Unit-02

Database Design using ER Model

Data Model

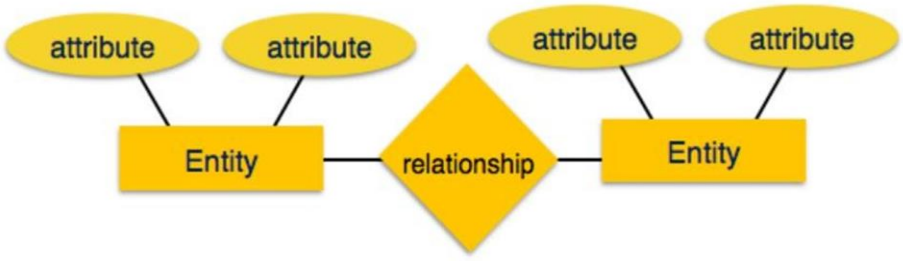
- ✓ A database model shows the logical structure of a database, including the relationships and constraints that determine how data can be stored and accessed.
- ✓ Data models define how the logical structure of a database is modeled.
- ✓ Data Models are fundamental entities to introduce abstraction in a DBMS.
- ✓ Data models define how data is connected to each other and how they are processed and stored inside the system.

Types of database models

1) Hierarchical database model
2) Relational model
3) Network model
4) Object-oriented database model
5) Entity-relationship model
6) Document model
7) Entity-attribute-value model
8) Star schema

Entity-Relationship (ER) model

- ➢ The Entity-Relationship (ER) model is a conceptual data model used in database design to represent the structure of a database in a visual and easy-to-understand manner. The ER Model creates entity set, relationship set, general attributes and constraints.



Entity Sets:

- ✓ An entity set is a collection of similar entities.
- ✓ An entity is a real-world object or concept that has a distinct identity and can be uniquely identified. For example, "Person," "Department," or "Product" can be entity sets.
- ✓ Each entity in the set has attributes that describe its properties. For instance, a "Person" entity may have attributes such as "Name," "Age," and "Address."
- ✓ In an ER diagram, entity sets are typically represented by rectangles. The name of the entity set is written inside the rectangle.

Relationship Sets:

- ✓ A relationship set is a collection of relationships of the same type.
- ✓ A relationship represents an association between two or more entities. It illustrates how entities are related to one another. For example, a "Works_In" relationship might associate employees with the departments they work in.
- ✓ Relationships can have attributes that provide additional information about the association. For instance, a "Works In" relationship might have an attribute like "StartDate" to indicate when an employee started working in a particular department.
- ✓ In an ER diagram, relationship sets are represented by diamonds connecting the participating entity sets.

Key components of the ER model include:

1. **Entity**

- ✓ An entity is a real-world object or concept that exists independently and has a distinct identity.
- ✓ Examples of entities include a person, place, event, or concept.

2. **Attribute:**

- ✓ An attribute is a property or characteristic of an entity.
- ✓ For example, for an entity "Person," attributes might include "Name," "Age," and "Address."

3. **Relationship:**

- ✓ A relationship represents an association between two or more entities.
- ✓ For example, a relationship between "Person" and "Address" might represent the fact that a person has a particular address.

4. **Key:**

- ✓ A key is an attribute or set of attributes that uniquely identifies an entity within an entity set.
- ✓ Primary key: A key that uniquely identifies an entity in a table.
- ✓ Foreign key: A key that refers to the primary key in another table, establishing a link between the two tables.

5. **Cardinality:**

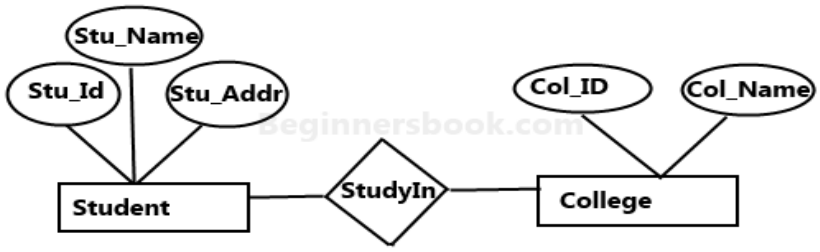
- ✓ Cardinality defines the number of instances of one entity that can be associated with the number of instances of another entity.
- ✓ Common cardinalities include one-to-one, one-to-many, and many-to-many relationships.

6. **Weak Entity:**

- ✓ A weak entity is an entity that cannot be uniquely identified by its attributes alone and relies on a related entity for identification.
- ✓ It is typically identified by a partial key, which is a set of attributes that identifies the weak entity in conjunction with the related entity.

7. **Generalization and Specialization:**

- ✓ Generalization is the process of defining a more general entity type from a set of more specialized entity types.
- ✓ Specialization is the reverse process, where a more specialized entity type is defined from a more general entity type.



Sample E-R Diagram

Importance of ER model:

1. **Clarity and Simplicity**

→ The ER model provides a clear and simple representation of the essential components of a database: entities, relationships, and attributes. This simplicity makes it easy for both technical and non-technical stakeholders to understand the structure of the data.

2. **Visualization**

→ The graphical representation of the ER model, through ER diagrams, allows for a visual representation of the data model. This visualization aids in communication among stakeholders, helping them to grasp the relationships and structure of the data.

3. **Database Design**

→ The ER model serves as a blueprint for designing the structure of a relational database. It helps in identifying entities, attributes, relationships, and constraints, which are essential for creating a well-organized and efficient database schema.

4. **Communication**

→ The ER model provides a common language for communication between different stakeholders involved in the database design process. It bridges the communication gap between business analysts, system designers, and database administrators.

5. **Requirements Analysis**

→ During the initial stages of database design, the ER model is used for requirements analysis. It helps in identifying the entities and relationships that need to be represented in the database, ensuring that the database meets the information needs of the organization.

6. **Normalization**

→ The ER model assists in the normalization process, which is the systematic organization of data to reduce redundancy and improve data integrity. By identifying entities and their relationships, the model helps in achieving a normalized database schema.

7. **Data Integrity and Consistency**

→ Through the definition of relationships, keys, and constraints, the ER model helps ensure data integrity and consistency within the database. It allows for the specification of rules that govern the relationships between entities and maintain the accuracy of the data.

8. **Database Maintenance**

→ The ER model provides a foundation for database maintenance and evolution. As business requirements change, the model can be modified to reflect these changes, and the database schema can be updated accordingly.

9. **Database Documentation**

→ ER diagrams serve as valuable documentation for database systems. They provide a visual reference for understanding the database structure, making it easier for developers and administrators to maintain and enhance the database over time.

10. Tool Integration

→ Many database design tools support the ER model, allowing designers to create, modify, and visualize the database schema. These tools often generate SQL scripts or database schema diagrams directly from the ER model.

Types of DBMS Entities:

a. Strong Entity

→ The strong entity has a primary key. Weak entities are dependent on strong entities. Its existence is not dependent on any other entity. Strong Entity is represented by a single 'rectangle'.

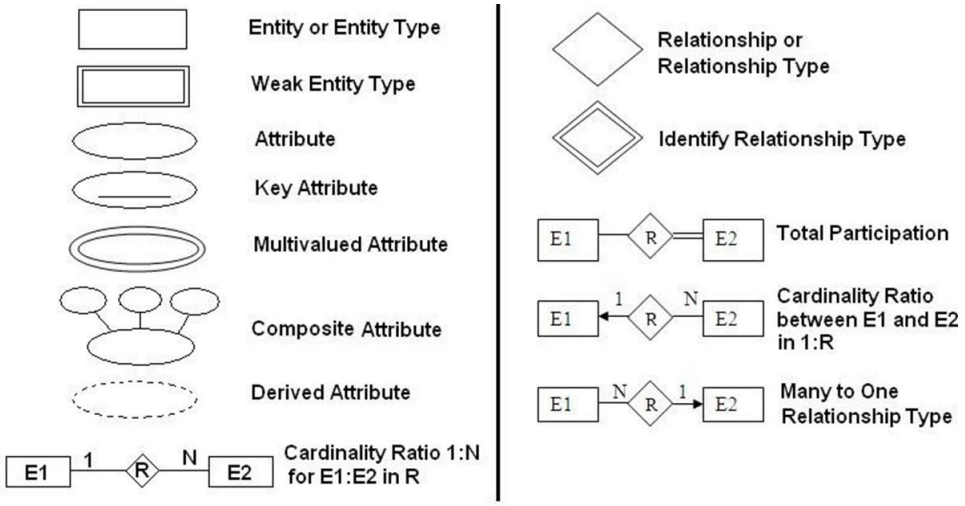
b. Weak Entity

→ The weak entity in DBMS do not have a primary key and are dependent on the parent entity. It mainly depends on other entities. Weak Entity is represented by 'double rectangle'.

Attributes and its types:

→ In the context of databases and data modeling, an attribute is a property or characteristic of an entity, relationship, or other data element. Attributes provide additional information about the data and help describe the entities within a database. There are various types of attributes, each serving a specific purpose. Here are common types of attributes:

1. **Composite Attribute:**
   - ✓ A composite attribute is composed of multiple simple attributes, each with its own meaning. It allows for a more detailed representation of a property.
   - ✓ Example: An address attribute might be composed of "Street," "City," and "Zip Code."
2. **Derived Attribute:**
   - ✓ A derived attribute is one whose value can be derived or calculated from other attributes in the database. It is not stored explicitly but is computed when needed.
   - ✓ Example: The "Age" attribute of a person can be derived from the "Date of Birth."
3. **Single-valued Attribute:**
   - ✓ A single-valued attribute can hold only a single value for a particular entity.
   - ✓ Example: The "Date of Birth" attribute for a person is typically single-valued.
4. **Multi-valued Attribute:**
   - ✓ A multi-valued attribute can hold multiple values for a single entity. It represents a set of values, rather than a single atomic value.
   - ✓ Example: In a "Person" entity, the attribute "Phone Numbers" can be multi-valued.
5. **Null Attribute:**
   - ✓ A null attribute is one that may not have a value for a particular entity. It represents the absence of data.
   - ✓ Example: If an employee's "Middle Name" is not available, it can be represented as a null attribute.



Mapping Cardinalities

→ Mapping cardinalities in the context of the Entity-Relationship (ER) model describe the number of entities to which another entity can be associated through a relationship. Mapping cardinalities define the relationships between entities and help to specify the constraints on those relationships. There are three main types of mapping cardinalities: one-to-one (1:1), one-to-many (1:N), and many-to-many (N:M).

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

✓ In a one-to-one relationship, each entity in the first set is related to at most one entity in the second set, and vice versa.

✓ Example: Consider a relationship between "Person" and "Passport," where each person has only one passport, and each passport is associated with only one person.

2. **One-to-Many (1:N):**

✓ In a one-to-many relationship, each entity in the first set can be related to multiple entities in the second set, but each entity in the second set is related to at most one entity in the first set.

✓ Example: A relationship between "Department" and "Employee," where each department can have multiple employees, but each employee belongs to only one department.

3. **Many-to-Many (N:M):**

✓ In a many-to-many relationship, each entity in the first set can be related to multiple entities in the second set, and vice versa.

✓ Example: A relationship between "Student" and "Course," where each student can enroll in multiple courses, and each course can have multiple students.

Constrains on ER Model

→ In the Entity-Relationship (ER) model, constraints are rules and conditions that define the valid and permissible states of the database. These constraints help maintain the integrity and accuracy of the data within the database. There are several types of constraints in the ER model:

1. **Entity Integrity Constraint**

→ Ensures that each entity in the entity set has a unique and non-null identifier, typically represented by a primary key.

2. **Referential Integrity Constraint**

→ Governs the relationships between entities and ensures that references between entities are valid.

3. **Key Constraints**

→ Specifies the attributes or combinations of attributes that form the primary key or candidate key of an entity.

4. **Domain Constraints**

→ Defines the allowable values for attributes.

→ Specifies the range of valid values, data types, and formats for each attribute.

5. **Cardinality Constraints**

→ Determine the number of entities that can be associated with one another in a relationship.

6. **Participation Constraints**

→ Specify whether each entity in an entity set must participate in a relationship.

7. **Attribute Constraints**

→ Define rules and conditions for individual attributes.

8. **Generalization Constraints**

→ Apply to the generalization (inheritance) hierarchy and define rules for how entities are related in a hierarchy.

9. **Multiplicity Constraints**

→ Specify the minimum and maximum number of instances that can participate in a relationship.

10. **Temporal Constraints**

→ Deal with time-related aspects, specifying conditions related to the validity period of data in the database.

E-ER diagram

→ EER diagram is a visual representation of data, based on the E-ER model that is an extension of the original entity relationship (ER) model.

E-ER

- Today the complexity of the data is increasing so it becomes more and more difficult to use the traditional ER model for database modeling. To reduce this complexity of modeling we have to make improvements or enhancements to the existing ER model to make it able to handle the complex application in a better way.
- Enhanced entity-relationship diagrams are advanced database diagrams very similar to regular ER diagrams which represent the requirements and complexities of complex databases.
- It is a diagrammatic technique for displaying the Sub Class and Super Class; Specialization and Generalization; Union or Category; Aggregation etc.

Key feature of E-ER

i) Sub Class and Super Class
ii) Generalization and Specialization
iii) Inheritance
iv) union

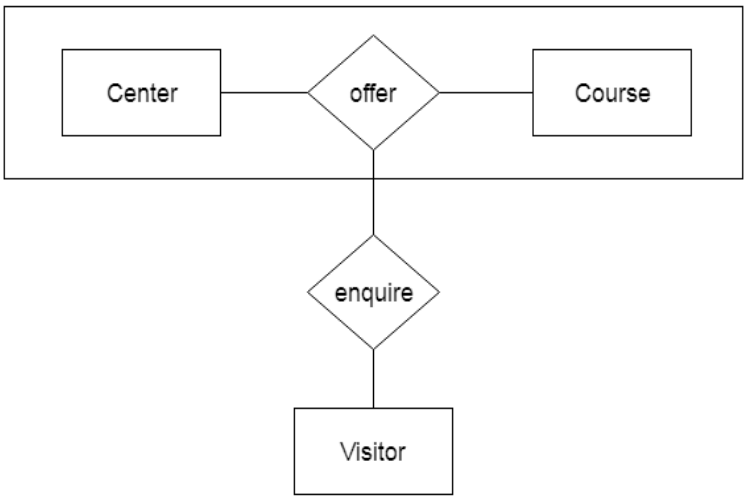
Extended ER Features: Aggregation, Specialization and Generalization, Constraints on Specialization/Generalization

→ Extended Entity-Relationship (ER) features, including aggregation, specialization, and generalization, enhance the expressiveness and modeling capabilities of the ER model.

Additionally, constraints on specialization/generalization further refine the structure and behavior of the model. Let's explore each of these concepts:

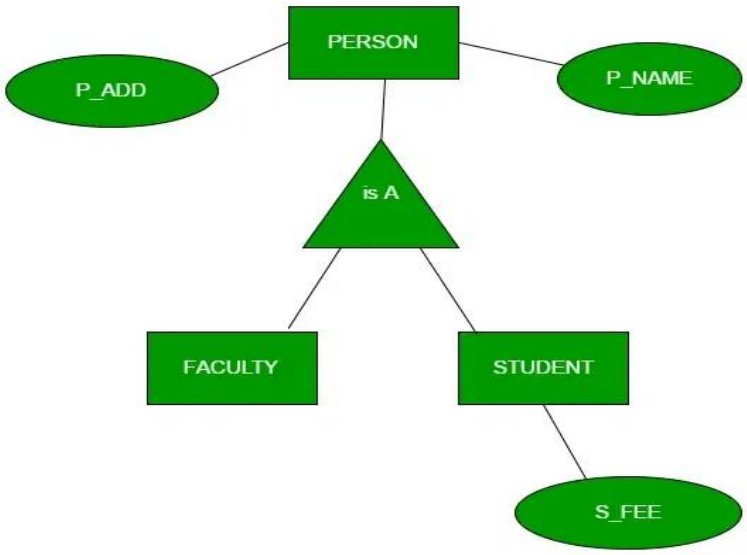
1. Aggregation

- ✓ Aggregation is a modeling concept that allows you to treat a relationship between entities as an entity itself.
- ✓ It is represented by a diamond shape in an ER diagram.



2. Specialization and Generalization

- ✓ **Specialization:** The process of defining a subset of entities from a super-entity based on specific characteristics.
- ✓ **Generalization:** The reverse process, where common attributes and relationships are generalized into a super-entity.



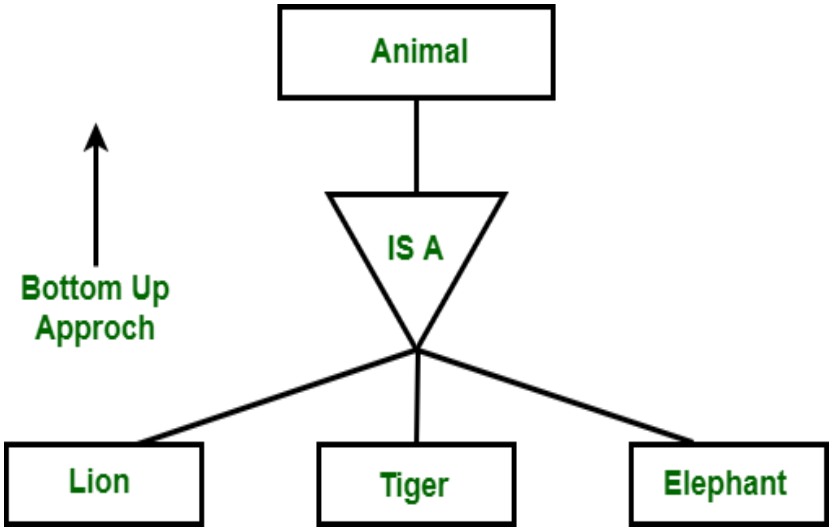
3. Constraints on Specialization/Generalization

Disjointness Constraint:

- ✓ Specifies whether subtypes are mutually exclusive (disjoint) or can overlap (overlap).
- ✓ **Disjoint:** An entity can belong to only one subtype.
- ✓ **Overlap:** An entity can belong to multiple subtypes.

Completeness Constraint:

- ✓ Specifies whether an entity must belong to at least one subtype (total) or can choose not to belong to any (partial).
- ✓ **Total:** Every entity must be a member of at least one subtype.
- ✓ **Partial:** An entity can choose not to belong to any subtype.

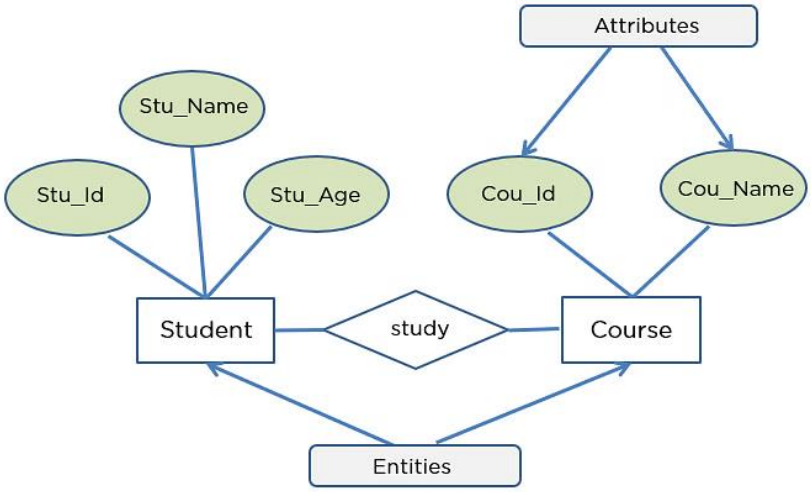


These extended ER features and constraints enhance the modeling capabilities of the traditional ER model, allowing for more nuanced and accurate representations of real-world scenarios in a database.

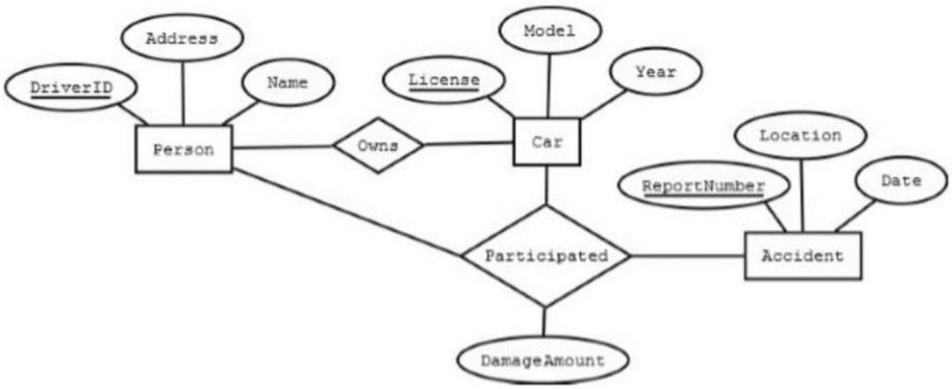
Conversion of ER diagram into relational table

The basic rules for converting the ER diagrams into tables are-

- → Convert all the Entities in the diagram to tables.
- → All the entities represented in the rectangular box in the ER diagram become independent tables in the database.
- → All single valued attributes of an entity is converted to a column of the table.
- → All the attributes, whose value at any instance of time is unique, are considered as columns of that table.
- → Key attribute in the ER diagram becomes the Primary key of the table.
- → Declare the foreign key column, if applicable.
- → Any multi-valued attributes are converted into a new table.
- → Any composite attributes are merged into the same table as different columns.
- → One can ignore the derived attribute, since it can be calculated at any time.

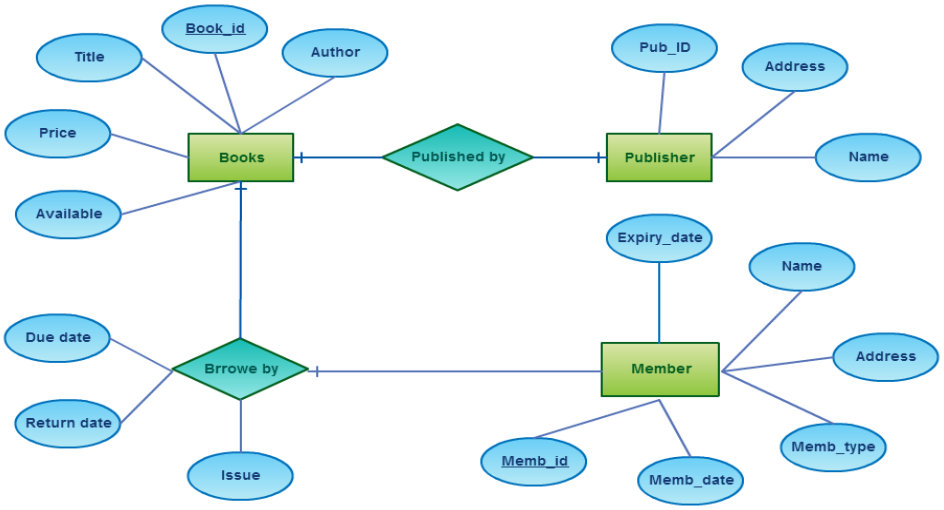


1. **Construct an ER diagram for car insurance company that has a set of customers each of whom owns one or more cars. Each car has associated with it zero to any number of recorded accidents**

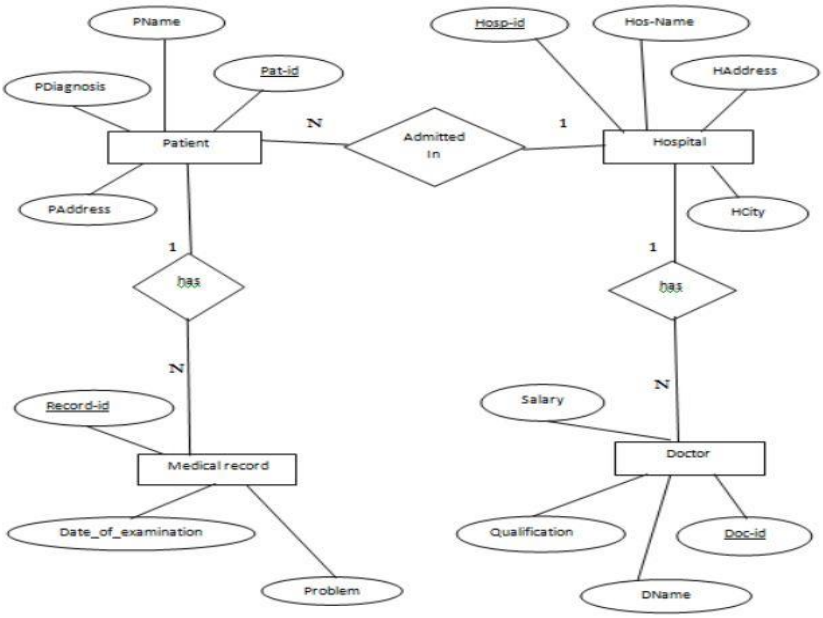


ER-diagram of library management system

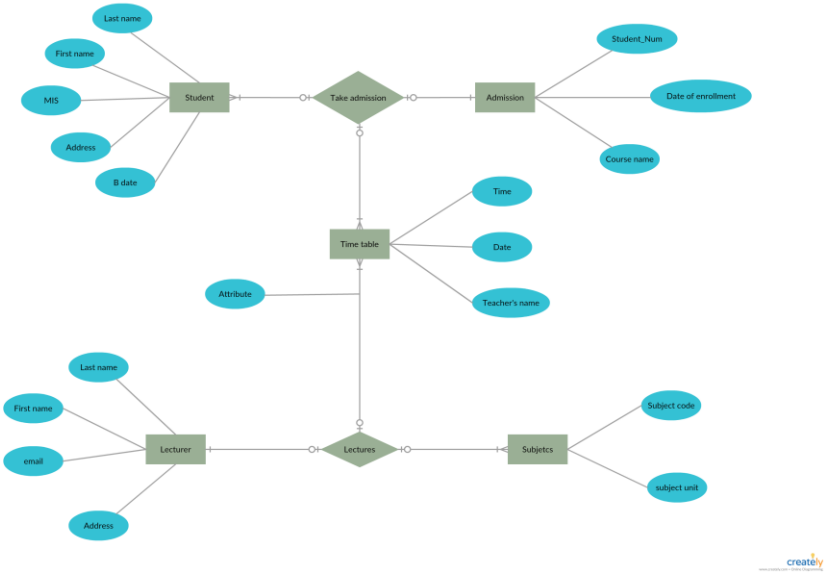
E-R Diagram of Library Management System



ER-diagram of hospital management system



ER-diagram of college management system



THE END