ST. XAVIER’S COLLEGE

**Maitighar,Kathmandu**

**(Affiliated to Tribhuvan University)**



**Database Management System**

**LAB ASSIGNMENT #5**

**Submitted By**

Abhishek Tamrakar

B.Sc. CSIT

Year II/IV Semester

013BSCIT003

**Submitted To:**

Er. Sanjay Kumar Yadav

Lecturer,

Department of Computer Science

St. Xavier’s College

Maitighar, Kathmandu

**Submitted On:**

27th August, 2015

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

An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and can be used as the foundation for a relational database.

An entity-relationship diagram—otherwise known as an ERD—is a data modeling technique that creates an illustration of an information system's entities and the relationships between those entities.

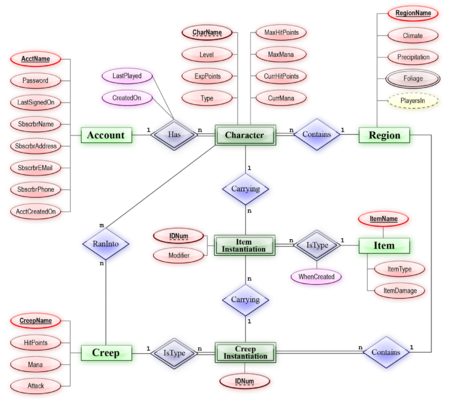
As noted above, there are 3 ingredients in a standard entity-relationship diagram:

Entities, which represent people, places, items, events, or concepts.

Attributes, which represent properties or descriptive qualities of an entity. These are also known as data elements.

Relationships, which represent the link between different entities.

Entities, attributes, and relationships can be represented in one of three ways: with a conceptual model, logical model, or physical model. These models increase in complexity as you move from conceptual to logical to physical. It's usually best to start with a conceptual ERD model, so you can understand—at the highest level—the entities in your data and how they relate to each other. As you transform a conceptual ERD to a physical model, you'll learn exactly how to implement modeled information into the database of your choice.



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

Entity is the real world object with has unique attributes. An entity is an object that exists and is distinguishable from other objects. For instance, John Harris with S.I.N. 890-12-3456 is an entity, as he can be uniquely identified as one particular person in the universe.An entity may be concrete (a person or a book, for example) or abstract (like a holiday or a concept). An entity set is a set of entities of the same type (e.g., all persons having an account at a bank).

An entity is represented by a set of attributes:

* E.g. name, S.I.N., street, city for ``customer'' entity.
* The domain of the attribute is the set of permitted values (e.g. the telephone number must be seven positive integers).

1. **Explain the different between an entity class and an entity instance.**
2. **Define attribute and its types.**

Attributes are the descriptive properties of an object which describes it to be represented as an unique singularity.

The types of attributes are as follows:

Simple and composite

-Simple attribute is the attribute which cannot be sub divided further.

-Composite attributes are the attributes which can be divided further. For example:Address as an attribute can be divided into HouseNo, StreetNo.

Singled valued and multivalued

-single valued attributes are those attributes which take in only single value in its input parameter.

-multivalued attributes are those attributes which take in multiple values as an input parameter. For example: phoneNo, familyMembersName etc.

Stored and derived attributes

-Stored attributes are those attributes which are stored so that it can be derived in other attributes in future.

-Derived attributes are those attributes which are derived from other attributes. For example: the attribute birthday can be derived from the dateOfBirth attribute.

1. **What is derived attributes?**

Derived attributes are those attributes which are derived from other attributes. For example: the attribute birthday can be derived from the dateOfBirth attribute.

Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average\_salary in a department should not be saved directly in the database, instead it can be derived. For another example, age can be derived from data\_of\_birth.

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

The association among entities is called a relationship. For example, an employee works\_at a department, a student enrolls in a course. Here, Works\_at and Enrolls are called relationships.

1. **Explain the difference between a relationship class and a relationship instance.**
2. **Define degree of relationship.**

The number of participating entities in a relationship defines the degree of the relationship.

Binary = degree 2

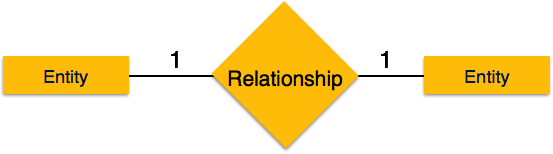
Ternary = degree 3

n-ary = degree

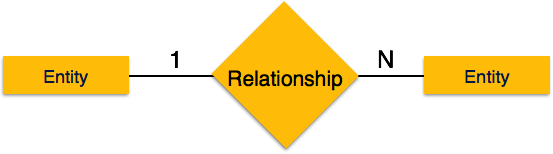
1. **List and give an example of the three types of binary relationships. Draw an E-R diagram for each**

A relationship where two entities are participating is called a **binary relationship**. Cardinality is the number of instance of an entity from a relation that can be associated with the relation.

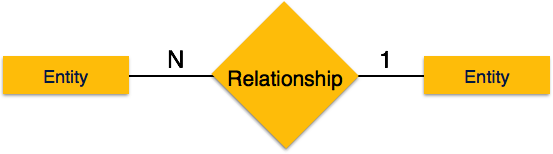
* **One-to-one** − when only one instance of an entity is associated with the relationship, it is marked as '1:1'. The following image reflects that only one instance of each entity should be associated with the relationship. It depicts one-to-one relationship.



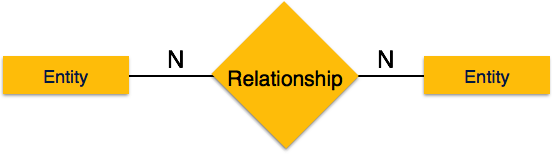
* **One-to-many** − When more than one instance of an entity is associated with a relationship, it is marked as '1:N'. The following image reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts one-to-many relationship.



* **Many-to-one** − When more than one instance of entity is associated with the relationship, it is marked as 'N:1'. The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts many-to-one relationship.



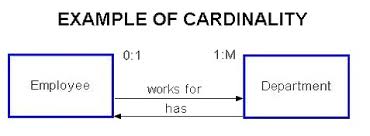
* **Many-to-many** − The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts many-to-many relationship.



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

The minimum number of instances of one entity that may associated with each instance of another entity is known as minimum cardinality.

The maximum number of instances of one entity that may associated with each instance of another entity is known as maximum cardinality.



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

Super Keys : Super key stands for superset of a key. A Super Key is a set of one or more attributes that are taken collectively and can identify all other attributes uniquely.

Candidate Keys: Candidate Keys are super keys for which no proper subset is a super key. In other words candidate keys are minimal super keys.

Primary Key: It is a candidate key that is chosen by the database designer to identify entities with in an entity set. Primary key is the minimal super keys. In the ER diagram primary key is represented by underlining the primary key attribute. Ideally a primary key is composed of only a single attribute. But it is possible to have a primary key composed of more than one attribute.

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

The Entity-Relationship model is a top-down approach to design database that is based on uniquely identifiable object. If begins by identifying things that are uniquely distinguishable called entities and relationships among these entities.

The main building modules of the Entity-Relationship model are:

a. Entities

b. Relationships

c. Attributes

Entities

An Entity is a basic object of ER-model which is an object in real world that can be distinguishable and can exists independently.

Relationships

Relationship defines the association among two entities. Suppose, consider student and a class are the two entities. These entities are associated as “student studies in class”. Hence studies is a relationship between the two entities, student and class.

Attributes

The properties of the entities are called attributes.

For example if we consider a mobile phone as an entity then each mobile well have its own color, design, model company. All these are the attributes of the mobile entity.

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

Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first\_name and last\_name. It is used when we need to sub divide the attributes to give more detail and concise information about an object.

**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.**

**20. What do you mean by mapping cardinalities ?explain various type of cardinalities.**

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

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

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

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

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

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

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

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

**29. Write short notes on:**

**Specialization**

**Generalization**

**Aggregation**