

**Database Management Design Project**

**MARIST HOUSING MOVE OUT**

**12/01/2014**

Submitted By:

**Siting Wang**

**Table of Contents**

Table of Contents

[**Executive Summary** 3](#_Toc406858151)

[**Entity Relationship Diagram** 4](#_Toc406858152)

[**Objective** 4](#_Toc406858153)

[**Entities** 5](#_Toc406858154)

[**STUDENT TABLE** 5](#_Toc406858155)

[**BUILDING TABLE** 6](#_Toc406858156)

[**VISIT TABLE** 7](#_Toc406858157)

[**VISITOR TABLE** 9](#_Toc406858158)

[**STAFF TABLE** 10](#_Toc406858159)

[**MOVEOUT TABLE** 11](#_Toc406858160)

[**Views** 12](#_Toc406858161)

[**Reports and Queries:** 13](#_Toc406858162)

[**Stored procedures** 14](#_Toc406858163)

[**Triggers** 14](#_Toc406858164)

[**Security** 16](#_Toc406858165)

[**Implementation Notes and Known Problems:** 16](#_Toc406858166)

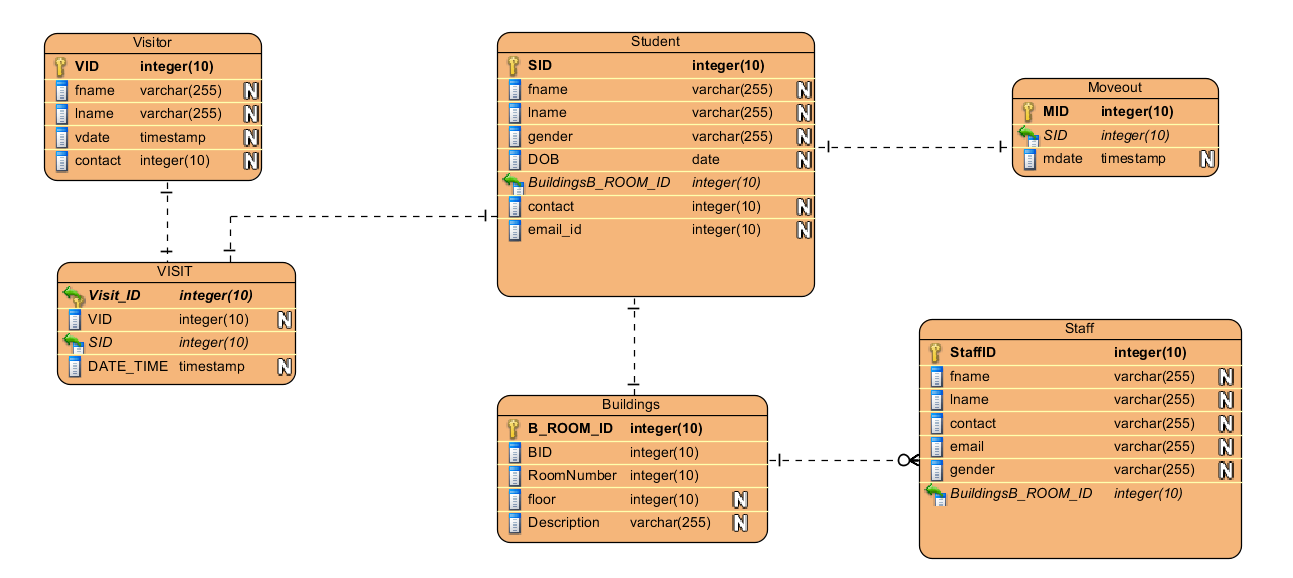
[**Future Enhancements** 16](#_Toc406858167)

# **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**



# **Objective**

This database 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.

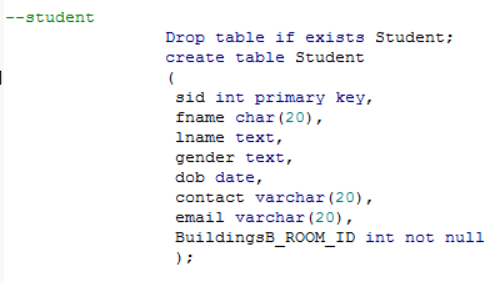
# **Entities**

## **STUDENT TABLE**

**Description:**

This table is needed to keep track the information of students who are living in the campus housing.

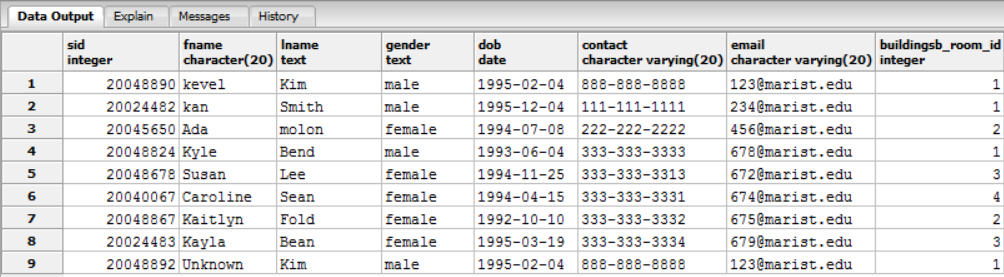
**Create Statement:**



**Insert Statements:**



**Sample Data:**



**Functional Dependencies:**

Sid->fname, lname, gender, birth, contact, email.

## **BUILDING TABLE**

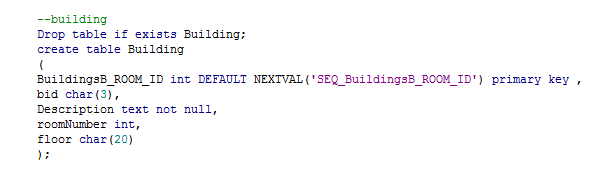
**Description:**

This building table consists of the freshmen buildings, and lists of buildings, floor, and room number.

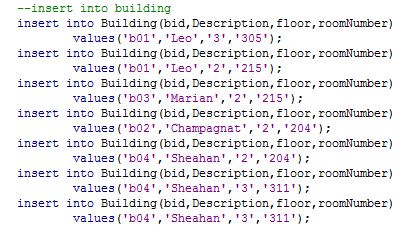
**Create a sequence for Building\_ROOM\_ID column:**



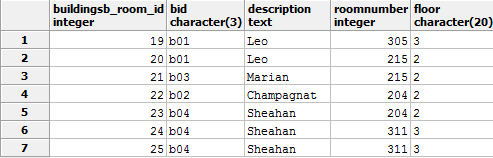
**Create Statement:**



**Insert Statements:**



**Sample Data:**



**Functional Dependencies:**

BuildingsB\_ROOM\_ID-> BID, building, Description, floor, roomNumber

## **VISIT TABLE**

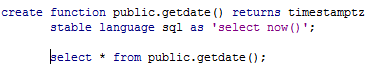
**Description:**

This table contain the information of the visits has been paid to the students staying in the housing.

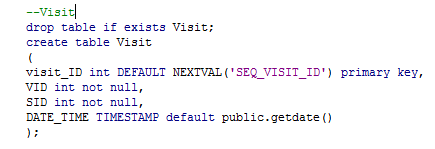
**Create Sequence:**



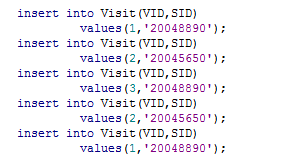
**Create Default current date and time function:**

****

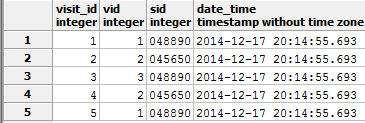
**Create Statement:**



**Insert Statements:**



**Sample Data:**



**Functional Dependency:**

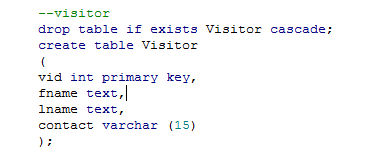
Visit\_id->Vid, Sid, date\_time

## **VISITOR TABLE**

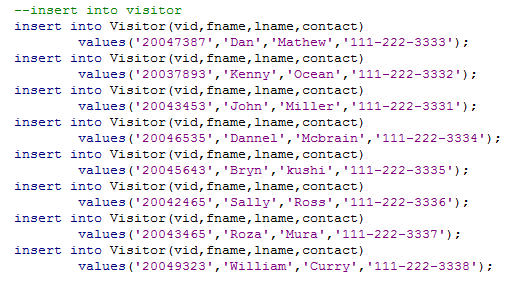
**Description:**

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.

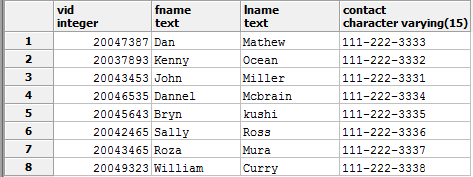
**Create Statement:**



**Insert Statements:**



**Sample Data:**



**Functional Dependency:**

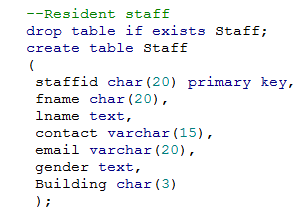
vid->vid,fname,lname,contact

## **STAFF TABLE**

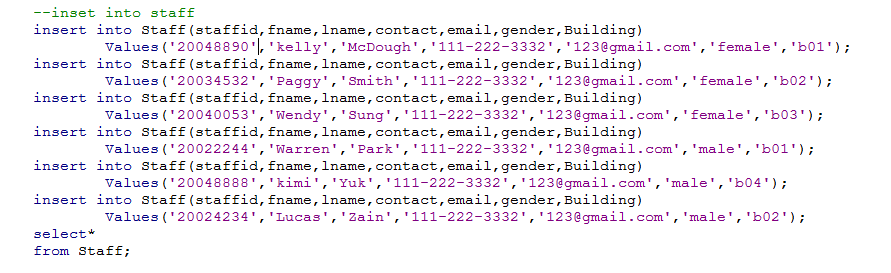
**Description:**

This table lists the staff’s first name, last name, gender, building that they in charge, and all related information.

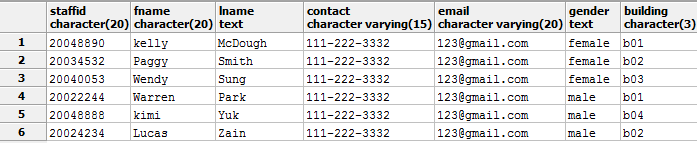
**Create Statement:**



**Insert Statements:**

****

**Sample Date:**



**Functional Dependency:**

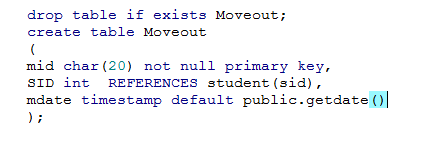
Staid->fname, lname, contact, email, gender, building

## **MOVEOUT TABLE**

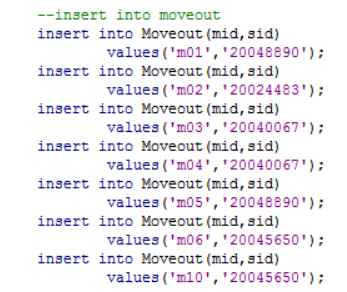
**Description:**

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

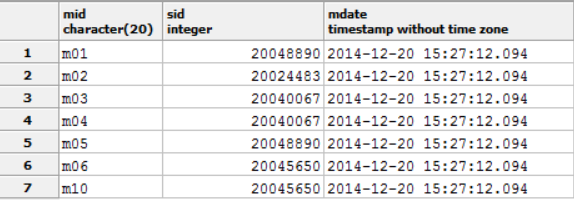
**Create Statement:**



**Insert Statements:**



**Sample Date:**



**Functional Dependency:**

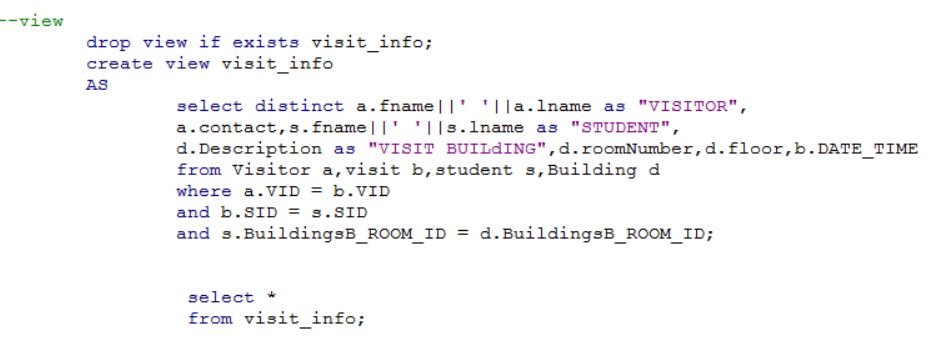
mid->fname,lname,mdate,mtime

# **Views**

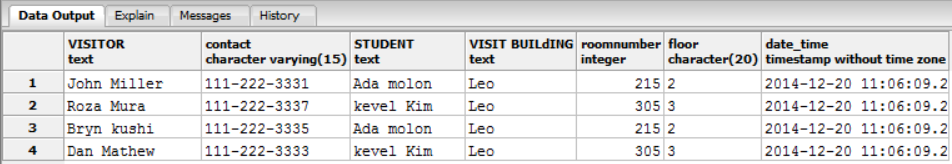
**Description:**

This view displays anyone who visit the students.

**Create Statement:**



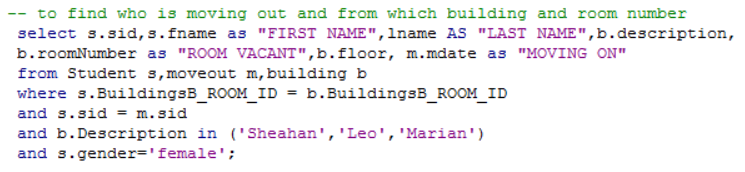
**Sample data:**



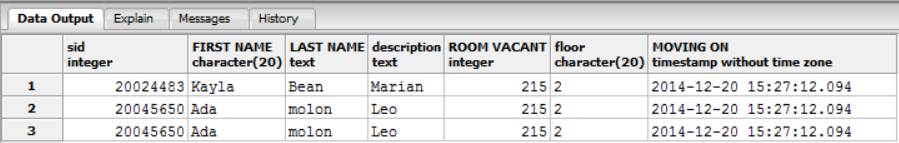
# **Reports and Queries:**

This query gets the information of students who moved out from the housing.

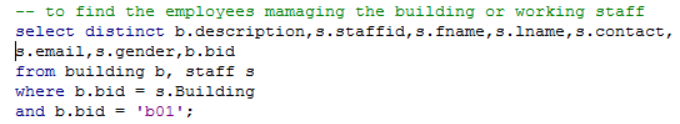
**Create Statement:**



**Sample Data:**



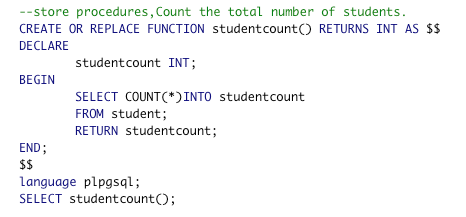
**Create Statement:**



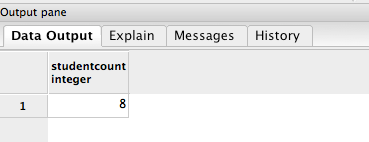
# **Stored procedures**

Count the total number of students.

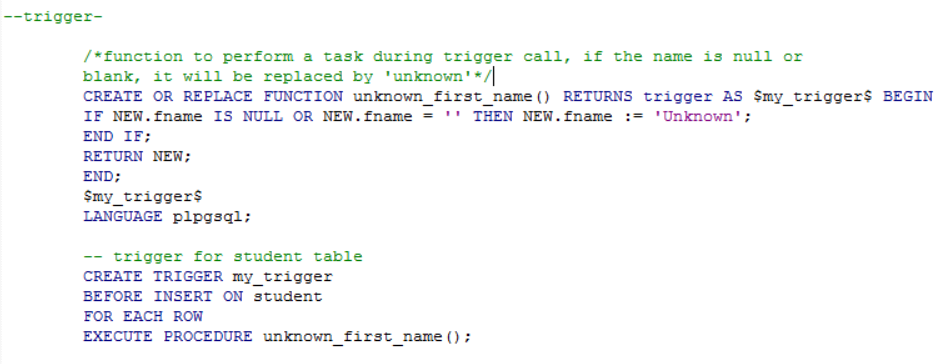
**Create Statement:**



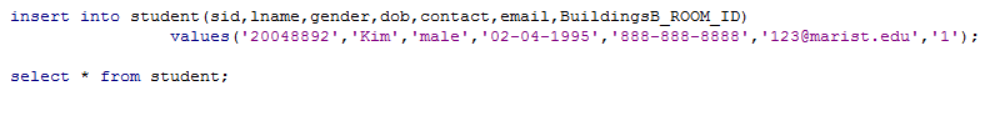
**Sample Data:**



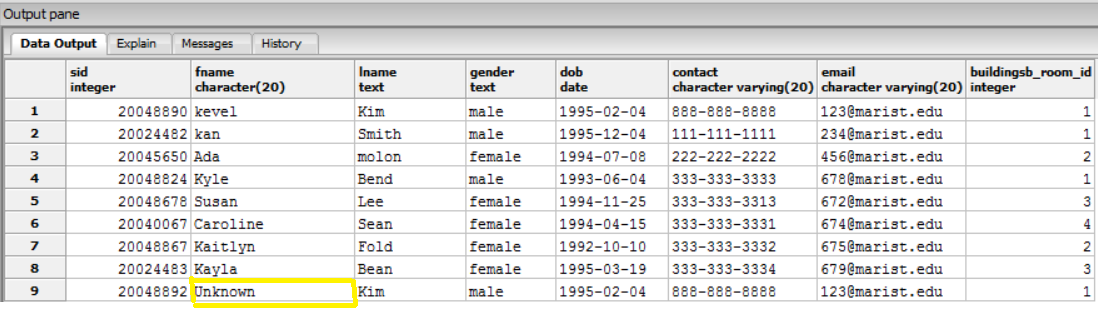
# **Triggers**



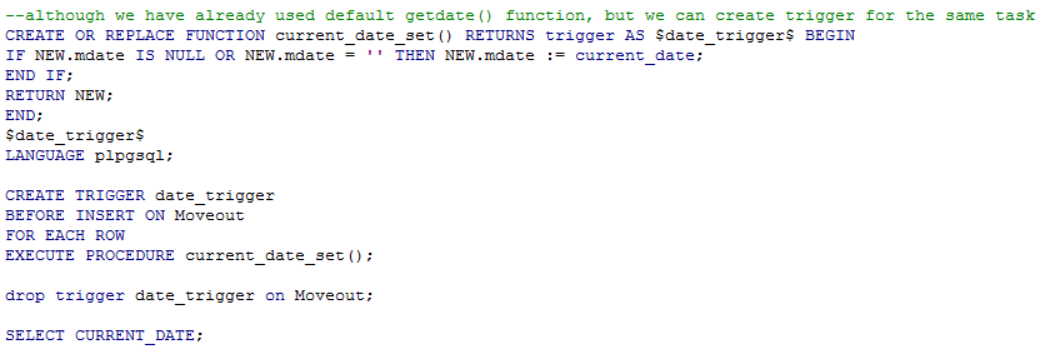
Here, we are not giving any value to fname of the student table, the trigger will automatically set the null value to “Unknown”



Sample output:



Date trigger:



# **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;***”**

1. 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.