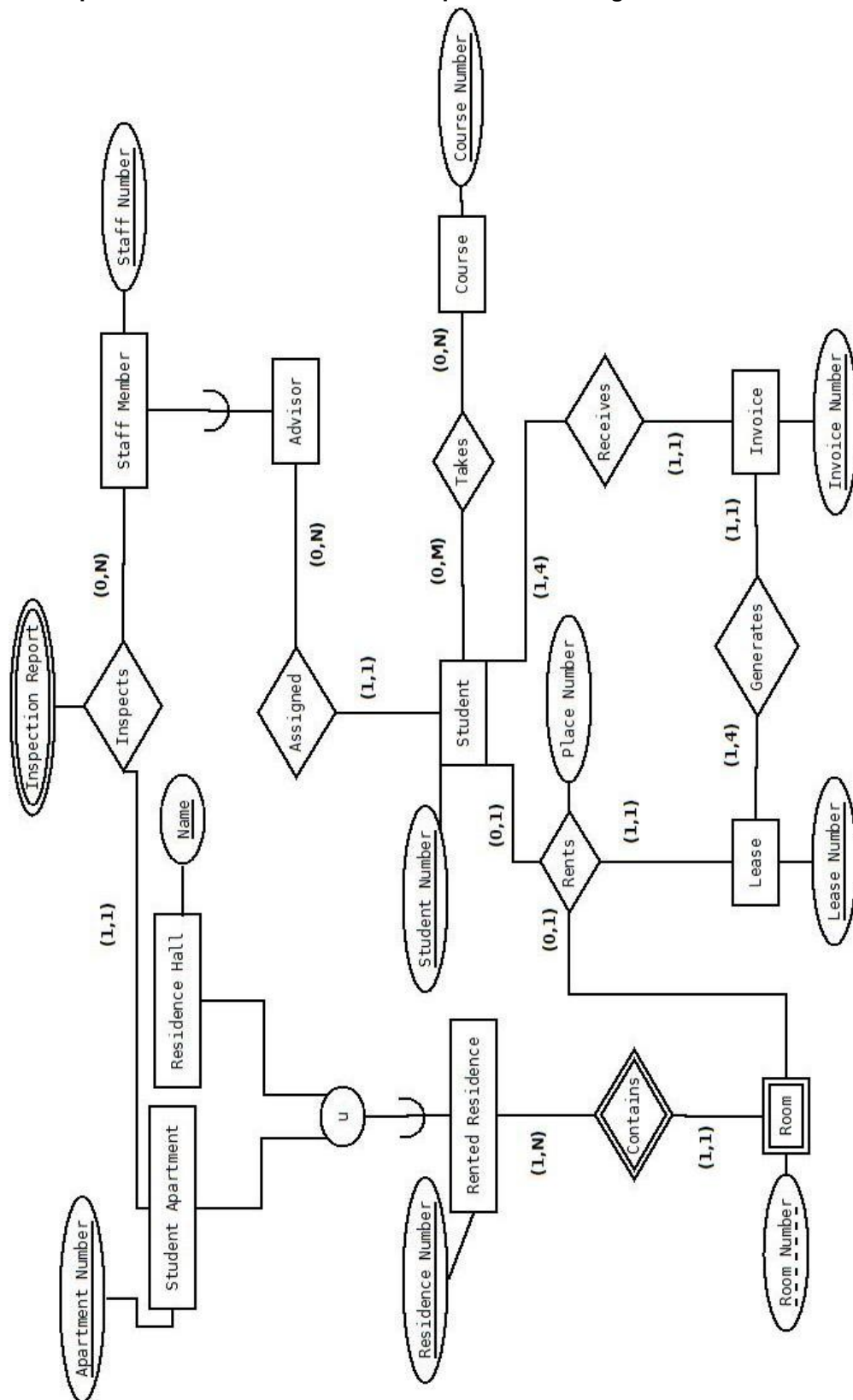
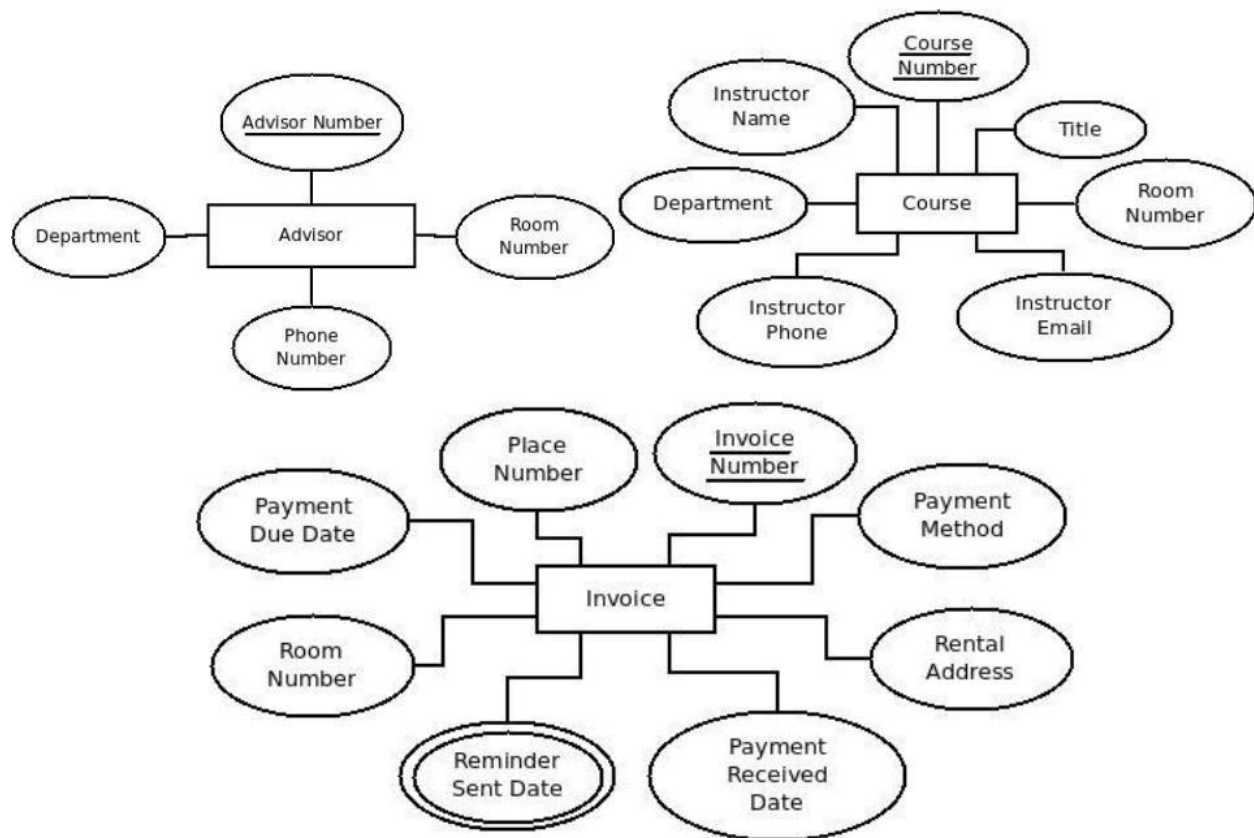


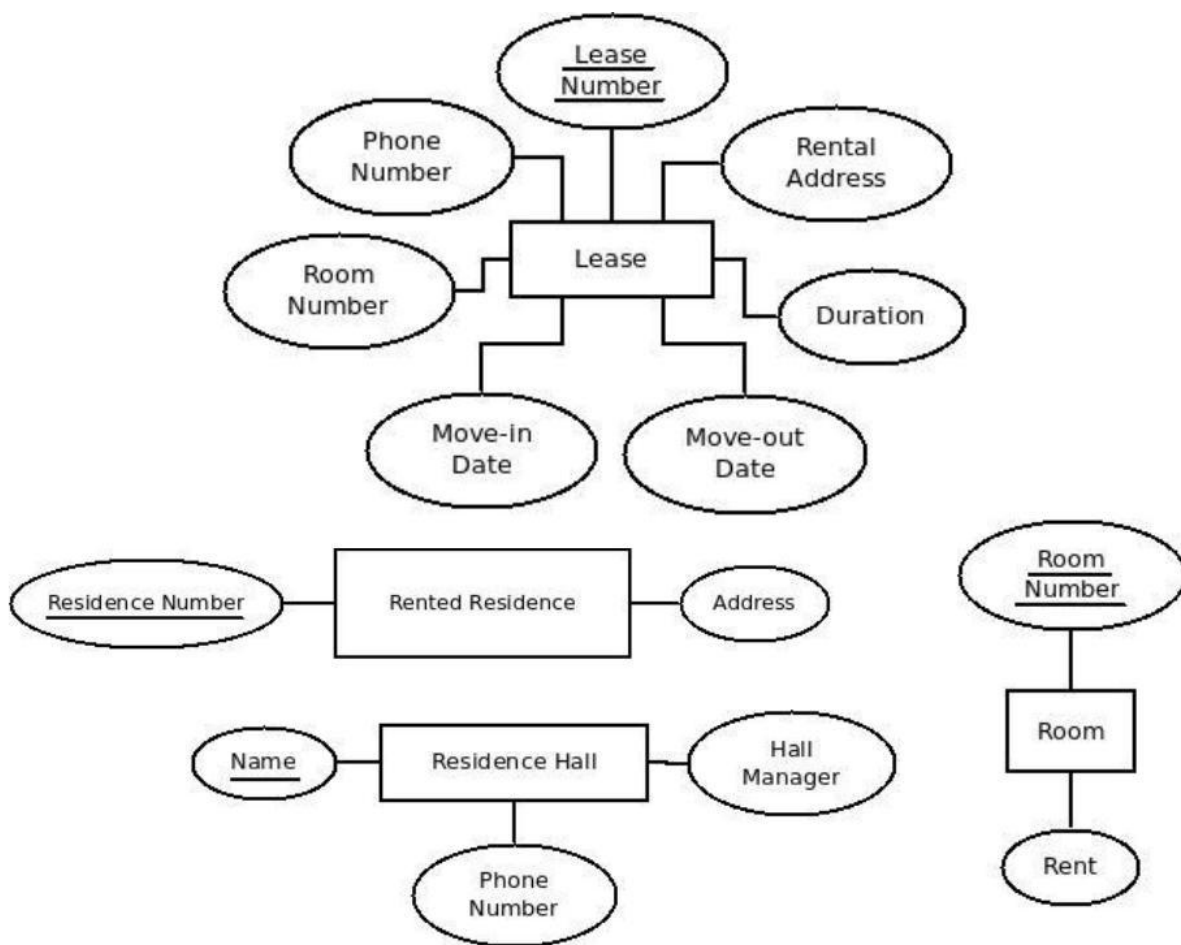
Revised Conceptual Model: Part A2 - Relationships/Entities using Structural Constraints



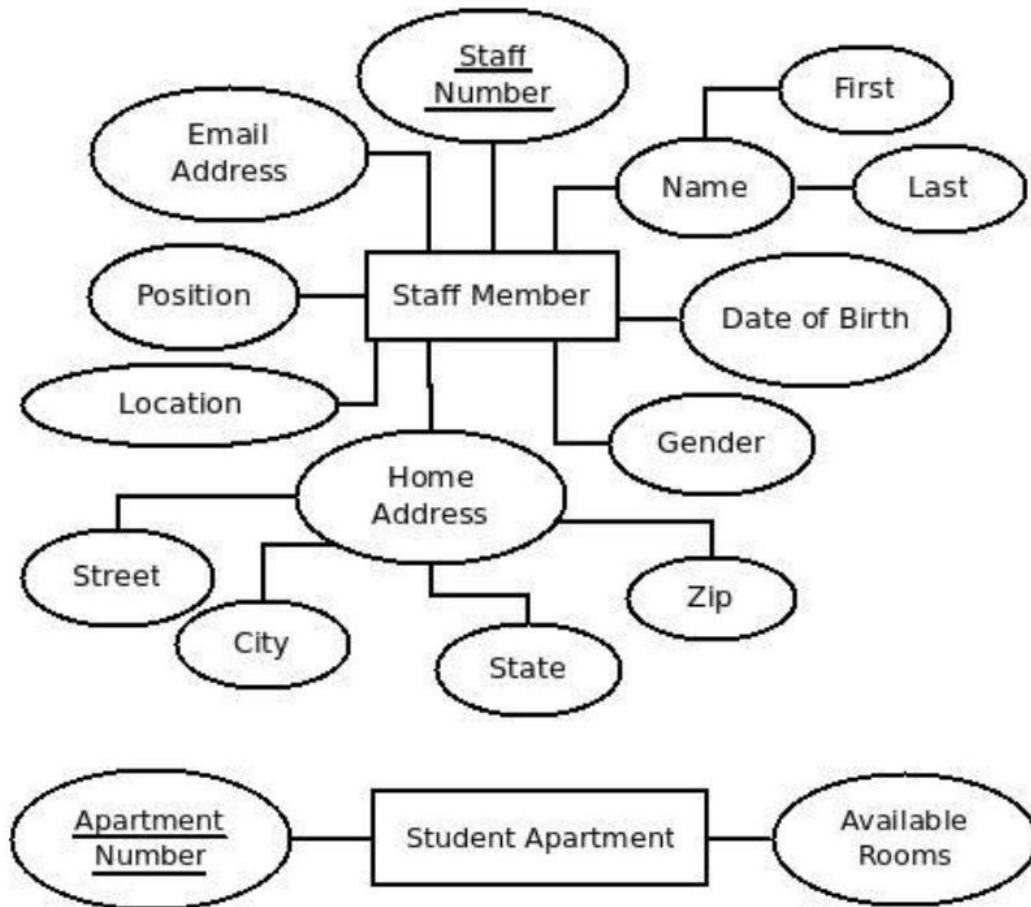
Revised Conceptual Model: Part B - Entities/Attributes



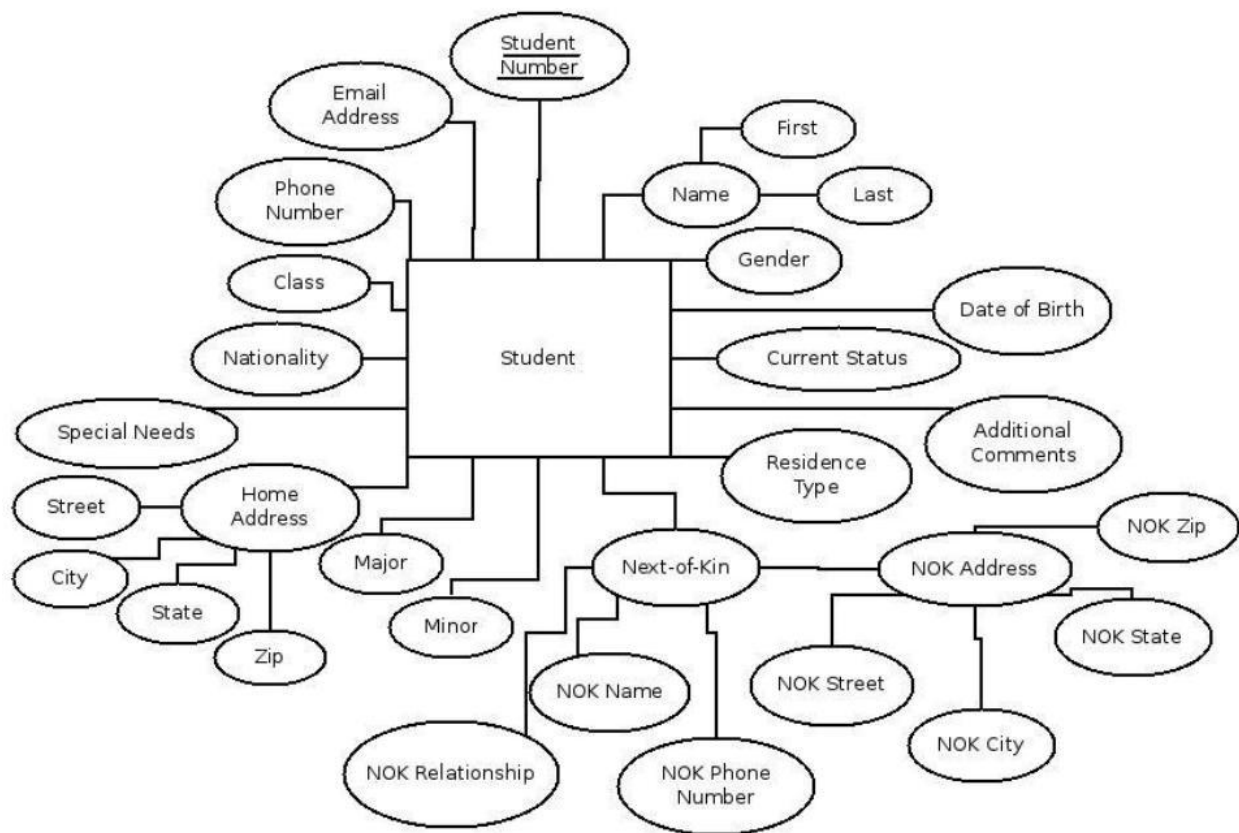
Revised Conceptual Model: Part B - Entities/Attributes continued



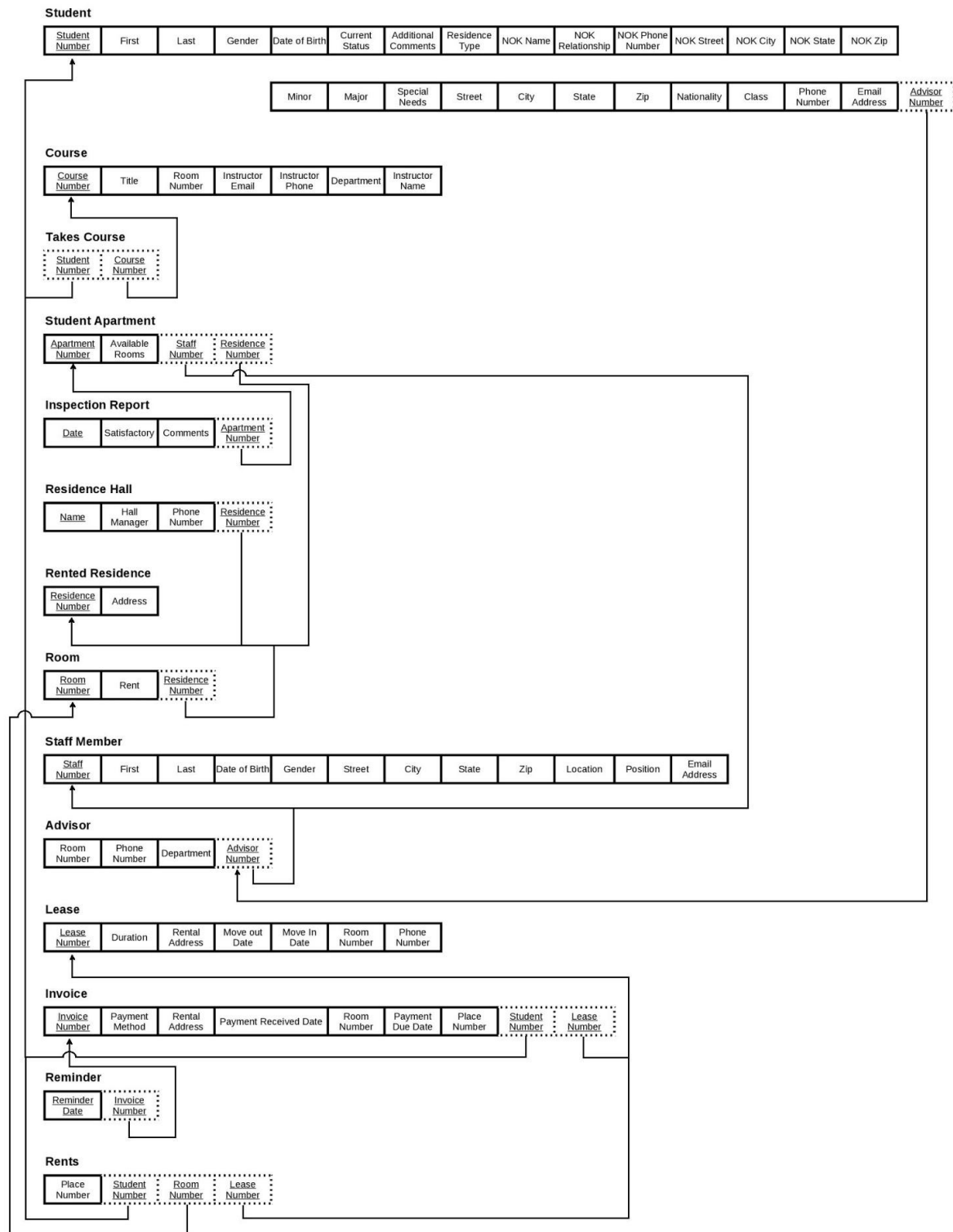
Revised Conceptual Model: Part B - Entities/Attributes continued



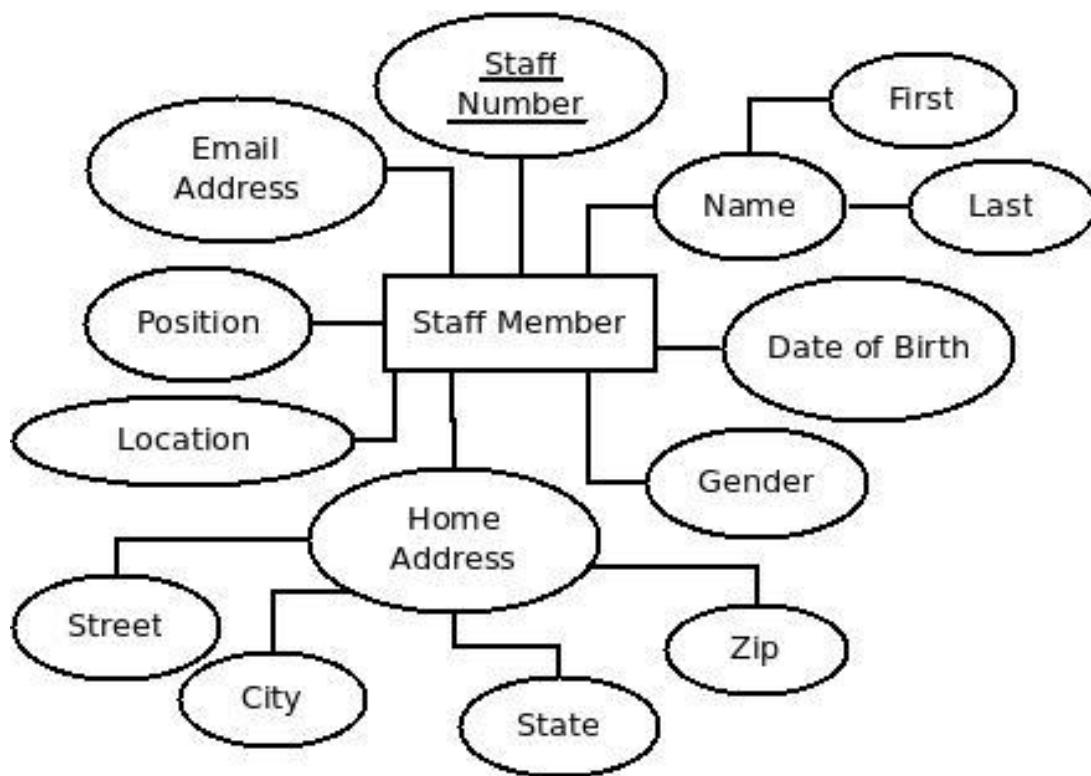
Revised Conceptual Model: Part B - Entities/Attributes continued



Revised Relational Model Schema



1a. Strong Entity with a Composite Attribute - Revised



Entity Diagram for Staff Member

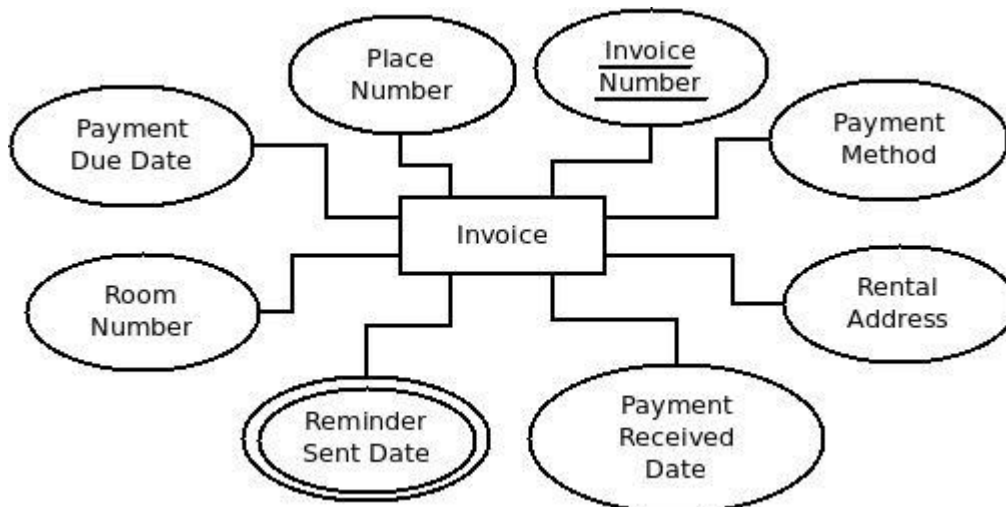
Staff Member

<u>Staff Number</u>	First	Last	Date of Birth	Gender	Street	City	State	Zip	Location	Position	Email Address
---------------------	-------	------	---------------	--------	--------	------	-------	-----	----------	----------	---------------

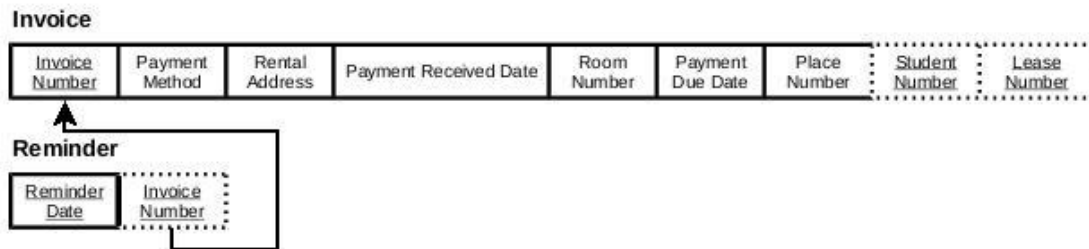
Relational Diagram for Staff Member

Explanation: In the case of Staff Member's composite attribute 'Home Address', the parent attribute was omitted. All that was needed to be done was to place all the leaf attributes that were connected to the composite in the relation. In this case, Street, City, State, & Zip were placed directly into the relation.

1b. Strong Entity with a Multi-valued Attribute - Revised



Entity Diagram for Invoice



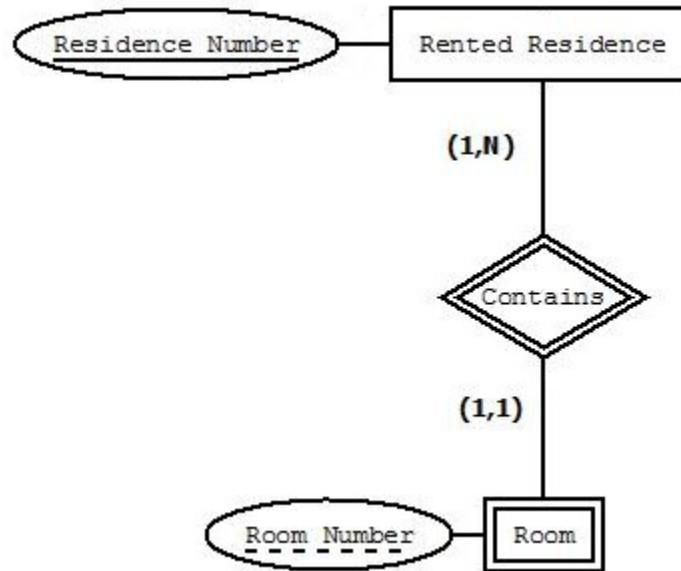
Relational Diagram for Invoice

Explanation: To convert the Invoice entity into an equivalent relational model it was necessary to employ both the 1st and 6th step mentioned in Chapter 9 of the text. The primary key for Invoice, 'Invoice Number' along with all the singular attributes are included in the Invoice table. Appended to the end of the Invoice relation is the foreign key 'Lease Number' which links Invoice to the Lease entity via the Generates relationship. In order to properly convert Invoice's multi-valued attribute 'Reminder Sent Date', a new relation, called Reminder, was formed which includes the attribute itself along with Invoice's primary key 'Invoice Number' as a foreign key.

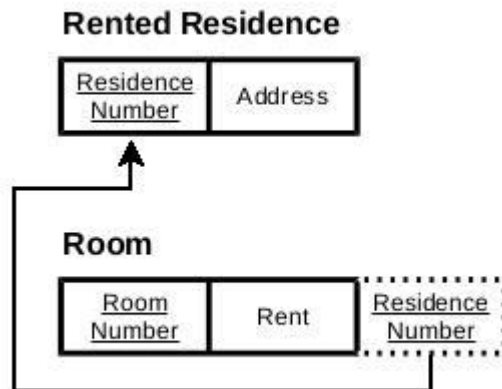
1c. Strong Entity with a Derived Attribute

NOT APPLICABLE FOR OUR PROJECT

1d. Weak Entity - Revised



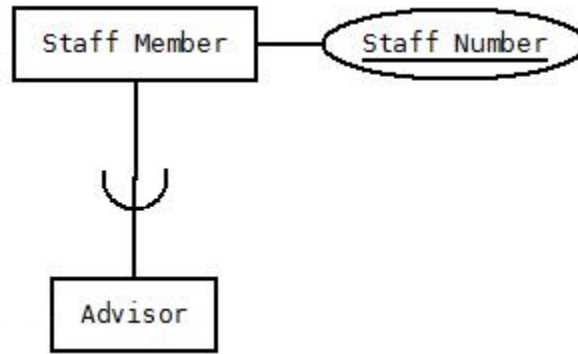
Entity-Relationship Diagrams for Room and Rented Residence



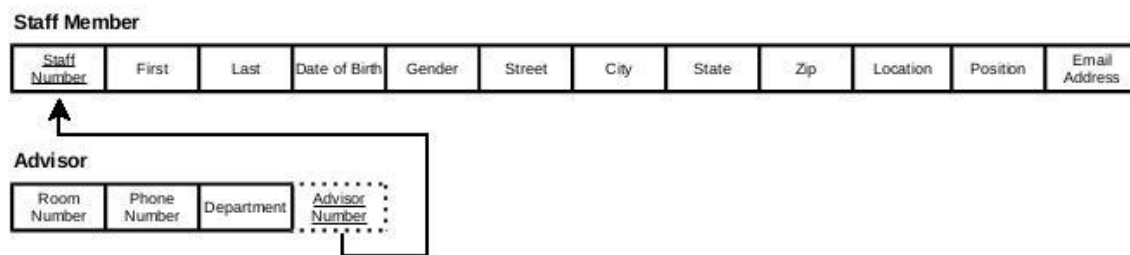
Relational Diagrams for Room and Rented Residence

Explanation: Converting the Room entity into a table is straightforward, as it only has two attributes, namely Room Number and Rent. Also included in the relation for Room is Residence Number, the foreign key linking Room to the Rented Residence entity via the identifying-relationship Contains.

1e. Specialization/Generalization/Aggregation - Revised



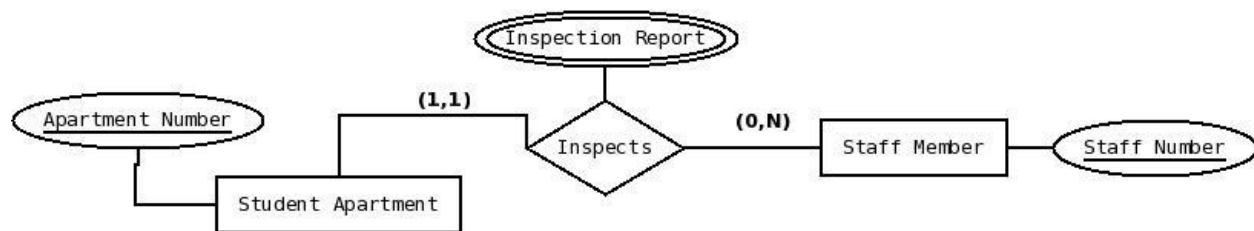
Entity-Relationship Diagram for Staff Member and Advisor



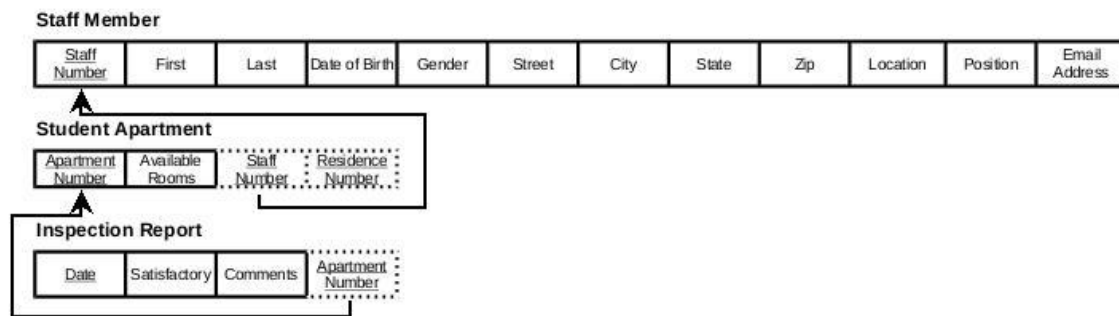
Relational Diagram for Staff Member and Advisor

Explanation: To convert the specialization relationship that exists between Staff Member and Advisor, the Staff Member relation and all its attributes was added, along with the Advisor relation and all its attributes. Option 8D from chapter 9 is designed to handle overlapping subclasses by including a Boolean flag or type fields. This option was not chosen because it would create many NULL values because we have partial participation. Option 8C is used to handle disjoint subclasses by including a single type attribute to indicate to which of the m subclasses each tuple belongs. This option was not chosen because if the specialization is partial, it would also create NULL values similar to 8D. Option 8B works well only when both the disjoint and total constraints hold. If the specialization is not total, an entity that does not belong to any of the subclasses is lost. Option 8A was used to make the conversion because it eliminated any NULL values and clearly showed the relationship between the two relations. The primary key from Staff Member, Staff Number, was added as a foreign key to Advisor. By adding this as a foreign key, the relational schema now matches the entity relationship above.

1f. 1-M Relationship - Revised



Entity-Relationship Diagram for Staff Member, Student Apartment and Inspection Report



Relation Diagram for Staff Member, Student Apartment and Inspection Report

Entities: Staff Member, Apartment

Relationship Attributes: {Inspection Report (Date, Satisfactory, Additional Comments)}

Cardinality Explanation: A staff member may inspect many apartments and each apartment inspection is done by one staff member.

Participation: Apartment has total participation in the relationship because all apartments must be inspected. Staff Member has partial participation because not all staff members carry out inspections.

Conversion Explanation: Because Apartment has total participation in the Inspects relationship (which is 1-M), the primary key of Staff Member is added to the Apartment relation as a foreign key. 'Inspection Report', the composite, multi-valued attribute of the Inspects relationship becomes the 'Inspection Report' relation with all of the composite attributes and the primary key of Apartment, 'Apartment Number' as a foreign key.

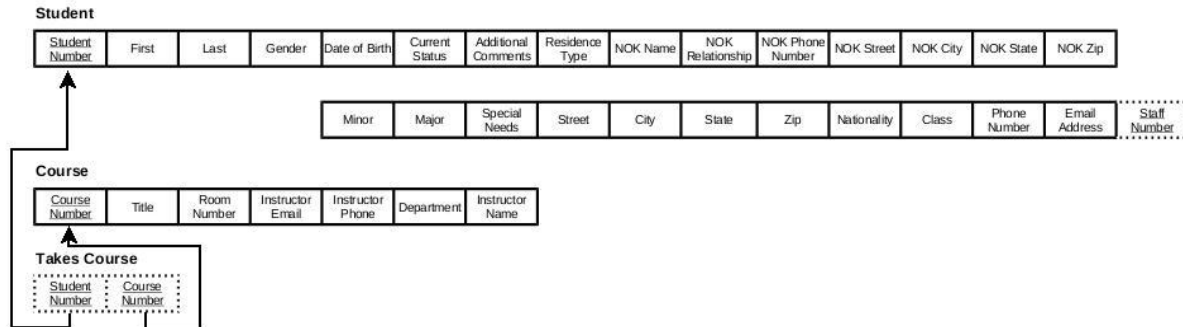
1g. 1-1 Relationship

NOT APPLICABLE FOR OUR PROJECT

1h. M-N Relationship - Revised



Entity-Relationship Diagram for Student and Course



Relation Diagram for Student, Course, and Takes Course

Entities: Student, Course

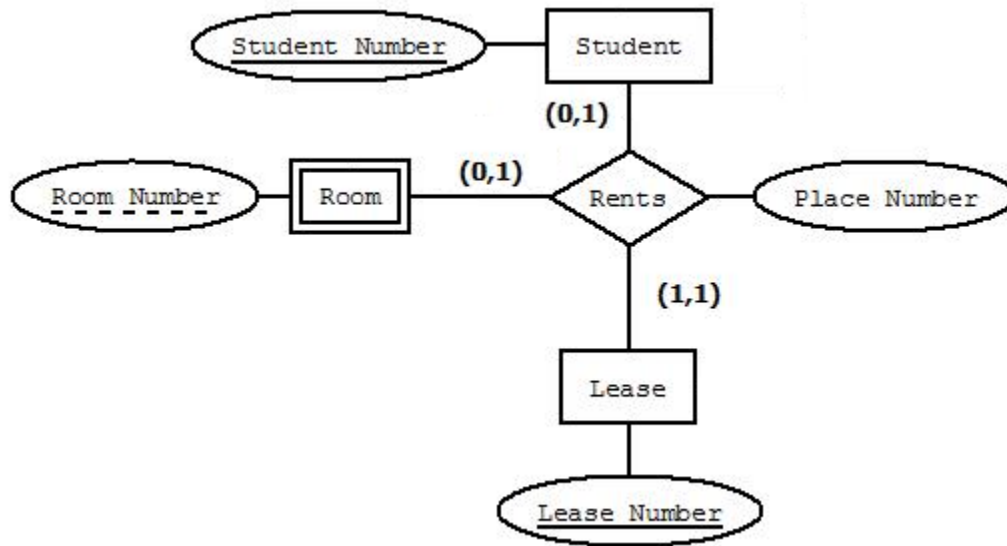
Relationship Attributes: None

Cardinality Explanation: Looking at the Entity Relationship diagram, it shows that a student can enroll in 0 to many different courses, while a course can have 0 to many students as well.

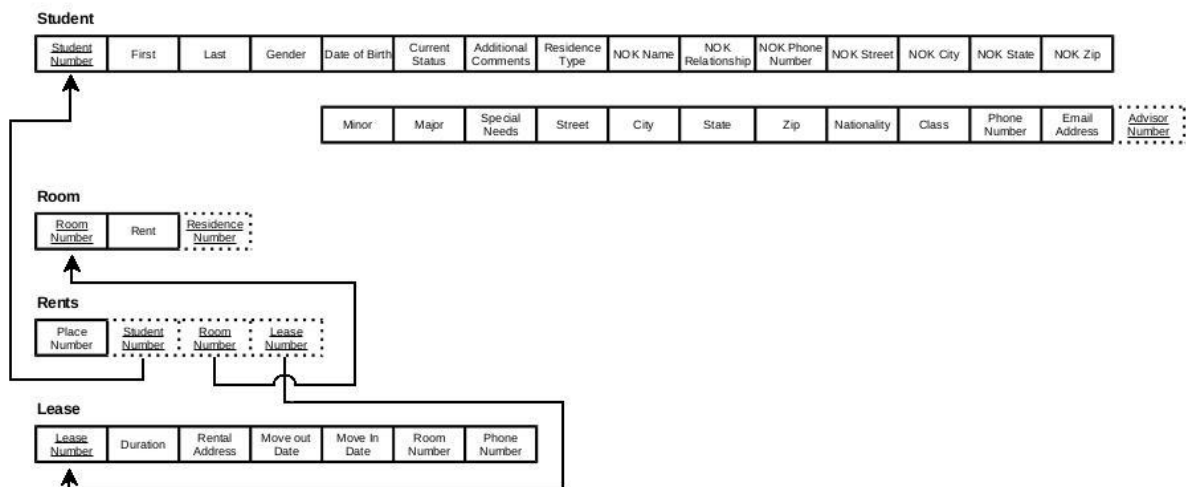
Participation: Both sides have partial participation

Conversion Explanation: In order to convert the Entity Relationship diagram to the relational diagram, you first start out with step 1 from Chapter 9 of the textbook and add as a relation each entity involved in the many to many relationship. In this case, it is the Student and Course entities. Included in each relation is all the attributes associated with each entity. Next to show the many to many relationship between the two entities, we need a new relation that we called 'Takes Course'. This relation only has two attributes. First is the primary key of the Student relation, and second is the primary key from the Course relation. Both of these primary keys are added as foreign keys in the 'Takes Course' relation.

1i. Ternary Relationship - Revised



Entity-Relationship Diagram for Student, Room, and Lease



Relation Diagram for Student, Room, Lease, and Rents

Entities: Student, Room, Lease

Relationship Attributes: Place Number

Cardinality Explanation: Looking at the Entity Relationship diagram, it shows a ternary relationship between the entities. The structural constraints show that a student and a room can exist without this relationship, but a lease, since it's min is 1, has to be involved with one rents relationship to exist. It also shows that the rents relationship will contain only one of each of the entities, Student, Room, and Lease. These three entities together form the ternary relationship.

1i. Ternary Relationship Continued - Revised

Participation: Student and Room have partial participation in the relationship, while Lease has total participation.

Conversion Explanation: In order to convert this ternary ER diagram into a relational schema, a new relation must be created. This relation, that is called Rents, ties the other three relations together. The primary keys of Student, Room, and Lease are included in this Rents relation as foreign keys. We also add the one relationship attribute, Place Number, to the rents relation.

Comments on Project - Relational Model Phase

A. Difficulties you faced in doing this implementation phase and how they were resolved?

The main difficulty we faced as a group was determining where a foreign key should point when specialization and generalization is involved. We weren't sure if the arrow should point to the super-class

or the sub-class. In the end, we spoke with Dr. Mendoza who informed us that in our case, it should point to the sub-class. The only other difficulty was just making sure we didn't miss a foreign key anywhere. Even after going through the rules from Chapter 9 one at a time, we still were noticing things we were missing on the first go through.

B. Likes and dislikes about this part of the project?

Carbajal, Enrique: I liked this part of the project because it seems more straight forward. I mean this because we have already done the hard work in creating the conceptual model to our liking. Now, in this part of the project we simply need to create relations that seem to follow the conceptual model naturally.

Small, James: I liked that on this part of the project, there is really only one way to convert our ER diagram to a relational schema. This seems like it will make it easier as there isn't room for interpretation. I disliked having to try and fit the relational schema on one page. Some of our entities have a lot of attributes that made it difficult.

Takahashi, Mark: I enjoyed all portions of the second part of the project. It was very straight forward and Dr. Mendoza did a great job explaining what she expected us to do in class.

Urbach, Daniel: I enjoyed the process of transforming the concepts laid on in the ER diagram into actual tables and figuring out how to properly map their interactions.

C. What was the most challenging aspect of this part of the design?

Carbajal, Enrique: This most challenging part of this design phase was trying to make it to the scheduled meeting time. It is difficult for me to meet on time with my group because I have class during the scheduled time meeting.

Small, James: The most challenging aspect was making sure we had everything included in our relational schema. There were a lot of arrows to be shown and it got confusing making sure they all were there where they are supposed to be.

Takahashi, Mark: The most challenging part was cutting up portions of our entity diagrams and entity relation diagrams and neatly organizing them on the final document to turn in.

Urbach, Daniel: The most challenging aspect of this portion of the project for me was understanding when to include foreign keys in the various steps used to translate the ER diagrams into relations.

D. Suggestions on how to improve this part of project?

Carbajal, Enrique: Thus far I have no suggestions on how to improve this phase of the project.

Comments on Project - Relational Model Phase

Everything smooth and well.

Small, James: Not much to say here as there really wasn't anything that needed to be improved on. Like you had mentioned, this was a very easy part of the project that we hopefully got right.

Takahashi, Mark: I believe we could have improved this part of the project if we had more time to work together as a team. However this part of the project was much easier and less work to do.

Urbach, Daniel: I have no suggestions for this segment of the project

E. Did you make changes to your communication methods? If yes, why?

Carbajal, Enrique: We didn't not make any changes to the way we communicated for this phase of the project. Most of the work was done in class and during a scheduled meeting. The rest of the work was done over email to tie up loose ends.

Small, James: No changes to the communication methods. Email, and in person were the primary means of communication.

Takahashi, Mark: We continued to use the same communication methods.

Urbach, Daniel: Our communication remains unchanged; there was really no reason to change it because it has been working out well so far.