**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Database Management System**

**Lab Assignment #5**

**Submitted by:**

Divya Jyoti Pokharel

013BSCCSIT019

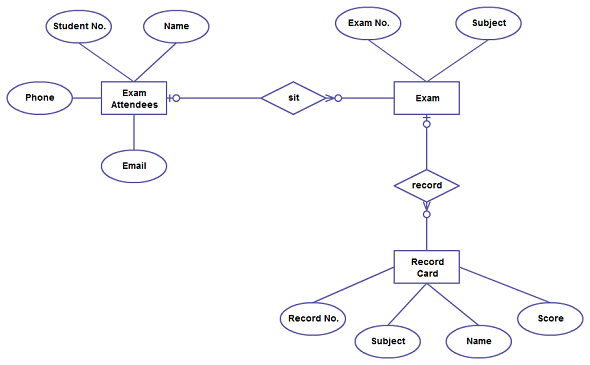
**Submitted to:**

|  |  |
| --- | --- |
| **Er. Sanjay Kumar Yadav**  Lecturer  St. Xavier’s College |  |

**Date of Submission:** August 27, 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.



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

In general, an entity is an existing or real thing. An entity is a single person, place, or thing about which data can be stored. A person, partnership, organization, or business that has a legal and separately identifiable existence.

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

Attributes is characteristics of an entity.

In Entity Relationship(ER) Model attributes can be classified into the following types.

* Simple and Composite Attribute
* Single Valued and Multi Valued attribute
* Stored and Derived Attributes
* Complex Attribute

Simple attribute  consists of a single atomic value. A simple attribute cannot be subdivided. For example the attributes age, sex etc are simple attributes.

A composite attribute is an attribute that can be further subdivided. For example the attribute ADDRESS can be subdivided into street, city, state, and zip code.

A single valued attribute can have only a single value. For example a person can have only one 'date of birth', 'age' etc. That is a single valued attributes can have only single value. But it can be simple or composite attribute. That is 'date of birth' is a composite attribute, 'age' is a simple attribute. But both are single valued attributes.

Multivalve attributes can have multiple values. For instance a person may have multiple phone numbers, multiple degrees etc. Multivalve attributes are shown by a double line connecting to the entity in the ER diagram.

The value for the derived attribute is derived from the stored attribute. For example 'Date of birth' of a person is a stored attribute. The value for the attribute 'AGE' can be derived by subtracting the 'Date of Birth'(DOB) from the current date. Stored attribute supplies a value to the related attribute.

Stored Attribute: An attribute that supplies a value to the related attribute.

Example: Date of Birth

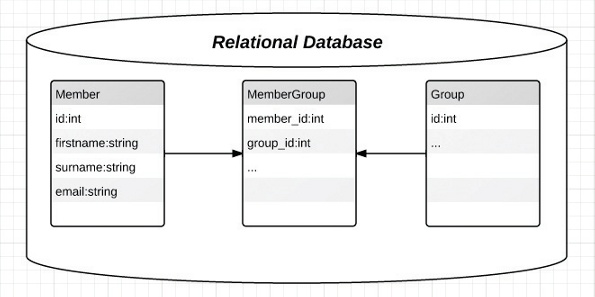
Derived Attribute: An attribute that’s value is derived from a stored attribute.

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

A complex attribute that is both composite and multi valued.

1. **What is derived attributes?**
2. **Define relationship and give an example.**

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.



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

A relationship's degree indicates the number of associated entities or participants.

* Unary Relationship
* Binary Relationship
* Ternary Relationship

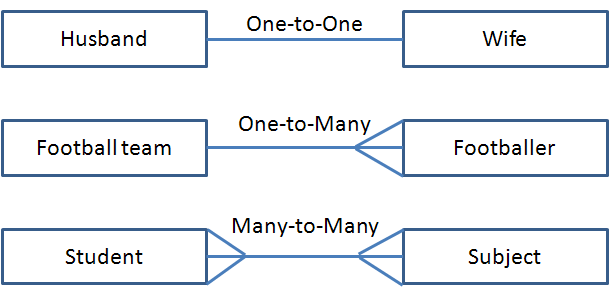
A unary relationship exists when an association is maintained with in a single entity. A binary relationship exists when two entities are associated. A ternary relationship exists when three entities are associated.

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

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.**

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.

Example: Consider a Relation or Table R1. Let A,B,C,D,E are the attributes of this relation.

R(A,B,C,D,E)

A→BCDE This means the attribute 'A' uniquely determines the other attributes B,C,D,E.

BC→ADE This means the attributes 'BC' jointly determines all the other attributes A,D,E in the relation.

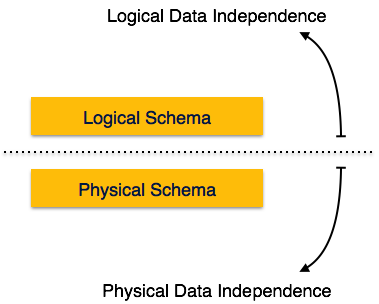
Primary Key :A

Candidate Keys :A, BC

Super Keys : A,BC,ABC,AD

1. **What are the main building modules of the entity relationship model? Discuss each one.**
2. **What is composite attributes, when it is used?**
3. **Explain the difference between single-value attributes and simple attributes.**
4. **Discuss the difference between a composite key and a composite attribute. How would each indicated in an E-R diagram?**
5. **What two courses of action are available to a designer when a multivalued attribute is encountered ?**
6. **Explain the various terms of an E-R model and how are they represented in an E-R model?**
7. **Explain the concept of dependent entities? Give example.**

A database system normally contains a lot of data in addition to users’ data. For example, it stores data about data, known as metadata, to locate and retrieve data easily. It is rather difficult to modify or update a set of metadata once it is stored in the database. But as a DBMS expands, it needs to change over time to satisfy the requirements of the users. If the entire data is dependent, it would become a tedious and highly complex job.



Metadata itself follows a layered architecture, so that when we change data at one layer, it does not affect the data at another level. This data is independent but mapped to each other.

Logical Data Independence

Logical data is data about database, that is, it stores information about how data is managed inside. For example, a table (relation) stored in the database and all its constraints, applied on that relation. Logical data independence is a kind of mechanism, which liberalizes itself from actual data stored on the disk. If we do some changes on table format, it should not change the data residing on the disk.

Physical Data Independence

All the schemas are logical, and the actual data is stored in bit format on the disk. Physical data independence is the power to change the physical data without impacting the schema or logical data. For example, in case we want to change or upgrade the storage system itself − suppose we want to replace hard-disks with SSD − it should not have any impact on the logical data or schemas.

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

* Specialization
* Generalization
* Aggregation