Daniel Patrick(CS), Lauren Vaughn(CS)

Assignment 2

11 October 2017

**Database Analysis**

In order to meet both the organizational and information needs of Marshall University Housing, a conceptual database model has been designed. Requirements of this conceptual model, including organizational rules, response time, security needs, and data volume, have been addressed in detail.

**Business Rules**

Business rules include organizational policy, calculations, regulations, and other constraints.

The following **MUHousing** rules have been identified according to database needs:

1. Each student can identify at most one next of kin
2. No two students can identify the same next of kin.
3. A student is assigned to exactly one adviser.
4. An adviser is assigned to at least one student.
5. Each student has at most one lease or does not have a lease (status is ‘waiting’).
6. Each lease belongs to exactly one student.
7. One lease is for one single room in an apartment or one single room in a residence hall.
8. Each single room in a residence hall or apartment has exactly one lease (note: every is rented)
9. Each apartment is inspected regularly by members of staff.
10. Inspection details are recorded for each apartment by members of staff.
11. A member of staff can inspect many apartments.
12. A member of staff can only work in a residence hall or apartment.
13. An apartment or a residence hall must have at least one employee.
14. Each hall can be managed by exactly one member of staff.
15. A student flat contains one or more single rooms.
16. Each single room in a flat is contained in a exactly one flat.
17. Each lease generates exactly one invoice.
18. Each invoice is generated by exactly one lease.
19. A payment is made to exactly one invoice.
20. Invoice is paid by exactly one payment.
21. Each student can register at most one vehicle with the residence office.
22. A vehicle is registered to exactly one student
23. Each vehicle can be designated to park in exactly one lot.
24. A parking lot can have zero or more vehicles registered.
25. A student must be enrolled in at least one course.
26. A course must have at least one student enrolled.
27. Student course information is kept for each semester.

**Response Time**

Information such as student room numbers etc. very quickly. For a given room number, course, vehicle, etc. information should be readily accessible. The response time for most queries should not be more than 10 seconds.

**Conceptual Data Model**

To capture the information needs of MUHousing, an ER Model has been developed. First, the Entity types are identified and then business rules are used to establish the relationships between the entity types.

***Identification of entity types***

The main entity and attribute types are as follows:

* **StudentInfo**: The attributes are muNum, name, homeAddress, mobile, email, birthDate, gender, category, nationality, specialNeeds, addComments, currentStatus, major, minor
* **Adviser**: The attributes are adviserNo, fullName, position, deptName, onCampusPhone, email, roomNo
* **Residence Hall:** The attributes are name, address, phoneNum, roomNo, placeNo, monthlyRent
* **Student Flats**: The attributes are flatNo, address, numAvailableBeds
* **FlatsSingleRooms**: The attributes are monthlyRent, roomNo, placeNo
* **ResidenceStaff**: The attributes are staffNo, name, homeAddress, dob, gender, position, workLocation
* **LeaseInfo:** The attributes are leaseNo, leaseDuration, studentName, muNum, placeNo, addressDateIn, dateOut
* **Payment:** The attributes are datePaid, methodOfPayment, firstRemDate, secondRemDate
* **Invoice**: The attributes are invoiceNum, leaseNo, semester, studentName, muNum, paymentDue, placeNo, roomNo, address
* **NextOfKin:** The attributes arename, address, phoneNum
* **ParkingLot**: The attributes are lotNum, lotName, address, maxSpace, availability
* **Vehicle**: The attributes are vinNum, color, manufacturer, brand
* **Courses**: The attributes are courseNo, courseTitle, instructor, onCampusPhone, email, roomNo, deptName

***Identification of Relations***

A thorough revision of the Business Rules shows that there is a many to zero or one relationship between StudentInfo and leaseInfo. From LeaseInfo there is a one to zero or one relationship between LeaseInfo and FlatSingleRooms. The FlatSingleRooms has another many to one relationship with StudentFlats. It is clear to see that StudentFlats has an n to m (many to many) relationship with ResidenceStaff. It is important to note that the inspection depends on both ResidenceStaff and StudentFlat. Therefore, the information about inspection must be part of the relationship between ResidenceStaff and StudentFlat. In addition to LeaseInfo, there exists another relationship between LeaseInfo and ResidenceHall that is zero or one to many. From ResidenceHall there is a zero or one to many relationship with ResidenceStaff. Lastly for the LeaseInfo, there is another one to one relationship between LeaseInfo and Invoice. From Invoice, there exists a one to one relationship with StuPay

Secondly, StudentInfo also has another many to one relationship with Adviser.

Thirdly, StudentInfo also holds an n to one relationship with NextOfKin.

Furthermore, StudentInfo and Courses has an n to m (many to many) relationship. It is important to note that student course information depends on both StudentInfo and Courses. Therefore, the information in StuCourse must be part of the relationship between StudentInfo and Courses.

Lastly, there is a noticeable one to zero or one relationship between StudentInfo and VehicleInfo. We can also see that there is a many to one relationship from VehicleInfo to ParkingLot.

Figure 1 Includes an ER Model to meet the needs of the MUHousing database. Visual Paradigm has been used to create this model. Figure 1 includes two association classes (entity type) that contain inspection details and student course details. The association class, Inspection, hangs from the relationship labeled Inspects. It means that the association class holds the information captured by the Inspects relationship. The association class, StuCourse, hangs from the relationship labeled Enrolled. It means that this association class holds the information captured by the Enrolled relationship.