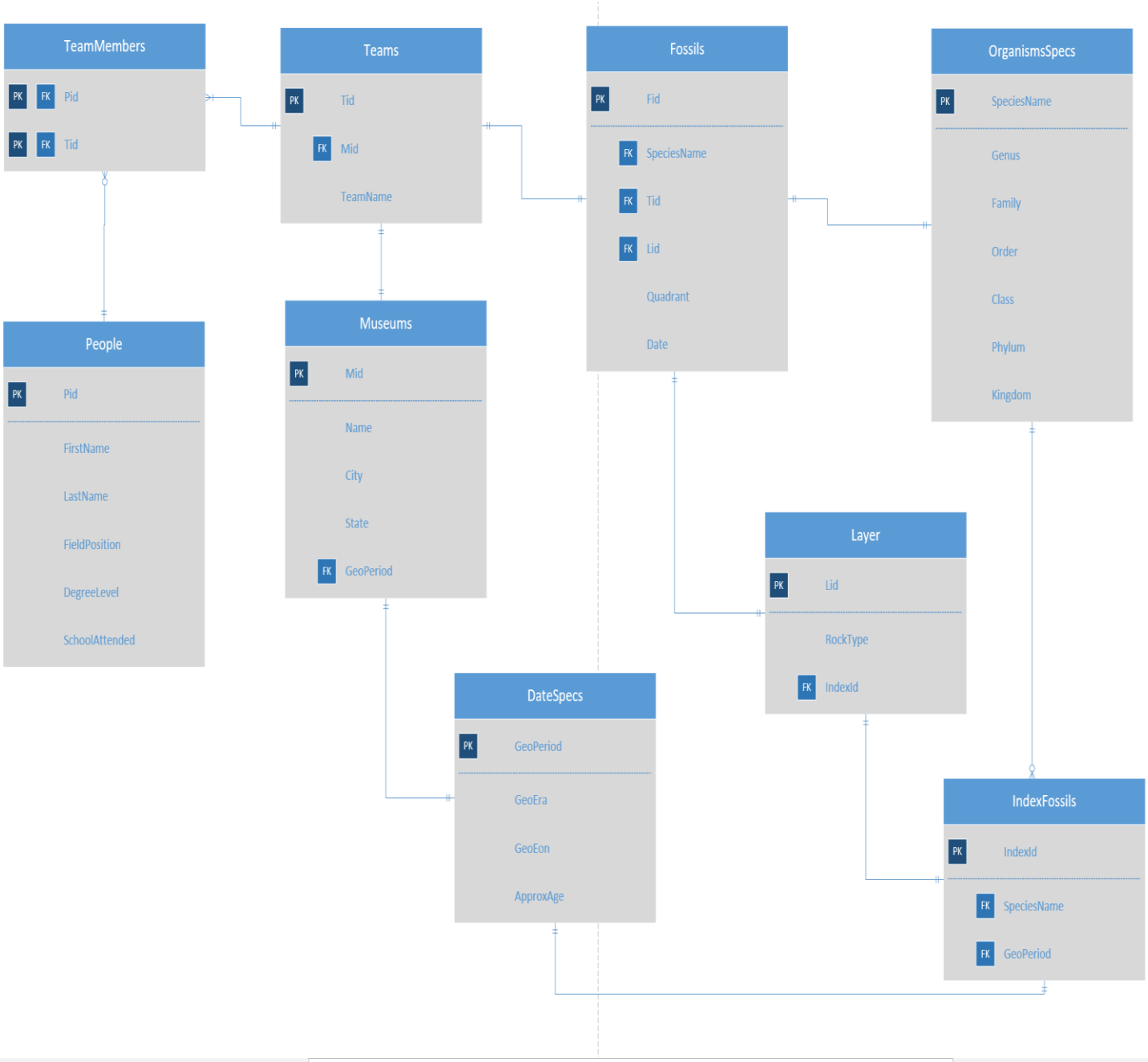
Objectives:

In order to properly document, identify, and catalogue the different fossils produced by the teams, a database has been commissioned to log information throughout the expedition. This database will keep track of each fossil excavated, and log various characteristics about them, including the species and where and when each fossil is found. Additionally, the database will help in the identification of the age of each fossil as well as including the various geologic ages associated with each fossil. Information about the members of each team, and individual museums will also be available.

Entity Relationship Diagram

Tables

People



The purpose of the People table is to store data on the individuals participating in this dig. Among this data is the first and last name of the individual, their position in the dig (Supervisor, Geologist, or Paleontologist), the school they attended, and the degree they earned. There is a check constraint on the degree, as the Organization specified that only those with a Masters degree or higher in their respective fields would be able to participate.

Create Statement:

```
CREATE TABLE People (
Pid SERIAL NOT NULL,
FirstName text NOT NULL,
LastName text NOT NULL,
FieldPosition text NOT NULL CHECK (FieldPosition='Supervisor' OR
FieldPosition='Paleontologist' OR FieldPosition='Geologist'),
DegreeLevel text NOT NULL CHECK (DegreeLevel='Masters' OR
DegreeLevel='PHD'),
SchoolAttended text NOT NULL,
PRIMARY KEY(Pid)
);
```

Functional Dependencies:

Pid>>>FirstName, LastName, FieldPosition, DegreeLevel, SchoolAttended

Sample Data:

	pid integer	firstname text	lastname text	fieldposition text	degreelevel text	schoolattended text
1	1	John	Smith	Supervisor	PHD	Marist College
2	2	Alan	Labouseur	Supervisor	PHD	Marist College
3	3	Marcus	Zimmermann	Paleontologist	Masters	Harvard University
4	4	G	Leaden	Geologist	PHD	University of Pheonix
5	5	Troy	Demers	Supervisor	Masters	Dartmouth University
6	6	William	Wise	Supervisor	PHD	Berkley College
7	7	Kevin	Kleinschmidt	Paleontologist	Masters	Culinary Institute of America
8	8	James	Bond	Paleontologist	PHD	Classified University of London
9	9	Sean	Connery	Supervisor	Masters	Prinston University
10	10	Daniel	Craig	Geologist	Masters	Dutchess County Community College
11	11	Ian	Sniffen	Supervisor	PHD	Boston College
12	12	Brandon	Litwin	Geologist	Masters	Clown College
13	13	Maria	Valenti	Paleontologist	PHD	Boston University
14	14	Danielle	Markus	Supervisor	PHD	Tufts University
15	15	Taylor	Albert	Supervisor	PHD	Brown University
16	16	Olivia	Milano	Paleontologist	Masters	Johns Hopkins University
17	17	Doctor	Pepper	Supervisor	PHD	Bard College
18	18	Elizabeth	Garrison	Geologist	Masters	Vassar College
19	19	Jeffrey	Lupia	Supervisor	PHD	University of Pennsylvania
20	20	Margaret	Caulfield	Supervisor	PHD	Adelphi University
21	21	Ariana	Singer	Supervisor	PHD	New York University
22	22	Frankie	Amodio	Geologist	Masters	Rutgers University
23	23	Jamie	Cross	Paleontologist	Masters	University of Arizona
24	24	Harry	Potter	Geologist	Masters	Hogwarts School of Witchcraft and Geolo

TeamMembers

The purpose of the TeamMembers table is to act as an intermediate table between teams, and people. This table shows what people are on what team.

Create Statement:

```
CREATE TABLE TeamMembers(
Pid SERIAL NOT NULL REFERENCES People(Pid),
Tid SERIAL NOT NULL REFERENCES Teams(Tid),
PRIMARY KEY(Pid, Tid)
);
```

Functional Dependencies:

Pid>>>Tid
Tid>>>Pid

Sample Data:

	pid integer	tid integer
1	1	1
2	2	2
3	5	3
4	6	4
5	9	5
6	11	6
7	14	7
8	15	8
9	17	9
10	19	10
11	20	11
12	21	12
13	3	1
14	4	2
15	7	3
16	8	4
17	10	5
18	12	6
19	13	7
20	16	8
21	18	9
22	22	10
23	23	11
24	24	12

Teams

The Teams table provides a unique team id for each team. These teams each come from an individual museum, so the table is linked to the museum table through *Mid*. The Team table also has a team name field, which identifies each team by a specific name. The actual names of the teams correspond with the Geological period of the exhibit their museum is building.

Create Statement:

```
CREATE TABLE Teams(
Tid SERIAL NOT NULL,
Mid SERIAL NOT NULL REFERENCES Museums(Mid),
TeamName text NOT NULL,
PRIMARY KEY(Tid)
);
```

Functional Dependencies:

Tid>>>Mid, TeamName

Sample Data:

	tid integer	mid integer	teamname text
1	1	1	Team Quaternary
2	2	2	Team Neogene
3	3	3	Team Paleogene
4	4	4	Team Cretaceous
5	5	5	Team Jurassic
6	6	6	Team Triassic
7	7	7	Team Permian
8	8	8	Team Carboniferous
9	9	9	Team Devonian
10	10	10	Team Silurian
11	11	11	Team Ordovician
12	12	12	Team Cambrian

Museums

The Museums table provides information about each museum. This includes the unique identifier (Mid), as well as the city and state in which each is located. This table also includes the name of the museum, and the Geological Period for which it is creating an exhibit. This period is related to the GeoPeriod key of the

Create Statement:

```
CREATE TABLE Museums(
Mid SERIAL NOT NULL,
Name text NOT NULL,
City text NOT NULL,
State text NOT NULL,
GeoPeriod text NOT NULL REFERENCES DateSpecs(GeoPeriod),
PRIMARY KEY(Mid)
);
```

Functional Dependencies:

Mid>>>Name, City, State, GeoPeriod

Sample Data:

	mid integer	name text	city text	state text	geoperiod text
1	1	Marcus Rock Museum	Poughkeepsie	New York	Quaternary
2	2	Marcus Shiny Rock Museum	Carbondale	Pennsylvania	Neogene
3	3	Marcus Fossil Museum	San Diego	California	Paleogene
4	4	Antons Museum of Rocks	Chicago	Illinois	Cretaceous
5	5	Antons Museum of Shiny Rocks	Greensboro	North Carolina	Jurassic
6	6	Antons Rock Museum	Warwick	Rhode Island	Triassic
7	7	Marcus Museum of Rocks	Danbury	Connecticut	Permian
8	8	Marcus Museum of Shiny Rocks	Salem	Oregon	Carboniferous
9	9	Antons Shiny Rock Museum	Kansas City	Kansas	Devonian
10	10	Antons Fossil Museum	Dallas	Texas	Silurian
11	11	Antons Museum of Fossils	Pheonix	Arizona	Ordovician
12	12	Marcus Museum of Fossils	Colorado City	Colorado	Cambrian

DateSpecs

The DateSpecs table provides information about different geological periods. The primary key GeoPeriod is comprised of the 12 most recent geological periods. Also included are the Geological era's and eon, as well as an approximate date range regarding the geological periods.

Create Statement:

```
CREATE TABLE DateSpecs(
GeoPeriod text NOT NULL,
GeoEra text NOT NULL,
GeoEon text NOT NULL,
ApproxAge int4range NOT NULL,
PRIMARY KEY(GeoPeriod)
);
```

Functional Dependencies:

GeoPeriod >>> GeoEra, GeoEon, ApproxAge

Sample Data:

	geoperiod text	geoera text	geoeon text	approxage int4range
1	Quaternary	Cenozoic	Phanerozoic	[0,1500001)
2	Neogene	Cenozoic	Phanerozoic	[1500001,23000001)
3	Paleogene	Cenozoic	Phanerozoic	[23000001,66000001)
4	Cretaceous	Mesozoic	Phanerozoic	[66000001,145000001)
5	Jurassic	Mesozoic	Phanerozoic	[145000001,201000001)
6	Triassic	Mesozoic	Phanerozoic	[201000001,252000001)
7	Permian	Paleozoic	Phanerozoic	[252000001,299000001)
8	Carboniferous	Paleozoic	Phanerozoic	[299000001,359000001)
9	Devonian	Paleozoic	Phanerozoic	[359000001,419000001)
10	Silurian	Paleozoic	Phanerozoic	[419000001,444000001)
11	Ordovician	Paleozoic	Phanerozoic	[444000001,485000001)
12	Cambrian	Paleozoic	Phanerozoic	[485000001,541000001)

Layers

The Layers table provides information about the different layers being excavated at the dig-site. The further down the layers go, the kind of rock changes, and so does the age of the rock, and subsequent fossils found. For the purposes of this dig, 12 layers are identified by the primary key Lid, and each layer correlates to a different geological period referenced by GeoPeriod.

Create Statement:

```
CREATE TABLE Layers(
Lid SERIAL NOT NULL,
GeoPeriod text NOT NULL REFERENCES DateSpecs(GeoPeriod),
RockType text NOT NULL,
PRIMARY KEY(Lid)
);
```

Functional Dependencies:

Lid>>> GeoPeriod, RockType

Sample Data:

	lid integer	geoperiod text	rocktype text
1	1	Quaternary	Sandstone
2	2	Neogene	Limestone
3	3	Paleogene	Shale
4	4	Cretaceous	Sandstone
5	5	Jurassic	Dolomite
6	6	Triassic	Limestone
7	7	Permian	Dolomite
8	8	Carboniferous	Shale
9	9	Devonian	Sandstone
10	10	Silurian	Shale
11	11	Ordovician	Limestone
12	12	Cambrian	Dolomite

Fossils

The Fossils table records information about each fossil excavated by the team. Data for each fossil is meant to be inputted by the experts working the dig. Fid is used as a unique identifier for each fossil found. The species, of the fossil, the finder, when it was found, and the layer and which of the 16 areas or *quadrants* it was discovered in are also recorded.

Create Statement:

```
CREATE TABLE Fossils(
Fid SERIAL NOT NULL,
SpeciesName text NOT NULL,
Tid SERIAL NOT NULL REFERENCES Teams(Tid),
Lid SERIAL NOT NULL REFERENCES Layers(Lid),
Quadrant INT NOT NULL,
DateFound date NOT NULL,
PRIMARY KEY(Fid)
);
```

Functional Dependencies:

Fid, >>> SpeciesName, Tid, Lid, Quadrant, DateFound

Sample Data:

	fid integer	speciesname text	tid integer	lid integer	quadrant integer	datefound date
18	18	Neptunea Tabulata	1	1	5	2016-12-09
19	19	Neptunea Tabulata	7	1	14	2016-12-11
20	20	Neptunea Tabulata	11	1	12	2016-12-15
21	21	Neptunea Tabulata	1	1	1	2016-12-18
22	22	Neptunea Tabulata	5	1	7	2016-12-19
23	23	Neptunea Tabulata	5	1	9	2016-12-16
24	24	Calyptraphorus Velatus	3	2	9	2016-12-18
25	25	Venericardia Planicosta	10	3	9	2016-12-13
26	26	Scaphites Hippocrepis	12	4	9	2016-12-09
27	27	Nerinea Trindosa	2	5	9	2016-12-10
28	28	Trophites Subbullatus	8	6	9	2016-12-11
29	29	Trophites Subbullatus	5	6	9	2016-12-18
30	30	Leptodus Americanus	6	7	9	2016-12-22
31	31	Dictyoclostus Americanus	1	8	9	2016-12-20
32	32	Mucrospirifer Mucronatus	4	9	9	2016-12-19
33	33	Hexamoceras Hertzeri	11	10	9	2016-12-09
34	34	Bathyurus Extans	7	11	9	2016-12-06
35	35	Paradoxides Pinus	9	12	9	2016-12-07

Views

Team1Fossils:

This view shows which fossils were found and catalogued by team one, and includes, the species, and the date found.

Create Statement:

```
Create View Team1Fossils AS

SELECT Fid, SpeciesName, DateFound

FROM Fossils INNER JOIN Teams ON
(Fossils.Tid=Teams.Tid)

WHERE Teams.Tid = 1;
```

Roles

The Current roles for this database are:

Admin

TeamMember

CREATE ROLE Admin;

CREATE ROLE TeamMember;

A TeamMember has access to the Fossils Table

GRANT INSERT, SELECT, DELETE, UPDATE ON Fossils TO TeamMember;

An Admin has access to all privileges of the database

GRANT INSERT, SELECT, DELETE, UPDATE ON Fossils TO Admin;

GRANT INSERT, SELECT, DELETE, UPDATE ON People TO Admin;

GRANT INSERT, SELECT, DELETE, UPDATE ON Teams TO Admin;

GRANT INSERT, SELECT, DELETE, UPDATE ON Layers TO Admin;

GRANT INSERT, SELECT, DELETE, UPDATE ON TeamMembers TO Admin;

GRANT INSERT, SELECT, DELETE, UPDATE ON Museums TO Admin;

GRANT INSERT, SELECT, DELETE, UPDATE ON DateSpecs TO Admin;

Known Problems

-One problem is that the artificial field *Mid* is used as the Primary Key for the Museum table, however, the database could be changed to use the name of the museum, as that would be a more proper way to designate the table.

-Additionally, the Layer ID is determined based on how many layers it is from the surface however, sometimes tectonic activities can shift the rock and make this an inaccurate way to judge the date. The database does not yet have provisions to deal with that.

Future Enhancements

-In addition to creating provisions in order to fix the known problems, I would have liked to have included a taxonomic table regarding the organisms(from the fossils) such a table would include the genus, species, class, phylum, ect.

-I would also create a table for Index fossils(Specific Fossils researchers use to approximate the age of fossils around them) and use it in collaboration with the dateSpecs table to determine age.