ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Database Management System Assignment #5**

**Submitted by:**

Bishal Pandey  
013BSCCSIT016

**Submitted to:**

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

Entity Relationship model

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.

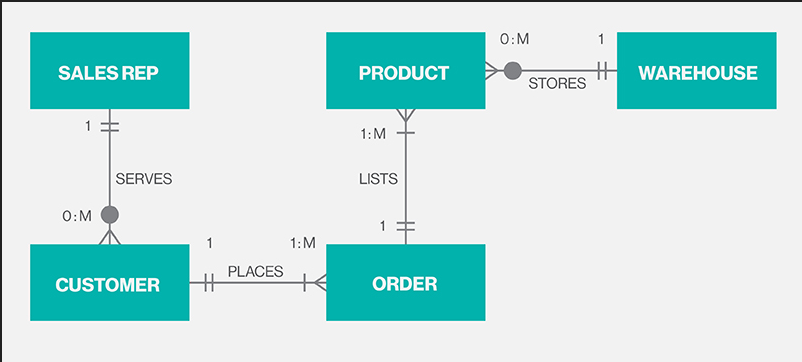


Fig. ER Diagram.

1. Define entity and give an example.

An entity is a “thing” or “object” in the real world that is distinguishable from another object. For example: • Specific customer , Particular course in university Entities can be described by a set of properties called attributes. For example: customer\_id, customer\_name, customer\_address are attributes for entity customer. Similarly, course\_id, course\_name are attributes for entity course.

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

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.

1. Define attribute and its types.

In simple, attribute is descriptive property of entity set. Set of attributes describes entity set.

For example

customer = (customer-id, customer-name, customer-city)

account=( account\_number, balance)

loan = (loan\_number, amount)

**Types of Attributes**

Simple and Composite attribute

Attribute which can not be divide into subparts (i.e. into other attributes) called simple attribute. For example, customer\_id in customer entity set is simple attribute, since it can not divide into sub attributes. Attribute that can further divide into subparts called composite attribute.

For example, customer\_name in customer entity set is composite attribute since it can be divided into sub attributes: customer\_fname, customer\_mname and customer\_lname. Composite attributes helps to group related attributes, which makes modeling clearer.

**Single-valued and Multivalued attributes**

Attribute that can take only one value in every entry called singled-valued attribute. For example, attribute customer\_name in customer entity set is single-valued attribute since it can not contain more than one customer name in any entry. An attribute that can take more than one values in any entry called multivalued attribute. For example, in a customer entity set attribute customer\_phonenumber is multivalued attribute since customer may have zero or one or several phone number.

**Stored and Derived attribute**

Attribute whose values can be derived from the values of other related attributes or entities called derived attribute. For example, in customer entity set, attribute age is derived attribute if customer entity set has attribute date\_of\_birth. We can derive age of customer from date\_of\_birth and current\_date. Here the attribute date\_of\_birth is stored attribute and the attribute age is derived attribute. The value of derived attribute is not stored, it is computed when required.

1. What is derived attributes?

Attribute whose values can be derived from the values of other related attributes or entities called derived attribute. For example, in customer entity set, attribute age is derived attribute if customer entity set has attribute date\_of\_birth. We can derive age of customer from date\_of\_birth and current\_date. Here the attribute date\_of\_birth is stored attribute and the attribute age is derived attribute. The value of derived attribute is not stored, it is computed when required.

1. Define relationship and give an example.

A relationship is an association among two or more entities. A relationship set is a set of relationship of same type.

Formally, if E1, E2, . . ,En (n≥2) are entity sets then a relationship set R is a subset of {(e1,e2, . . ,en)│e1∈E1,e2 ∈E2, . . en ∈En} where (e1,e2, . . ,en) is relationship.

Example: For two entity sets customer and account, we can define relationship set depositor which associates each customer to their corresponding account he/she has.

1. 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.

1. Define degree of relationship.

Degree is the number of entities that participate in a relationship. The relationship ASSIGNMENT associates a CLIENT with an ATTORNEY with a TASK.

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

1:1, ATTORNEY to COMPUTER

1:N, VEHICLE to REPAIR

N:M, VEHICLE to OWNER

1. Define the terms maximum cardinality and minimum cardinality.

Maximum cardinality is the maximum number of instances of an entity that can participate in an instance of a relationship. Minimum is the least number of instances of an entity that can participate in an instance of a relationship.

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

**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.

**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.

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

Attribute that can further divide into subparts called composite attribute. For example, customer\_name in customer entity set is composite attribute since it can be divided into sub attributes: customer\_fname, customer\_mname and customer\_lname. Composite attributes helps to group related attributes, which makes modeling clearer.

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

Attribute which can not be divide into subparts (i.e. into other attributes) called simple attribute whereas Attribute that can take only one value in every entry called singled-valued attribute.

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.

|  |  |
| --- | --- |
| Weak Entity | Strong Entity |
| The relationship between two strong entity set is represented by a double diamond symbol. | The relationship between two strong entity set is represented by a single diamond symbol. |
| The line connecting strong entity set with the relationship is double. | The line connecting strong entity set with the relationship is single. |
| Member of a weak entity set is a subordinate entity. | Member of a strong entity set is a dominant entity. |
| It is represented by a double rectangle. e primary key of a weak entity is found by taking the primary key of the strong entity on which it is existence-dependent, plus the discriminator of the weak entity set. | It has its own primary key and it is represented by a rectangle. |

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

* **Specialization**: It is a top-down approach in which one higher level entity can be broken down into two lower level entities.
* **Generalization:** Here the design process takes place in bottom up manner. Multiple entity sets are synthesized into a higher level entity set on the basis of common features.

Example: Employee and Customer entities can be synthesized into a higher level entity Person.

* **Attribute Inheritance:** The attributes of higher level entity set are inherited by lower level entity set.
* **Aggregation:** Aggregation is an abstraction in which relationship sets are treated as higher level entity sets. Here a relationship set is embedded inside an entity set, and these entity sets can participate in relationships.

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

Data aggregation is any process in which information is expressed in a summary form for purposes such as reporting or analysis. Ineffective data aggregation is currently a major component that limits query performance. And, with up to 90 percent of all reports containing aggregate information, it becomes clear why proactively implementing an aggregation solution can generate significant performance benefits, opening up the opportunity for companies to enhance their organizations’ analysis and reporting capabilities.

27. Explain the distinction between disjoint and overlapping constraints

**Disjoint:**

In a disjoint design constraint, an entity can be almost one of the subclasses of the specialization. i.e. Employee is disjoint to secretery,technician,engineer .

**Overlapping:**

On the other hand, overlapping gen are revert in the sense that the super class can belong to more than one subclass. for eg: person as a super class and passport, driving license as its overlapping subclass.

.

28. Explain the distinction between total and partial constraints.

**Total Constrains:**

* If an entity can exist, only if it participates in at least one relationship instance, then that is called total participation, meaning that every entity in one set, must be related to at least one entity in a designated entity set.
* An example would be the Employee and Department relationship. If company policy states that every employee must work for a department, then an employee can exist only if it participates in at least one relationship instance (i.e. an employee can’t exist without a department)
* It is also sometimes called an existence dependency.
* Total participation is represented by a double line, going from the relationship to the dependent entity.

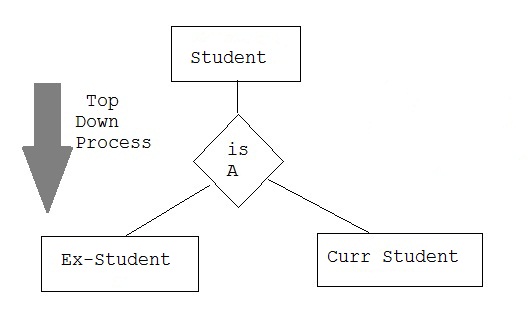
**Partial**:

* If only a part of the set of entities participate in a relationship, then it is called partial participation.
* Using the Company example, every employee will not be a manager of a department, so the participation of an employee in the “Manages” relationship is partial.
* Partial participation is represented by a single line.

29. Write short notes on:

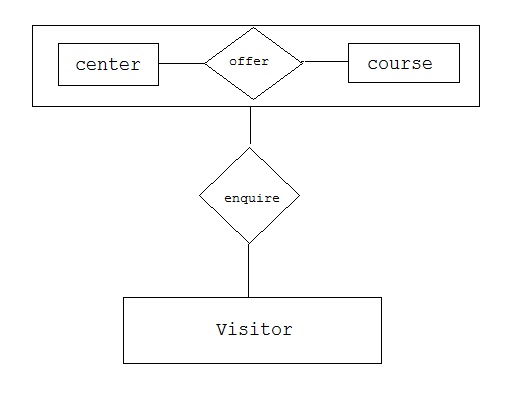
· **Specialization**

It is opposite to Generalization. It is a top-down approach in which one higher level entity can be broken down into two lower level entity. In specialization, some higher level entities may not have lower-level entity sets at all.



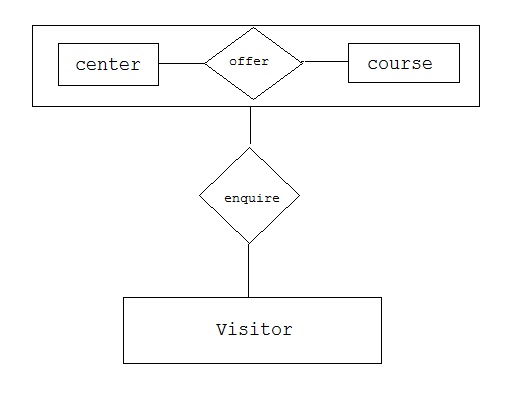
· **Generalization**

It is a bottom-up approach in which two lower level entities combine to form a higher level entity. In generalization, the higher level entity can also combine with other lower level entity to make further higher level entity.



**· Aggregation**

It is a process when relation between two entity is treated as a single entity. Here the relation between Center and Course, is acting as an Entity in relation with Visitor.

****