ST. XAVIER'S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**DATABASE MANAGEMENT SYSTEM**

**LAB ASSIGNMENT # 5**

**SUBMITTED BY:**

Dikita Tuladhar  
013BSCCSIT018

2nd Year/4th Sem

**SUBMITTED TO:**

|  |  |
| --- | --- |
| **Er. Sanjay Kumar Yadav**  Supervisor,  Department of Computer Science  St. Xavier’s College |  |

**Date of Submission:** 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](http://searchdatamanagement.techtarget.com/definition/data-modeling) technique that can help define business processes and can be used as the foundation for a [relational database](http://searchsqlserver.techtarget.com/definition/relational-database).

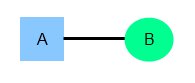
An entity relationship model, also called an entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of [data](http://www.webopedia.com/TERM/D/data.html) within [databases](http://www.webopedia.com/TERM/D/database.html) or information systems. An entity is a piece of data-an [object](http://www.webopedia.com/TERM/O/object.html)or concept about which data is stored.

Relationships between Entities

A relationship is how the data is shared between entities. There are three types of relationships between entities:

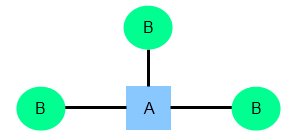
1. One-to-One

One instance of an entity (A) is associated with one other instance of another entity (B). For example, in a database of employees, each employee name (A) is associated with only one social security number (B).



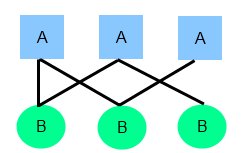
2. One-to-Many

One instance of an entity (A) is associated with zero, one or many instances of another entity (B), but for one instance of entity B there is only one instance of entity A. For example, for a company with all employees working in one building, the building name (A) is associated with many different employees (B), but those employees all share the same singular association with entity A.



3. Many-to-Many

One instance of an entity (A) is associated with one, zero or many instances of another entity (B), and one instance of entity B is associated with one, zero or many instances of entity A. For example, for a company in which all of its employees work on multiple projects, each instance of an employee (A) is associated with many instances of a project (B), and at the same time, each instance of a project (B) has multiple employees (A) associated with it.



1. Define entity and give an example.

In an [entity-relationship model](http://dictionary.reference.com/browse/entity-relationship%20model), an entity is a type of thing being modeled such as "person" or "product". Different entities have different sets of attributes such as "name" or "price" and are connected via relationships like "bought". Entities are closely related to classes (class).

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

A class is a set of entities, which are called the instances of the class. An entity can be an instance of many classes, which are called its types, and a class can be a type of many classes. A frame denoting a class is called a class frame, and a frame denoting an entity that is an instance of a class is called an instance frame. A class can also be an instance, i.e., an instance of a class of classes (a metaclass).

 A class CSUB is a sub of class CSUPER iff all instances of CSUB are also instances of CSUPER . In other words, all instances of the subclass are instances of the superclass, and the superclass may have other instances.

Classes have descriptive names, as do instances of entities. Instances of relationships have mechanically-generated IDs.

1. Define attribute and its types.

[Attributes](http://databasemanagement.wikia.com/wiki/Attributes?action=edit&redlink=1) are, simply put, the characteristics of entities. Some entities can have many attributes while others may only have a couple. As well, there are five categories that attributes are classified in. This simple table will be used to explain how each attribute can be a different type of attribute:

Student (stu\_LastName, stu\_MiddleName, stu\_FirstName, stu\_Age, stu\_Phone, stu\_Email)

1. Stored and Derived Attributes

The last category that attributes can be defined is called a derived attribute, where one attribute is calculated from another attribute. The derived attribute may not be stored in the database but rather calculated using algorithm.

**Example**:

In the entity **Student**, **stu\_Age** would be considered a derived attribute since it could be calculated using the student's date of birth with the current date to find their age.﻿﻿

Examples of derived attributes are: salary and age.

1. Single-valued and Multi-valued Attributes

Attributes can be classified as single or multi-value. The single-value attribute can only have one value, while the multi-valued attributes usually can store multiple data in them.

**Example**:

In the entity **Student**, **stu\_Address** could be considered a multi-value attribute since a student could have multiple addresses where he lives at. An example of a single-value attribute would be **stu\_LastName**since a student usually has one last name that uniquely identifies him/her.

1. Simple and Composite Attributes

Attributes can be classified as having many parts to them or just a single unbreakable attribute. The composite attribute is an attribute that can be subdivided into other single attributes with meanings of their own. A single attribute is just an attribute that cannot be subdivided into parts.

**Example**:

Imagine from the entity **Student**that instead of having the three attributes: **stu\_LastName, stu\_MiddleName, stu\_FirstName**it had one attribute called **stu\_Name**. The attribute **stu\_Name** would be considered a composite attribute since it can be subdivided into the other three attributes: **stu\_LastName, stu\_MiddleName, stu\_FirstName**. The rest of attributes would be consider single attributes since they can't be subdivided into parts.

1. What is derived attributes?

An attribute that’s value is derived from a stored attribute. The last category that attributes can be defined is called a derived attribute, where one attribute is calculated from another attribute. The derived attribute may not be stored in the database but rather calculated using algorithm.

Example: age, and it’s value is derived from the stored attribute Date of Birth.

1. Define relationship and give an example.

A relationship, in the context of databases, is a situation that exists between two relational database tables when one table has a foreign key that references the primary key of the other table. Relationships allow relational databases to split and store data in different tables, while linking disparate data items.

* One-to-one

One to one is implemented using single table by establishing relationship between same types of columns in a table.

* One-to-many

Implemented using two tables with primary key and foreign key relationships.

* Many-to-many

Implemented using a junction table.

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

The degree of relationship (also known as cardinality) is the number of occurrences in one entity which are associated (or linked) to the number of occurrences in another. Degree of relationship refers to the number of participating entities in a relationship. If there are two entities involved in relationship then it is referred to as binary relationship. If there are three entities involved then it is called as ternary relationship and so on.

There are three degrees of relationship, known as:

* one-to-one (1:1)
* one-to-many (1:M)
* many-to-many (M:N)

The latter one is correct, it is M:N and not M:M.

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

for each.

The three types of binary relationships are as follows:

1:1 - a single entity instance of one type is related to a single-entity instance of another type.

1:N - a single entity instance of one type is related to many-entity instances of another type.

M:N - many-entity instances of one type relate to many-entity instances of another type.

1. Define the terms maximum cardinality and minimum cardinality.
2. Explain the distinctions among the terms primary key, candidate key and super key.
3. What are the main building modules of the entity relationship model? Discuss each one.
4. What is composite attributes, when it is used?
5. Explain the difference between single-value attributes and simple attributes.
6. Discuss the difference between a composite key and a composite attribute. How would

each indicated in an E-R diagram?

1. What two courses of action are available to a designer when a multivalve attribute is

encountered?

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

model?

1. Explain the concept of dependent entities? Give example.
2. What is the difference total and partial participation? Explain.
3. What do you mean by mapping cardinalities? Explain various type of cardinalities.
4. What is the difference between single-value and multivalve attributes? Explain.
5. Explain the concept of participation constraints.
6. Difference the binary relationship with ternary relationship with example.
7. Explain the difference between weak and strong entity set.
8. Define the components of extended E-R features.
9. Define the concept of aggregation. Give two examples of where this concept is useful.
10. Explain the distinction between disjoint and overlapping constraints.
11. Explain the distinction between total and partial constraints.
12. Write short notes on:

· Specialization

· Generalization

· Aggregation