91.3902 Database Systems
Assignment #1

3.1 Discuss the role of a high-level data model in the database design process.
High-level data models assist in conceptual design and helps express data requirements of the users and includes detailed descriptions of the entity types, relationships, and constraints. The high-level data model is also used as a reference to ensure that all users' data requirements are met and that the requirements do not include conflicts.

3.2 List the various cases where use of a null value would be appropriate.
Null value would be appropriate when a particular entity does not have an applicable value for an attribute. There are two cases for such situations. The first case arises when it is known that the attribute value exists but is missing. An example for cases like such would be if the Height attribute of a person is listed as null. The second case arises when it is not known whether the attribute value exists. An example would be is the HomePhone attribute of a person is null.

3.3 Define the following terms: entity, attribute, attribute value, relationship instance, composite attribute, multivalued attribute, derived attribute, complex attribute, key attribute, value set (domain).

Entity
Entity is a "thing" in the real world with an independent existence. It can also be an object with physical existence (i.e. person, car) or conceptual existence (i.e. company, job).

Attribute
An attribute is a particular property that describes entity (i.e. person name company name).

Attribute Value
Attribute values are major data stored in the database.

Relationship Instance
Relationship Instance is an association of entities, where the association includes exactly one entity from each participating entity type.

Composite Attribute
Composite attribute is an attribute that can be divided into meaningful components.

Multivalued Attribute
Multivalued attribute is an attribute that can have many values.

Derived Attribute

Derived attribute is an attribute whose value is computed from another attribute or combination of attributes.

Complex Attribute

Complex attribute is composite and multivalued attributes nested in an arbitrary manner.

Key Attribute

Key attribute is an attribute whose values are distinct for each individual entity in the collection.

Value Set (Domain)

Value set specifies the set of values that may be assigned to that attribute for each individual entry.

3.4 What is an entity type? What is an entity set? Explain the differences among an entity, an entity type, and entity set.

Entity type defines a collection of entities that have the same attributes. Entity set is a named collection of related data. The differences between entity, an entity type, and entity set are:

1. **Entity is a person, place, thing, event, or even a concept. It may be tangible or intangible. Examples of an entity are employees,**
2. **Entity type describes the schema or intension for a set of entities that share the same structure.**
3. **Entity set is the extension of the entity type.**

3.5 Explain the difference between an attribute and a value set.

An attribute is a particular property that describes entity. A value set specifies the set of values that may be assigned to that attribute for each individual entry. An example of an attribute is EmployeeAge and we can set the value set for this attribute to a range of integer numbers.

3.6 What is a relationship type? Explain the differences among a relationship instance, a relationship type, and a relationship set.

Relationship type is the nature of a relationship between entities, expressed by the number of their possible occurrences in the related tables. The differences between a relationship instance, a relationship type, and a relationship set are:

1. **Relationship instance is an association of entities, where the association includes exactly one entity from each participating entity type.**
2. **Relationship type defines a relationship set among entities from these types.**
3. **Relationship set is a set of associations.**

3.7 What is a participation role? When is it necessary to use role name in the description of relationship type?

Participation role is the part that each entity participates in a relationship. It is necessary to use role name in the description of relationship type when the same entity type participates more than once in a relationship type in different roles. In other words, role names are necessary in recursive relationships.

3.8 Describe two alternatives for specifying structural constraints on relationship types. What are the advantages of each.

Two alternatives for specifying structural constraints on relationship types are cardinality ratio and participation. Cardinality ratio for a binary relationship specifies the number of relationship instances that an entity can participate in. The participation constraint specifies whether the existence of an entity depends on its being related to another entity via the relationship type.

3.9 Under what conditions can an attribute of a binary relationship type be migrated to become an attribute of one of the participating entity type?

An attribute of a binary relationship type can be migrated to become an attribute of one the participating entity type when the relationship type is 1:1 or 1:N. This is because each entity participates in at most one relationship instance. However, for 1:N relationship type, a relationship attribute can be migrated only to the entity type at the N-side of the relationship.

3.10 When we think of relationships as attributes, what are the value sets of these attributes? What class of data models is based on this concept?

The value sets of these attributes

3.11 What is meant by a recursive relationship type? Give some examples of recursive relationship types.

A recursive relationship type exists if an entity can be related to itself or in other words, the same entity type participates more than once in a relationship type in different roles. An example of a recursive relationship type is courses that require one or more other courses as prerequisites. The course entity is related to another course entity. In this case, the recursive relationship "course is a prerequisite to course" also happens to be a M:N relationship. This is because a course can have more than one prerequisite, and a course can be a prerequisite to many other courses.

Another example of a recursive relationship type is in a supervision relationship type between an employee and a supervisor. Both entities are members of the same Employee entity type. In this example, the employee entity type participates twice in the supervision relationships, once in the role of a supervisor, and once in the role of supervisee.

3.12 When is the concept of weak entity used in data modeling? Define the terms owner entity type, weak entity type, identifying relationship type, and partial key.
The concept of weak entity is used in data modeling when there are many attributes or when entity types do not have key attributes of their own.

Owner Entity Type
Owner entity type is an entity type that is related to weak entity types through combination of some of their attribute values.

Weak Entity Type
Weak entity type is an entity type that does not have key attributes of their own.

Identifying Relationship Type
Identifying relationship type is the relationship type that relates a weak entity type to its owner.

Partial Key
Partial key is a set of attributes that can uniquely identify weak entities that are related to the same owner entity.

3.13 Can an identifying relationship of a weak entity type be of a degree greater than two? Give examples to illustrate your answer.
An identifying relationship of a weak entity type can be of a degree greater than two. For a ternary identifying relationship type, the weak entity type has several owner entity types. An example for a ternary relationship such as Supply must be represented as weak entity type, with no partial key and with three identifying relationships. These participating entity types Supplier, Part, and Project are together the owner entity types. Hence, an entity in the weak entity type Supply is identified by the combination of its three owner entities from Supplier, Part, and Project.

3.14 Discuss the convention for displaying an ER schema as an ER diagram.
Entity types are shown in rectangular boxes. Relationship types are shown in diamond-shaped boxes attached to the participating entity types with straight lines. Attributes are shown in ovals, and each attribute is attached by a straight line to its entity type or relationship type. Component attributes of a composite attribute are attached to the oval representing the composite attribute. Multivalued attributes are shown in double ovals and key attributes have their names underlined. Derived attributes are shown in dotted ovals.

Weak entity types are distinguished by being placed in double rectangles and by having their identifying relationship placed in double diamonds. The partial key of the weak entity type is underlined with a dotted line.

The cardinality ratio of each binary relationship type is specified by 7attaching a 1, M, or N on each participating edge. The participation constraint is specified by a single line for partial participation and by double lines for total participation. We show the role names when one entity type plays both roles in a relationship.

3.15 Discuss the naming conventions used for ER schema diagram.
Entity types are usually singular names, rather than plural ones, because the entity type name applies to each individual entity belonging to that entity type. Entity type and relationship type names are in uppercase letters, attribute names are capitalized, and role names are in lowercase letters.

Given a narrative description of the database requirements, we usually use nouns for entity type names, and the verbs to indicate names of relationship types. Attribute names general arise from additional nouns that describe the nouns corresponding to the entity types.

Relationship names should be readable from left to right and from top to bottom.

3.19 Consider the ER diagram of Figure 3.16, which shows a simplified schema for an airline reservations system. Extract from the ER diagram the requirements and constraints that produced this schema. Try to be as precise as possible in your requirements and constraints specification.

Entity Types:

Airport

1. Each airport is identified by **AirportCode**.
2. We need to keep track of the **Name, City,** and **State** for each airport.

Airplane Type

1. Each airplane type is identified by **TypeName**.
2. We need to keep track of the **Max-seats** for each airplane type.

Airplane

1. Each airplane is identified by **AirplaneId**.
2. We need to keep track of the **Total-no-of-seats** for each airplane.

Flight Leg

1. We need to track of the **LegNo** for each flight leg.

Flight

1. Each flight is identified by **Number**.
2. We need to keep track of the **Weekdays** and **Airline** for each flight.

Leg Instance

1. We need to keep track of the **Date** and **No-of-avail-seats** for each leg instance.
2. Each leg instance must be assigned to at least one airplane.
3. Each leg instance must be an instance of a flight leg.

Relationships, attributes, and constraints:

1. **Relationship between FLIGHT and FARES**
   a. Each instance of FLIGHT may offer multiple instances of FARE.
   b. Instances of FARE must be sold for a specific instance of FLIGHT.

2. **Relationship between FLIGHT and FLIGHT LEG**
   a. Each instance of FLIGHT may consist of multiple instances of FLIGHT LEG.
   b. Instances of FLIGHT LEG must belong to a specific instance of FLIGHT

3. **Relationship between FLIGHT LEG and AIRPORT (ScheduledDepTime)**
   a. Each instance of AIRPORT may be scheduled for the departure of multiple instances of FLIGHT LEG.
   b. Instances of FLIGHT LEG must be scheduled for departure from a specific instance of AIRPORT.

4. **Relationship between FLIGHT LEG and AIRPORT (ScheduledArrTime)**
   a. Each instance of AIRPORT may be scheduled for the arrival of multiple instances of FLIGHT LEG.
   b. Instances of FLIGHT LEG must be scheduled for arrival at a specific instance of AIRPORT.

5. **Relationship between FLIGHT LEG and LEG INSTANCE**
   a. Each instance of FLIGHT LEG may consist of instances of LEG INSTANCE.
   b. Instances of LEG INSTANCE must be instances of FLIGHT LEG.

6. **Relationship between LEG INSTANCE and AIRPORT (ArrTime)**
   a. Each instance of AIRPORT may accommodate the arrival of multiple instances of LEG INSTANCE.
   b. Instances of LEG INSTANCE must arrive at a specific instance of AIRPORT.

7. **Relationship between LEG INSTANCE and AIRPORT (DepTime)**

a. Each instance of AIRPORT may accommodate the departure of multiple instances of LEG INSTANCE.
b. Instances of LEG INSTANCE must depart from a specific instance of AIRPORT.

8. **Relationship between LEG INSTANCE and SEAT (CustomerName, CPhone)**
   a. Each instance of LEG INSTANCE may have multiple instances of SEAT for reservation.
   b. Instances of SEAT must be reserved for a specific instance of LEG INSTANCE.

9. **Relationship between AIRPLANE and LEG INSTANCE**
   a. Each instance of AIRPLANE may be assigned to multiple instances of LEG INSTANCE.
   b. Instances of LEG INSTANCE must have a specific instance of AIRPLANE assigned to them.

10. **Relationship between AIRPLANE TYPE and AIRPLANE**
    a. Each instance of AIRPLANE TYPE may consist of multiple instances of AIRPLANE.
    b. Instances of AIRPLANE must belong to a specific instance of AIRPLANE TYPE.

11. **Relationship between AIRPORT and AIRPLANE TYPE**
    a. Multiple instances of AIRPORT may be available for multiple instances of AIRPLANE TYPE for landing.
    b. Multiple instances of AIRPLANE TYPE may land at multiple instances of AIRPORT.

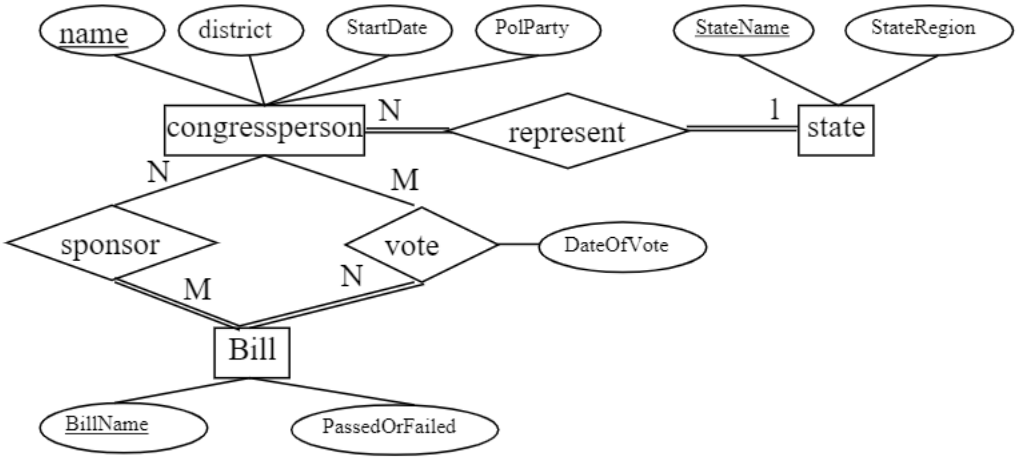
3.21 Design an ER diagram for keeping track of information about votes taken in the U.S. House of Representatives during the current two-year congressional session…. Draw an ER schema diagram for the above application. State clearly any assumption you make.

Entities:

STATE
Name, RegionOfState

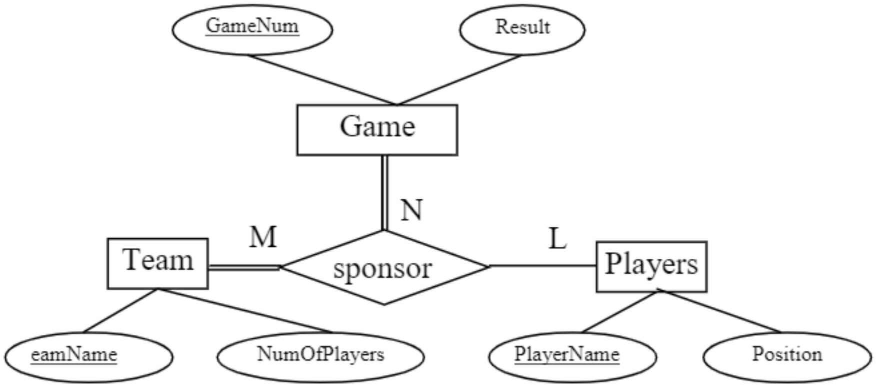
CONGRESSPERSON
Name (FName, MInit, LName), District, StartDate, PoliticalParty

BILL
BillName, PassedOrFailed,



3.21 Design an ER diagram for keeping track of information about votes taken in the U.S. House of Representatives during the two-year congressional session. The database needs to keep track of each U.S. state's Name (e.g. Texas, New York, California) and includes the region of the state (whose domain is {North East, Midwest, Southwest, West}). Each CONGRESSPERSON in the House of Representatives us described by their Name, and includes the District represented, the StartDate when they were first elected, and the political Party they belong to (whose domain is {Republican, Democrat, Independent, other}). The database keeps track of each BILL (i.e. proposed law), and includes the BillName, the DateOfVote on the bill, whether the bill PassedOrFailed (whose domain is {YES, NO}), and the sponsor (the congressperson(s) who sponsored – i.e., proposed – the bill). The database keeps track of how each congressperson voted on each bill (domain of vote attribute is {Yes, No, Abstain, Absent}). Draw an ER schema diagram for the above application. State clearly any assumption you make.

3.22 A database is being constructed to keep track of the teams and games of a sport league. A team has a number of players, not all of whom participate in each game. It is desired to keep track of the players participating in each game for each team, the positions they played in that games, and the result of the games. Try to design an ER schema diagram for this application, stating any assumption you make. Choose your favourite sport (soccer, baseball, football …)


GameNum
Result
Game
N
M
L
Team
sponsor
Players
eamName
NumOfPlayers
PlayerName
Position