**ST.XAVIER’S COLLEGE**

MAITIGHAR, KATHMANDU



**Database Management System**

Assignment #5

Submitted By:

Aashish Raj Shrestha

013BSCCSIT002

2nd year/ 4th semester

Submitted to:

|  |  |
| --- | --- |
| Er. Sanjay Kumar Yadav  Lecturer  Department of Computer Science |  |

Entity Relationship Model

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

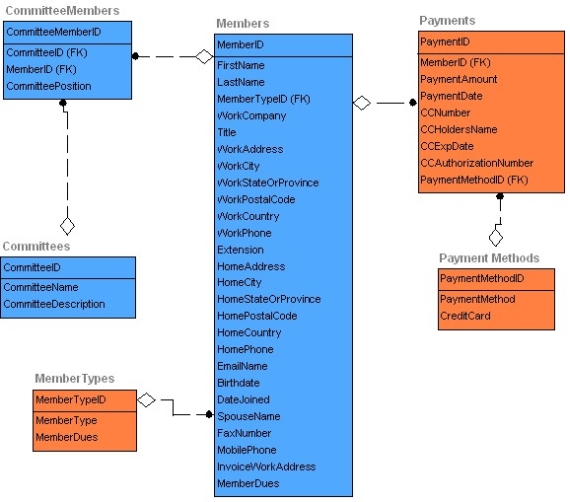
An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system’s entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

The elements of an ERD are:

* **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.

Steps involved in creating an ERD include:

1. Identifying and defining the entities
2. Determining all interactions between the entities
3. Analyzing the nature of interactions/determining the cardinality of the relationships
4. Creating the ERD [1][2]



*Figure I: ER- Diagram made with DeZign for Databases [3]*

1. Define entity and give an example.

In general, an entity is an existing or real thing. An entity exists and that's all it needs to do to be an entity. The fact that something exists also seems to connote separateness from other existences or entities. In programming, engineering, and probably many other contexts, the word is used to identify units, whether concrete things or abstract ideas, that have no ready name or label.

In relation to a database, an entity is a single person, place, or thing about which data can be stored. An entity is something of importance to the user. Something the user wants to track. Example: VEHICLE.

In data modeling (a first step in the creation of a database), an entity is some unit of data that can be classified and have stated relationships to other entities. [4] Example: PERSON’S NAME

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

An entity is a person, place, event, or thing about which data is collected. The entity class is described by the structure of the entities in that class. An instance is an occurrence of an entity. An entity instance is a representation of a particular entity within the entity class. The entity instance is described by the values of the attributes of the entity. An entity class, then, is a collection containing many entity instances.

An entity class is a group of entities of the same type, i.e. VEHICLE.

An entity instance is a particular entity, i.e. VEHICLE 12345 [5][6].

1. Define attribute and its types.

An attribute of an entity is a particular property that describes the entity. A relationship is the association that describes the interaction between entities. Cardinality, in the context of ERD, is the number of instances of one entity that can, or must, be associated with each instance of another entity. In general, there may be one-to-one, one-to-many, or many-to-many relationships [1].

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

1. Simple and Composite Attribute
2. Single Valued and Multi Valued attribute
3. Stored and Derived Attributes
4. Complex Attribute
5. **Simple and Composite 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.

Simple Attribute: Attribute that consist of a single atomic value.  
Example: Salary, age etc

Composite Attribute  : Attribute value not atomic.  
Example :   Address  :  ‘House\_no:City:State  
                  Name      :  ‘First Name: Middle Name: Last Name’ 

1. **Single Valued and Multi Valued attribute**

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.

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

Single Valued Attribute: Attribute that hold a single value  
Example1: Age  
Exampe2: City  
Example3:Customer id

Multi Valued Attribute: Attribute that hold multiple values.  
Example1: A customer can have multiple phone numbers, email id's etc  
Example2: A person may have several college degrees

1. **Stored and Derived Attributes**

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.

1. **Complex Attribute**

A complex attribute that is both composite and multi valued [7].

5. What is derived attributes?

Here 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 [8].

6. Define relationship and give an example.

A relationship is the association that describes the interaction between entities. Cardinality, in the context of ERD, is the number of instances of one entity that can, or must, be associated with each instance of another entity. In general, there may be one-to-one, one-to-many, or many-to-many relationships.

For example, let us consider two real-world entities, an employee and his department. An employee has attributes such as an employee number, name, department number, etc. Similarly, department number and name can be defined as attributes of a department. A department can interact with many employees, but an employee can belong to only one department, hence there can be a one-to-many relationship, defined between department and employee.

In the actual database, the employee table will have department number as a foreign key, referencing from department table, to enforce the relationship [1].

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

A relationship class is an association among entity classes.

A relationship instance is an association among entity instances.

8. Define degree of relationship.

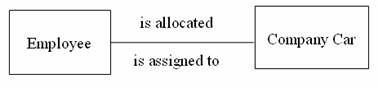
The degree of relationship (also known as cardinality) is the number of occurrences in one entity which are associated (or linked) to the number of occurrences in another.

There are three degrees of relationship, known as:

1. One-to-one (1:1)

This is where one occurrence of an entity relates to only one occurrence in another entity. A one-to-one relationship rarely exists in practice, but it can. However, you may consider combining them into one entity. For example, an employee is allocated a company car, which can only be driven by that employee.

Therefore, there is a one-to-one relationship between employee and company car.

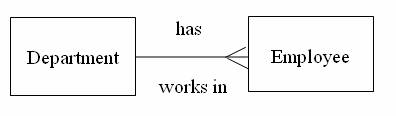


1. One-to-many (1:M)

Is where one occurrence in an entity relates to many occurrences in another entity?

For example, taking the employee and department entities shown on the previous page, an employee works in one department but a department has many employees.

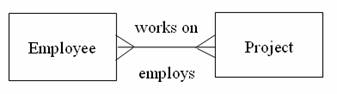
Therefore, there is a one-to-many relationship between department and employee.



1. Many-to-many (M:N)

This is where many occurrences in an entity relate to many occurrences in another entity. The normalization process discussed earlier would prevent any such relationships but the definition is included here for completeness. As with one-to-one relationships, many-to-many relationships rarely exist. Normally they occur because an entity has been missed. For example, an employee may work on several projects at the same time and a project has a team of many employees. Therefore, there is a many-to-many relationship between employee and project.

Entity Model



However, in the normalization process this many-to-many is resolved by the entity Project Team [9].

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

for each.

10. Define the terms maximum cardinality and minimum cardinality.

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

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

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

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

REFERENCES:

[1] Cory Janssen. ” Definition - What does Entity-Relationship Diagram (ERD) mean?”. *Entity-Relationship Diagram (ERD). Link:* https://www.techopedia.com/definition/1200/entity-relationship-diagram-erd

[2] “What is an ERD?”. Lucid Software Inc. 2015. Link: https://www.lucidchart.com/pages/what-is-ERD

[3] “What is an Entity Relationship Diagram (ERD)?”. Datanamic Solutions BV. Link: http://www.datanamic.com/support/lt-dez006-what-is-an-erd.html

[4] Margaret Rouse. “entity”. Link: http://whatis.techtarget.com/definition/entity

[5] hello67890. Quizlet. Link: https://quizlet.com/21468453/chapt-4-flash-cards/

[6] Willem Van Galen.“Difference between an entity class and an entity instance?”. Link: http://www.answers.com/Q/Difference\_between\_an\_entity\_class\_and\_an\_entity\_instance

[7] “Attribute Types”. JKiNFOLINE.2010.Link: http://www.jkinfoline.com/attribute-types.html

[8] “E/R Model: Type of Attributes”. *Database Management.* Link: http://databasemanagement.wikia.com/wiki/E/R\_Model:\_Type\_of\_Attributes

[9] “Degrees of Relationship (Cardinality)”. Link: http://www.sqa.org.uk/e-learning/SoftDevRDS02CD/page\_44.htm