ST. XAVIER’S COLLEGE

**Maitighar, Kathmandu**

**(Affiliated to Tribhuvan University)**



**Database Management System**

**Theory Assignment #5**

**Submitted By**

Alok Shrestha

B.Sc. CSIT

Year II/IV Semester

013BSCIT005

**Submitted To**

Er. Sanjay Yadav

Lecturer

Department of Computer Science

St. Xavier’s College

Maitighar, Kathmandu

**Submitted On**

August, 2015

**Entity Relationship model**

**1. What do you mean by Entity- Relationship Diagram? Explain.**

An Entity Relationship Diagram (ERD) is a visual representation of different data using conventions that describe how these data are related to each other. An E-R diagram can express the overall logical structure of a database graphically. E-R diagrams are simple and clear—qualities that may well account in large part for the widespread use of the E-R model. Major components of E-R diagram are:

* Rectangles: representing entity sets.
* Ellipses: representing attributes.
* Diamonds: representing relationship sets.
* Lines: linking attributes to entity sets and entity sets to relationship sets.
* Double ellipses: represents multivalued attributes.
* Dashed ellipses: represents derived attributes
* Double lines: represents total participation of entity in relationship sets
* Double rectangles: represents weak entity sets (discuss in later section)
* Underline: indicates primary key attributes

**2. Define entity and give an example.**

An entity is a “thing” or “object” in the real world that is distinguishable from other objects. For example, each person is an entity, and bank accounts can be considered as entities. The set of all entities of the same type is termed an entity set.

3. Explain the different between an entity class and an entity instance.

**4. Define attribute and its types.**

Attributes are descriptive properties possessed by each member of an entity set. The designation of an attribute for an entity set expresses that the database stores similar information concerning each entity in the entity set; however, each entity may have its own value for each attribute.

For example, possible attributes of the instructor entity set are ID, name, dept name, and salary. In real life, there would be further attributes, such as street number, apartment number, state, postal code, and country, but we omit them to keep our examples simple. Possible attributes of the course entity set are course id, title, dept name, and credits. Each entity has a value for each of its attributes. For instance, a particular instructor entity may have the value 12121 for ID, the value Wu for name, the value Finance for dept name, and the value 90000 for salary.

**5. What is derived attributes?**

* Simple and composite attributes: The attributes are simple if they cannot be divided into subparts. Composite attributes, on the other hand, can be divided into subparts (that is, other attributes). For example, an attribute name could be structured as a composite attribute consisting of first name, middle initial, and last name.

Using composite attributes in a design schema is a good choice if a user will wish to refer to an entire attribute on some occasions and to only a component of the attribute on other occasions. Suppose we were to add an address to the student entity-set. The address can be defined as the composite attribute address with the Attributes Street, city, state, and zip code. Composite attributes help us to group together related attributes, making the modeling cleaner.

* Single-valued and multivalued attributes: The attributes that have a single value for a particular entity is called single valued attribute. For example, the student ID attribute for a specific student entity refers to only one student ID.

There may be instances where an attribute has a set of values for a specific entity. Suppose we add to the instructor entity set a phone number attribute. An instructor may have zero, one, or several phone numbers, and different instructors may have different numbers of phones. This type of attribute is said to be multivalued. As another example, we could add to the instructor entity set an attribute dependent name listing all the dependents. This attribute would be multivalued, since any particular instructor may have zero, one, or more dependents.

* Derived attribute. The value for this type of attribute can be derived from the values of other related attributes or entities. For example, suppose that the instructor entity set has an attribute age that indicates the instructor’s age. If the instructor entity set also has an attribute date of birth, we can calculate age from date of birth and the current date. Thus, age is a derived attribute. In this case, date of birth may be referred to as a base attribute, or a stored attribute. The value of a derived attribute is not stored but is computed when required.

**6. Define relationship and give an example.**

A relationship is an association among several entities. For example, a member relationship associates an instructor with her department, a depositor relationship associates a customer with each account he or she has. The set of all relationships of the same type is termed a relationship set.

7. Explain the difference between a relationship class and a relationship instance.

**8. Define degree of relationship.**

The number of entity sets that participate in a relationship set is the degree of the relationship set.

**9. List and give an example of the three types of binary relationships. Draw an E-R diagram for each.**

Binary relationship set is the one that involves two entity sets. Most of the relationship sets in a database system are binary. Occasionally, however, relationship sets involve more than two entity sets. A binary relationship set is of degree 2

**10. Define the terms maximum cardinality and minimum cardinality.**

**11. Explain the distinctions among the terms primary key, candidate key and super key.**

The concept of key is important to distinguish one entity from another and one relationship from another relationship. In fact, values of attributes distinguish one entity from another entity. To distinguish one entity from another entity in entity set there must exist attribute/s whose values must not duplicate in entity set. It ensures no two entities in an entity set can exist with same values for all attributes.

* Super key: A super key is a set of one or more attributes which uniquely identifies an entity in entity set. For example: in customer relation single attribute customer\_id is sufficient to uniquely identify one customer entity to another. So customer\_id is a superkey in a customer relation. Since combination of customer\_id and customer\_name can also uniquely identifies one customer entity to another. So combination of attributes {customer\_id,customer\_name} is also superkey in relation customer. But single attribute customer\_name can not superkey in relation customer because customer name only can not uniquely identify one customer entity to another, there would be number of customers having same name.

The above example of supekey shows that superkey may contains extraneous attributes. That is, if K is superkey then any superset of K is superkey.

* Candidate key: The minimal superkey called candidate key. That is, candidate key is a superkey but its proper subset is not superkey. For example: customer\_id is a candidate key in customer relation. Similarly account\_id is a candidate key in account relation.
* Primary key: In a relation, it is possible that we can choose distinct set of attributes as a candidate key. For example: in customer we can choose single attribute {custome\_id} or set attributes {customer\_name, customer\_city} as candidate key. Candidate key chosen by database designer for particular relation known as primary key.

**12. What are the main building modules of the entity relationship model? Discuss each one.**

An entity is a “thing” or “object” in the real world that is distinguishable from another object. An entity set is a set of entities of the same type that share the same properties. Entities of an entity set has same set of attributes.

Attribute is descriptive property of entity set. Set of attributes describes entity set.

Major components of E-R diagram are:

* Rectangles: representing entity sets.
* Ellipses: representing attributes.
* Diamonds: representing relationship sets.
* Lines: linking attributes to entity sets and entity sets to relationship sets.
* Double ellipses: represents multivalued attributes.
* Dashed ellipses: represents derived attributes
* Double lines: represents total participation of entity in relationship sets
* Double rectangles: represents weak entity sets (discuss in later section)
* Underline: indicates primary key attributes

**13. What is composite attributes, when it is used?**

Composite attributes are the ones that can be divided into subparts (that is, other attributes). For example, an attribute name could be structured as a composite attribute consisting of first name, middle initial, and last name. Using composite attributes in a design schema is a good choice if a user will wish to refer to an entire attribute on some occasions and to only a component of the attribute on other occasions. Suppose we were to add an address to the student entity-set. The address can be defined as the composite attribute address with the Attributes Street, city, state, and zip code. Composite attributes help us to group together related attributes, making the modeling cleaner.

14. Explain the difference between single-value attributes and simple attributes.

15. Discuss the difference between a composite key and a composite attribute. How would each indicated in an E-R diagram?

16. What two courses of action are available to a designer when a multivalued attribute is encountered?

17. Explain the various terms of an E-R model and how are they represented in an E-R model?

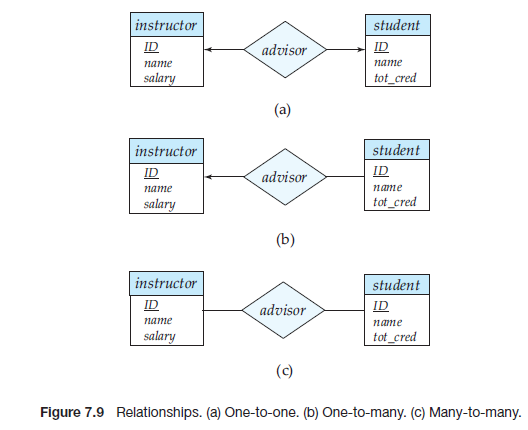
18. Explain the concept of dependent entities? Give example.

**19. What is the difference total and partial participation? Explain.**

Same as Q22

**20. What do you mean by mapping cardinalities? Explain various types of cardinalities.**

Mapping Cardinalities describes no. of entities to which another entity can be associated via relationship set. Mapping cardinalities are most useful in describing binary relationship sets but it can also describe relationship sets that involve more than two entity sets. For binary relationship set between entity set A and B mapping cardinality must one of the following.

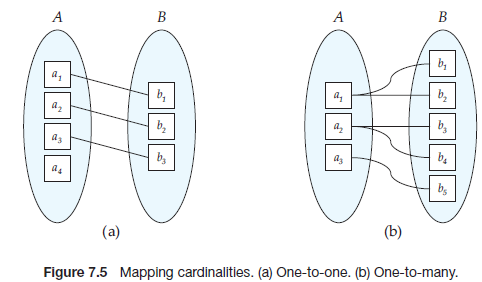
* One-to-one: We draw a directed line from the relationship set advisor to both entity sets instructor and student (see Figure 7.9a). This indicates that an instructor may advise at most one student, and a student may have at most one advisor.
* One-to-many: We draw a directed line from the relationship set advisor to the entity set instructor and an undirected line to the entity set student (see Figure 7.9b). This indicates that an instructor may advise many students, but a student may have at most one advisor.
* Many-to-one: We draw an undirected line from the relationship set advisor to the entity set instructor and a directed line to the entity set student. This indicates that an instructor may advise at most one student, but a student may have many advisors.
* Many-to-many: We draw an undirected line from the relationship set advisor to both entity sets instructor and student (see Figure 7.9c). This indicates that an instructor may advise many students, and a student may have many advisors. 

**21. What is the difference between single-value and multivalued attributes? Explain.**

The attributes that have a single value for a particular entity is called single valued attribute. For example, the student ID attribute for a specific student entity refers to only one student ID.

There may be instances where an attribute has a set of values for a specific entity. Suppose we add to the instructor entity set a phone number attribute. An instructor may have zero, one, or several phone numbers, and different instructors may have different numbers of phones. This type of attribute is said to be multivalued. As another example, we could add to the instructor entity set an attribute dependent name listing all the dependents. This attribute would be multivalued, since any particular instructor may have zero, one, or more dependents.

**22. Explain the concept of participation constraints.**



The participation of an entity set E in a relationship set R is said to be total if every entity in E participates in at least one relationship in R. If only some entities in E participate in relationships in R, the participation of entity set E in relationship R is said to be partial. In Figure 7.5a, the participation of B in the relationship set is total while the participation of A in the relationship set is partial. In Figure 7.5b, the participation of both A and B in the relationship set are total. For example, we expect every student entity to be related to at least one instructor through the advisor relationship. Therefore the participation of student in the relationship set advisor is total. In contrast, an instructor need not advise any students. Hence, it is possible that only some of the instructor entities are related to the student entity set through the advisor relationship, and the participation of instructor in the advisor relationship set is therefore partial.

**23. Difference the binary relationship with ternary relationship with example.**

Relationship set that involves only two entity sets known as binary-relationship set. For example: depositor relationship set is a binary relationship set where relationship set involves only two entity set “customer” and “account”. Most relationship sets in database system are binary. However relationship set may involves in more than two entity sets.

Relationship set that involves three entity sets known as ternary relationship. For example: the relationship set “work-on” among employee, branch and job is example of ternary relationship. The no. of entity sets that participate in relationship set refers degree of relationship set. Here degree of ternary relationship is 3.

**24. Explain the difference between weak and strong entity set.**

An entity set that does not have sufficient attributes to form a primary key is termed a weak entity set. An entity set that has a primary key is termed a strong entity set.

**25. Define the components of extended E-R features.**

Same as Q 29

**26. Define the concept of aggregation. Give two examples of where this concept is useful.**

Same as Q29 third part

**27. Explain the distinction between disjoint and overlapping constraints.**

* Disjoint. A disjointness constraint is the one that requires that an entity belong to no more than one lower-level entity set. For example, student entity can satisfy only one condition for the student type attribute; an entity can be either a graduate student or an undergraduate student, but cannot be both.
* Overlapping. In overlapping constraints, the same entity may belong to more than one lower-level entity set within a single generalization. For an illustration, consider the employee work-team example, and assume that certain employees participate in more than one work team. A given employee may therefore appear in more than one of the team entity sets that are lower level entity sets of employee.

**28. Explain the distinction between total and partial constraints.**

Same as Q 22

**29. Write short notes on:**

**· Specialization**

The process of designating subgroupings within an entity set is called specialization. Specialization follows top down design approach. Entity sets are subgroups in distinct entity sets. For example entity set person with attributes name, street and city can further subgroup into two entities sets customer and employee. Each of these person types can describes by set of attributes that includes all the attributes of entity set person plus all possible attributes of itself. For example, customer entity set can further described by set of attributes: customer\_id, enroll\_date etc. Similarly entity attributes can further describes by set of attributes: emplouee\_id, salary etc. The process of sub groupings within an entity set is called specialization. We can apply specialization repeatedly to refine a design schema. For instance bank employees may be further classified into officer, teller or secretary. In E-R diagram, specialization can be represented by a triangle component labeled ISA. The label ISA stands for “is a “. For example customer is a person, officer is an employee etc. The ISA relationship also called super class-subclass relationship.

Specialization stems from a single entity set; it emphasizes differences among entities within the set by creating distinct lower-level entity sets. These lower level entity sets may have attributes, or may participate in relationships, that do not apply to all the entities in the higher-level entity set. Indeed, the reason a designer applies specialization is to represent such distinctive features. If student and employee have exactly the same attributes as person entities, and participate in exactly the same relationships as person entities, there would be no need to specialize the person entity set.

**· Generalization**

Generalization is a containment relationship that exists between a higher-level entity set and one or more lower-level entity sets. Higher- and lower-level entity sets also may be designated by the terms superclass and subclass, respectively. For example, The person entity set is the superclass of the employee and student subclasses.

Generalization follows bottom-up approach in which multiple entity sets are synthesized into higher-level entity set on the basis of common features. For example, the database designer may have first identified a customer entity set with the attributes: name,street, city and customer\_id and employee entity set with the attributes name, street, city, employee\_id and salary. In both entities some attributes are common. These similarities between these two entities can be expressed by generalization. During the course of database design or E-R schema for enterprise database designer may use both specialization and generalization process. Specialization and generalization in E-R diagram represent by a same way. The terms specialization and generalization are used interchangeably.

Generalization proceeds from the recognition that a number of entity sets share some common features (namely, they are described by the same attributes and participate in the same relationship sets). On the basis of their commonalities, generalization synthesizes these entity sets into a single, higher-level entity set. Generalization is used to emphasize the similarities among lower-level entity sets and to hide the differences; it also permits an economy of representation in that shared attributes are not repeated.

**· Aggregation**

Aggregation is an abstraction through which relationships are treated as higher-level entities.

E-R model cannot express relationship among relationship. To illustrate this, let us consider quaternary relationship manages among employee, branch, job and manager. Its main job is to record managers who manages particular job/task perform by particular employee at particular branch. This quaternary relationship is required since binary relationship between manager and employee can not represent required information. This E-R diagram is able to represent the required information but information are redundant since every employee, branch and job exist both relationship set “work-on” and “manages”. Here aggregation is better to represent such information. Aggregation is in fact an abstraction it treats relationships as higher level entities. In our example, it treats relationship set work-on (including entity set employee, branch and job) as entity set. So now we can create binary relationship set “manages” between work-on and manager. This removes redundant information.

