Database Management Design Project

-- Marist Housing Move out

12/01/2012

Siting Wang

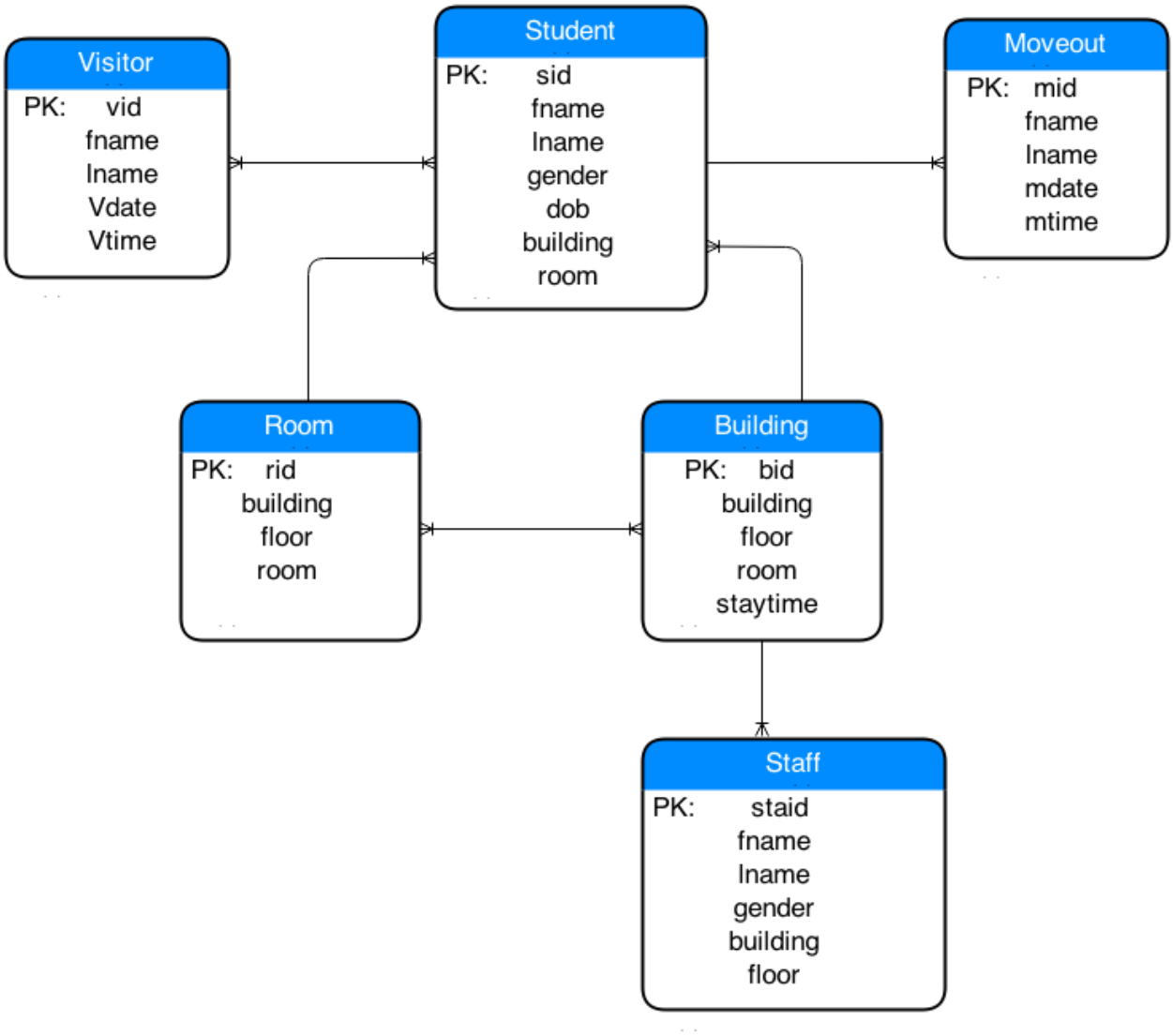
Table of Contents

Executive Summary

Housing selection happens every academic year. At the end of every school year, Marist resident students are require to move out of the building with their personal belongings, and will replace with new resident areas in the following year. When the moving out day approached, it brings a lots of work for housing staff and security officers. Housing staff need to keep track on the date and time that students are leaving; security officers need to get the information of the non-Marist students and faculty who are entering the building. With all the considering, creating a database with all the information will be easy for them, and it might reduce their quantities of work.

This database is designed to keep track of the people who are enter the building when they swipe their ID card, and for those who need to sign in with security people in order to get in the building. This database is also designed to keep track the date that student move out, so the housing staff can check with students when they leave.

Entity Relationship Diagram



Student Table

Description:

This table is needed to keep track the information of students who are living in the building and planning to move out the building at the end of school year. Table student, building, room, staff are benefit for housing people, and table visitor is benefit for security officers to track on people who are enter the building.

Create Statement:

--student

```
--student
        Drop table if exists Student;
        create table Student
        (
         sid char(20) primary key,
         fname char(20),
         lname text,
         gender text,
         dob date,
         building text not null,
         room char(20)
         );
```

Insert Statements:

```
--insert into student
insert into student(sid,fname,lname,gender,dob,building,room)
        values('20048890','kevel','Kim','male','02-04-1995','Leo','103');
insert into Student(Sid,fname,lname,gender,dob,building,room)
        values('20024482','kan','Smith','male','12-04-1995','Marian','304');
insert into student(sid,fname,lname,gender,dob,building,room)
        values('20045650','Ada','molon','female','07-08-1994','Sheahan','209');
insert into Student(Sid,fname,lname,gender,dob,building,room)
        values('20048824','Kyle','Bend','male','06-04-1993','Champagnat','604');
insert into student(sid,fname,lname,gender,dob,building,room)
        values('20048678','Susan','Lee','female','11-25-1994','Leo','509');
insert into Student(Sid,fname,lname,gender,dob,building,room)
        values('20040067','Caroline','Sean','female','04-15-1994','Marian','219');
insert into student(sid,fname,lname,gender,dob,building,room)
        values('20048867','Kaitlyn','Fold','female','10-10-1992','Sheahan','205');
insert into Student(Sid,fname,lname,gender,dob,building,room)
        values('20024483','Kayla','Bean','female','03-19-1995','Champagnat','903');
```

Sample Data:

Data Output | Explain | Messages | History

	sid character(20)	fname character(20)	lname text	gender text	birth date	building text	room character(20)
1	20048890	kevel	Kim	male	1995-02-04	Leo	103
2	20024482	kan	Smith	male	1995-12-04	Marian	304
3	20045650	Ada	molon	female	1994-07-08	Sheahan	209
4	20048824	Kyle	Bend	male	1993-06-04	Champganat	604
5	20048678	Susan	Lee	female	1994-11-25	Leo	509
6	20040067	Caroline	Sean	female	1994-04-15	Marian	219
7	20048867	Kaitlyn	Fold	female	1992-10-10	Sheahan	205
8	20024483	Kayla	Bean	female	1995-03-19	Champganat	903

Functional Dependencies:

Sid->fname, lname, gender, birth, building, room

Building Table

Description:

This building table consists of the freshmen buildings, and lists of building, floor, room, and student's stay time.

Create Statement:

```
--building
Drop table if exists Building;
create table Building
(
bid char(3) primary key,
building text not null,
floor char(20),
room char(20),
staytime date
```

Insert Statements:

```
--insert into building
insert into Building(bid,building,floor,room,staytime)
        values('b01','Leo','3','305','12-15-2014');
insert into Building(bid,building,floor,room,staytime)
        values('b02','Sheahan','2','215','12-19-2014');
insert into Building(bid,building,floor,room,staytime)
        values('b03','Leo','2','215','12-17-2014');
insert into Building(bid,building,floor,room,staytime)
        values('b04','Marian','2','204','12-18-2014');
insert into Building(bid,building,floor,room,staytime)
        values('b05','Champagnat','7','705','12-19-2014');
insert into Building(bid,building,floor,room,staytime)
        values('b06','Sheahan','3','311','12-16-2014');
```

Sample Data:

Output pane

Data Output | Explain | Messages | History

	bid character(3)	building text	floor character(20)	room character(20)	staytime date
1	b01	Leo	3	305	2014-12-15
2	b02	Sheahan	2	215	2014-12-19
3	b03	Leo	2	215	2014-12-17
4	b04	Marian	2	204	2014-12-18
5	b05	Champagnat	7	705	2014-12-19
6	b06	Sheahan	3	311	2014-12-16

Functional Dependencies:

bid-> building, floor, room, staytime

Room Table

Description:

This table is needed to keep track the information of students who are living in the building and planning to move out the building.

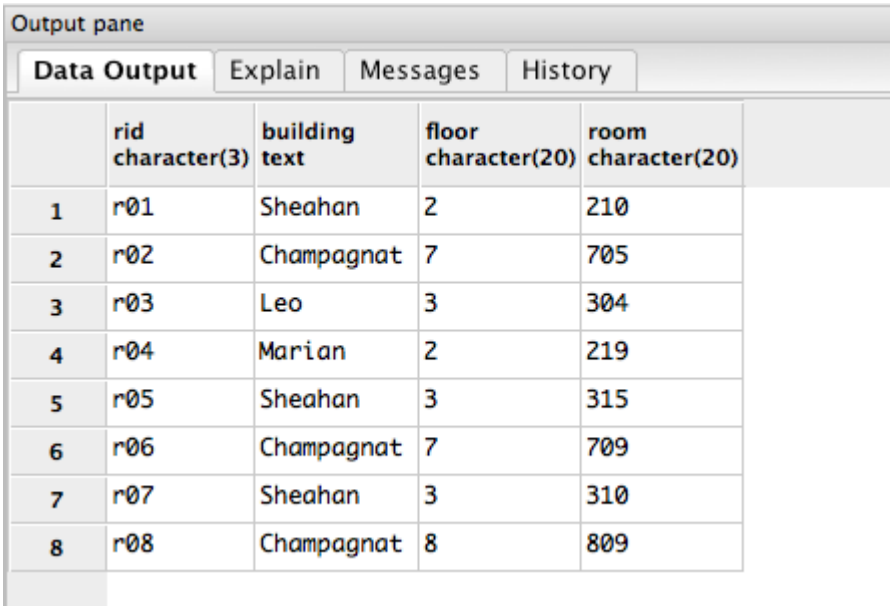
Create Statement:

```
--Room
drop table if exists Room;
create table Room
(
rid char(3) primary key,
building text not null,
floor char(20),
room char(20)
);
```

Insert Statements:

```
--insert into room
insert into Room(rid,building,floor,room)
        values('r01','Sheahan','2','210');
insert into Room(rid,building,floor,room)
        values('r02','Champagnat','7','705');
insert into Room(rid,building,floor,room)
        values('r03','Leo','3','304');
insert into Room(rid,building,floor,room)
        values('r04','Marian','2','219');
insert into Room(rid,building,floor,room)
        values('r05','Sheahan','3','315');
insert into Room(rid,building,floor,room)
        values('r06','Champagnat','7','709');
insert into Room(rid,building,floor,room)
        values('r07','Sheahan','3','310');
insert into Room(rid,building,floor,room)
        values('r08','Champagnat','8','809');
```

Sample Data:


Output pane

Data Output | Explain | Messages | History

	rid character(3)	building text	floor character(20)	room character(20)
1	r01	Sheahan	2	210
2	r02	Champagnat	7	705
3	r03	Leo	3	304
4	r04	Marian	2	219
5	r05	Sheahan	3	315
6	r06	Champagnat	7	709
7	r07	Sheahan	3	310
8	r08	Champagnat	8	809

Functional Dependency:

rid->building,floor,room

Visitor Table

Description:

This visitor table lists visitor's first name, last name, visit date, visit time. With more information of visitors, it will be easy for security officers to keep track on.

Create Statement:

```
--visitor
drop table if exists Visitor;
create table Visitor
(
vid char(20) primary key,
fname text,
lname text,
Vdate date not null,
Vtime time not null
);
```

Insert Statements:

```
--insert into visitor
insert into Visitor(vid,fname,lname,vdate,vtime)
        values('20047387','Dan','Mathew','10-15-2014','18:30');
insert into Visitor(vid,fname,lname,vdate,vtime)
        values('20037893','Kenny','Ocean','11-11-2014','10:25');
insert into Visitor(vid,fname,lname,vdate,vtime)
        values('20043453','John','Miller','10-23-2014','13:30');
insert into Visitor(vid,fname,lname,vdate,vtime)
        values('20046535','Dannel','Mcbrain','11-12-2014','11:35');
insert into Visitor(vid,fname,lname,vdate,vtime)
        values('20045643','Bryn','kushi','9-12-2014','9:30');
insert into Visitor(vid,fname,lname,vdate,vtime)
        values('20042465','Sally','Ross','11-11-2014','14:25');
insert into Visitor(vid,fname,lname,vdate,vtime)
        values('20043465','Roza','Mura','10-05-2014','20:30');
insert into Visitor(vid,fname,lname,vdate,vtime)
        values('20049323','William','Curry','12-01-2014','21:18');
```

Sample Data:

Output pane

Data Output | Explain | Messages | History

	vid character(20)	fname text	lname text	vdate date	vtime time without time zone
1	20047387	Dan	Mathew	2014-10-15	18:30:00
2	20037893	Kenny	Ocean	2014-11-11	10:25:00
3	20043453	John	Miller	2014-10-23	13:30:00
4	20046535	Dannel	Mcbrain	2014-11-12	11:35:00
5	20045643	Bryn	kushi	2014-09-12	09:30:00
6	20042465	Sally	Ross	2014-11-11	14:25:00
7	20043465	Roza	Mura	2014-10-05	20:30:00
8	20049323	William	Curry	2014-12-01	21:18:00

Functional Dependency:

vid->fname,lname,vdate,vtime

Staff Table

Description:

This table lists the staff's first name, last name, gender, building that they in charge, and floor they live.

Create Statement:

```
--Resident staff
drop table if exists Staff;
create table Staff
(
 staid char(20) primary key,
 fname char(20),
 lname text,
 gender text,
 building text not null,
 floor char(20)
 );
```

Insert Statements:

```
--inset into staff
insert into Staff(staid,fname,lname,gender,building,floor)
        Values('20048890','kelly','McDough','female','Marian','2');
insert into Staff(staid,fname,lname,gender,building,floor)
        Values('20034532','Paggy','Smith','female','Champanat','5');
insert into Staff(staid,fname,lname,gender,building,floor)
        Values('20040053','Wendy','Sung','female','Marian','1');
insert into Staff(staid,fname,lname,gender,building,floor)
        Values('20022244','Warren','Park','male','Champanat','4');
insert into Staff(staid,fname,lname,gender,building,floor)
        Values('20048888','kimi','Yuk','male','Marian','1');
insert into Staff(staid,fname,lname,gender,building,floor)
        Values('20024234','Lucas','Zain','male','Leo','1');
```

Sample Date:

Output pane

Data Output | Explain | Messages | History

	staid character(20)	fname character(20)	lname text	gender text	building text	floor character(20)
1	20048890	kelly	McDough	female	Marian	2
2	20034532	Paggy	Smith	female	Champanat	5
3	20040053	Wendy	Sung	female	Marian	1
4	20022244	Warren	Park	male	Champanat	4
5	20048888	kimi	Yuk	male	Marian	1
6	20024234	Lucas	Zain	male	Leo	1

Functional Dependency:

Staid->fname,lname,gender,building,floor

Moveout Table

Description:

This table lists the first name, last name, moveout date, and move time of students, so staff can check with students.

Create Statement:

```
--table moveout
drop table if exists Moveout;
create table Moveout
(
mid char(20) not null primary key,
fname text,
lname text,
mdate date not null,
mtime time not null
);
```

Insert Statements:

```
--insert into moveout
insert into Moveout(mid,fname,lname,mdate,mtime)
        values('m01','Cara','Leon','12-17-2014','15:30');
insert into Moveout(mid,fname,lname,mdate,mtime)
        values('m02','Lesly','Garcia','12-18-2014','16:10');
insert into Moveout(mid,fname,lname,mdate,mtime)
        values('m03','Darriel','NG','12-19-2014','17:20');
insert into Moveout(mid,fname,lname,mdate,mtime)
        values('m04','Brandon','Colon','12-17-2014','13:40');
insert into Moveout(mid,fname,lname,mdate,mtime)
        values('m05','Helen','Wood','12-19-2014','14:30');
insert into Moveout(mid,fname,lname,mdate,mtime)
        values('m06','Anna','Kuan','12-18-2014','13:40');
```

Sample Date:

Output pane

Data Output | Explain | Messages | History

	mid character(20)	fname text	lname text	mdate date	mtime time without time zone
1	m01	Cara	Leon	2014-12-17	15:30:00
2	m02	Lesly	Garcia	2014-12-18	16:10:00
3	m03	Darriel	NG	2014-12-19	17:20:00
4	m04	Brandon	Colon	2014-12-17	13:40:00
5	m05	Helen	Wood	2014-12-19	14:30:00
6	m06	Anna	Kuan	2014-12-18	13:40:00

Functional Dependency:

mid->fname,lname,mdate,mtime

Views

Description:

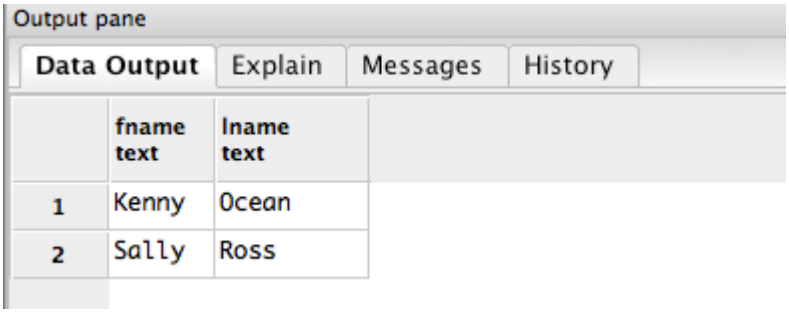
This view displays anyone who visit the building at 11-11-2014.

Create Statement:

```
drop view if exists vdate;
create view vdate
AS
        select distinct fname,lname
        from Visitor
        where vdate='11-11-2014';

         select *
         from vdate;
```

Sample data:



Reports and Queries:

This query gets the fname and lname of student who live in the Sheahan, and they are both female.

Create Statement:

```
--query
select fname,lname,room
from Student
where building='Sheahan' and gender='female';
```

Sample Data:

Output pane

Data Output | Explain | Messages | History

	fname character(20)	lname text	room character(20)
1	Ada	molon	209
2	Kaitlyn	Fold	205

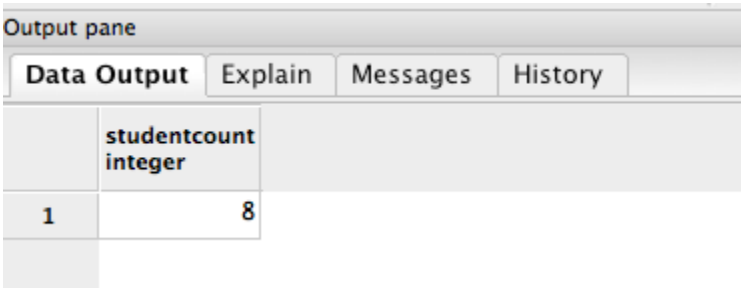
Stored Procedures

Count the total number of students.

Create Statement:

```
--store procedures,Count the total number of students.
CREATE OR REPLACE FUNCTION studentcount() RETURNS INT AS $$
DECLARE
        studentcount INT;
BEGIN
        SELECT COUNT(*)INTO studentcount
        FROM student;
        RETURN studentcount;
END;
$$
language plpgsql;
SELECT studentcount();
```

Sample Data:



Triggers

```
CREATE OR REPLACE FUNCTION current_date_set() RETURNS trigger AS
$date_trigger$ BEGIN
IF NEW.mdate IS NULL OR NEW.mdate = '' THEN NEW.mdate = current_date;
END IF;
RETURN NEW;
END;
$date_trigger$
LANGUAGE plpgsql;

CREATE TRIGGER date_trigger
BEFORE INSERT ON Moveout
FOR EACH ROW
EXECUTE PROCEDURE current_date_set();

drop trigger _trigger on Moveout;
SELECT CURRENT_DATE;
```

Security

There are two types of users identified for the database.

1. Database admins are able to change, update, and maintain all the tables on the database.
   ```
   CREATE ROLE_admin
   GRANT SELECT, INSERT, UPDATE, DELETE
   ON ALL TABLES
   TO db_admin;
   ```
2. Users are only allow to see the database, but not able to modify anything.
   ```
   CREATE ROLE user
   GRANT SELECT
   ON ALL TABLE IN SCHMA USER
   TO user;
   ```

Implementation Notes and Known Problems:

Implementation seems pretty good so far. With all the databases, admin will able to know more about the move out date of students. This database could be more interesting if admin add more items into each table, and add more tables. Since there are all sample database, it would be more useful if having the real database, because it will avoid the redundancies. Also, in the database, it separate the student and staff which means in this case, student do not have the role of being staff. However, in many real situation, student can also play a role as staff. So if there is student who is also staff, there might be more problems.

Future Enhancements

Each table could have include more specific items. For the staff table, it would be better to point out that they are more referring to resident assistant or resident director. Plus, it will be better to add a table to indicate the connection between staff and student, because student can also be a staff in some case.