# CHAPTER-2

## ER-MODEL

**Data model:**

The data model describes the structure of a database. It is a collection of conceptual tools for describing data, data relationships and consistency constraints and various types of data model such as

1. Object based logical model
2. Record based logical model
3. Physical model

Types of data model:

1. Object based logical model
    a. ER-model
    b. Functional model
    c. Object oriented model
    d. Semantic model
2. Record based logical model
    a. Hierarchical database model
    b. Network model
    c. Relational model
3. Physical model

## Entity Relationship Model

The entity-relationship data model perceives the real world as consisting of basic objects, called entities and relationships among these objects. It was developed to facilitate data base design by allowing specification of an enterprise schema which represents the overall logical structure of a data base.

**Main features of ER-MODEL:**

- Entity relationship model is a high level conceptual model
- It allows us to describe the data involved in a real world enterprise in terms of objects and their relationships.
- It is widely used to develop an initial design of a database
- It provides a set of useful concepts that make it convenient for a developer to move from a baseid set of information to a detailed and description of information that can be easily implemented in a database system
- It describes data as a collection of entities, relationships and attributes.

**Basic concepts:**

The E-R data model employs three basic notions : entity sets, relationship sets and attributes.

**Entity sets:**

An entity is a “thing” or “object” in the real world that is distinguishable from all other objects. For example, each person in an enterprise is an entity. An entity has a set properties and the values for some set of properties may uniquely identify an entity.
BOOK is entity and its properties(calles as attributes) bookcode, booktitle, price etc .

An entity set is a set of entities of the same type that share the same properties, or attributes. The set of all persons who are customers at a given bank, for example, can be defined as the entity set customer.

**Attributes:**

An entity is represented by a set of attributes. Attributes are descriptive properties possessed by each member of an entity set.

**Customer** is an entity and its attributes are **customerid, custmername, custaddress** etc.

An attribute as used in the E-R model , can be characterized by the following attribute types.

a) **Simple and composite attribute:**

simple attributes are the attributes which can’t be divided into sub parts
eg: customerid,empno
composite attributes are the attributes which can be divided into subparts.
eg: name consisting of first name, middle name, last name
address consisting of city,pincode,state

b) **single-valued and multi-valued attribute:**
The attribute having unique value is single –valued attribute
eg: empno,customerid,regdno etc.
The attribute having more than one value is multi-valued attribute
eg: phone-no, dependent name, vehicle

c) **Derived Attribute:**

The values for this type of attribute can be derived from the values of existing attributes
eg: age which can be derived from (currentdate-birthdate)
experience_in_year can be calculated as (currentdate-joindate)

d) **NULL valued attribute:**
The attribute value which is unknown to user is called NULL valued attribute.

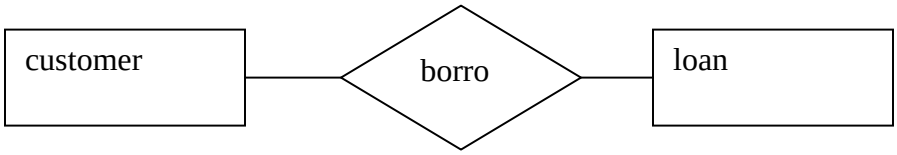
**Relationship sets:**

A relationship is an association among several entities.

A relationship set is a set of relationships of the same type. Formally, it is a mathematical relation on n>=2 entity sets. If E1,E2…En are entity sets, then a relation ship set R is a subset of

{(e1,e2,…en)|e1Є E1,e2 Є E2..,en Є En}

where (e1,e2,…en) is a relation ship.




Consider the two entity sets customer and loan. We define the relationship set borrow to denote the association between customers and the bank loans that the customers have.

**Mapping Cardinalities:**

Mapping cardinalities or cardinality ratios, express the number of entities to which another entity can be associated via a relationship set.

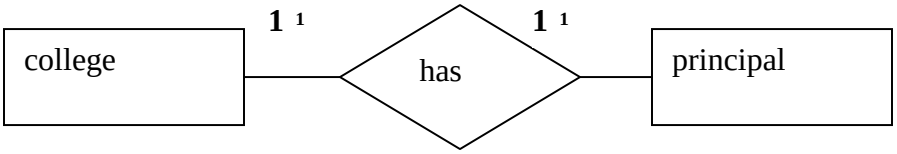
Mapping cardinalities are most useful in describing binary relationship sets, although they can contribute to the description of relationship sets that involve more than two entity sets.

For a binary relationship set R between entity sets A and B, the mapping cardinalities must be one of the following:

**one to one:**

An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.

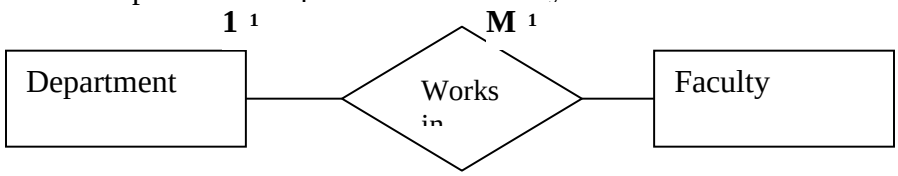
Eg: relationship between college and principal



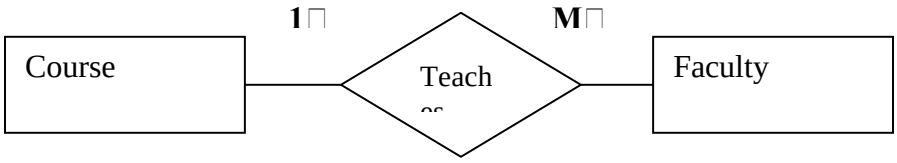

**One to many:**

An entity in A is associated with any number of entities in B. An entity in B is associated with at the most one entity in A.

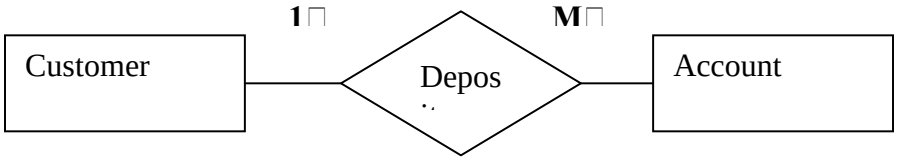
Eg: Relationship between department and faculty



**Many to one:**
An entity in A is associated with at most one entity in B. An entity in B is associated with any number in A.




**Many –to-many:**
Entities in A and B are associated with any number of entities from each other.




**More about entities and Relationship:**
**Recursive relationships:**
When the same entity type participates more than once in a relationship type in different roles, the relationship types are called recursive relationships.
**Participation constraints:**
The participation constraints specify whether the existence of any entity depends on its being related to another entity via the relationship. There are two types of participation constraints
**Total :**
.When all the entities from an entity set participate in a relationship type , is called total participation. For example, the participation of the entity set student on the relationship set must 'opts' is said to be total because every student enrolled must opt for a course.
**Partial:**
When it is not necessary for all the entities from an entity set to particapte ion a relationship type, it is called participation. For example, the participation of the entity set student in 'represents' is partial, since not every student in a class is a class representative.
**Weak Entity:**
Entity types that do not contain any key attribute, and hence can not be identified independently are called weak entity types. A weak entity can be identified by uniquely only by considering some of its attributes in conjunction with the primary key attribute of another entity, which is called the identifying owner entity.
Generally a partial key is attached to a weak entity type that is used for unique identification of weak entities related to a particular owner type. The following restrictions must hold:

- The owner entity set and the weak entity set must participate in one to may relationship set. This relationship set is called the identifying relationship set of the weak entity set.

- The weak entity set must have total participation in the identifying relationship.

**Example:**

Consider the entity type dependent related to employee entity, which is used to keep track of the dependents of each employee. The attributes of dependents are : name ,birthrate, sex and relationship. Each employee entity set is said to its own the dependent entities that are related to it. How ever, not that the 'dependent' entity does not exist of its own., it is dependent on the employee entity. In other words we can say that in case an employee leaves the organization all dependents related to without the entity 'employee'. Thus it is a weak entity.

**Keys:**

**Super key:**

A super key is a set of one or more attributes that taken collectively, allow us to identify uniquely an entity in the entity set.

For example , customer-id,(cname,customer-id),(cname,telno)

**Candidate key:**

In a relation R, a candidate key for R is a subset of the set of attributes of R, which have the following properties:

- *Uniqueness:* no two distinct tuples in R have the same values for the candidate key
- *Irreducible:* No proper subset of the candidate key has the uniqueness property that is the candidate key.

Eg: (cname,telno)

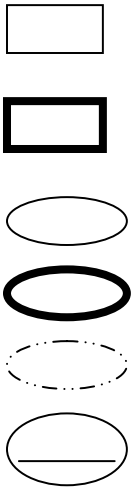
**Primary key:**

The primary key is the candidate key that is chosen by the database designer as the principal means of identifying entities with in an entity set. The remaining candidate keys if any, are called *alternate key*.

**ER-DIAGRAM:**

The overall logical structure of a database using ER-model graphically with the help of an ER-diagram.

**Symbols use ER- diagram:**



entity

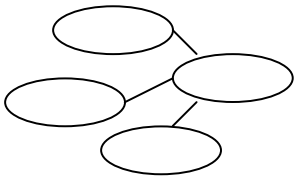
Weak entity

attribute

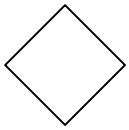
Multi valued attribute

Derived attribute

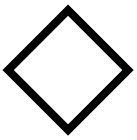
Key attribute



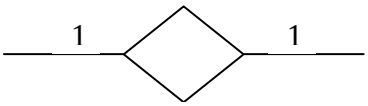
composite attribute



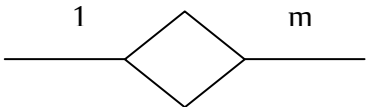
Relationship



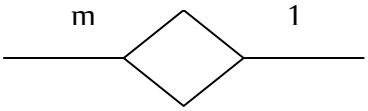
Identifying Relationship



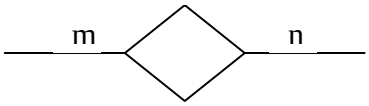

One-to -one



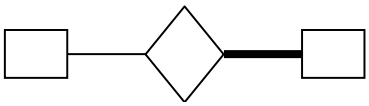

One-to -many



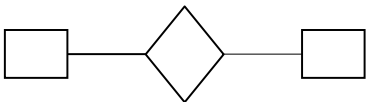

many-to -one




many-to -many



Total participation



Partial participation

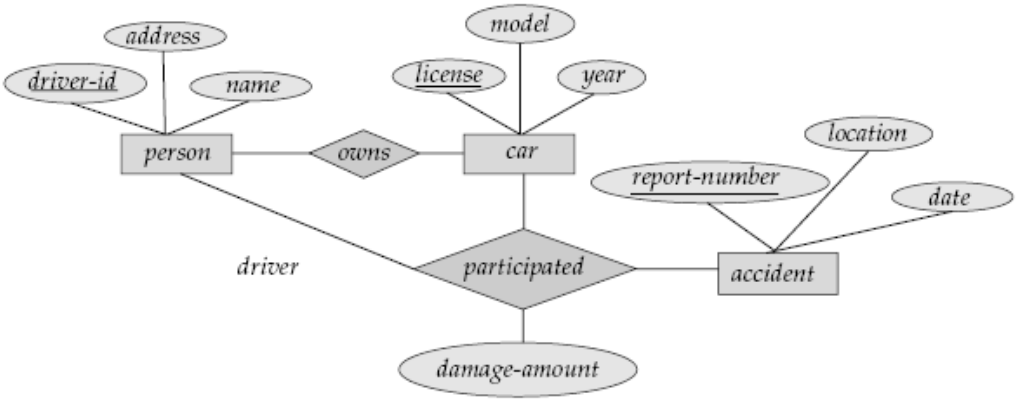



Figure 2.1 E-R diagram for a Car-insurance company.

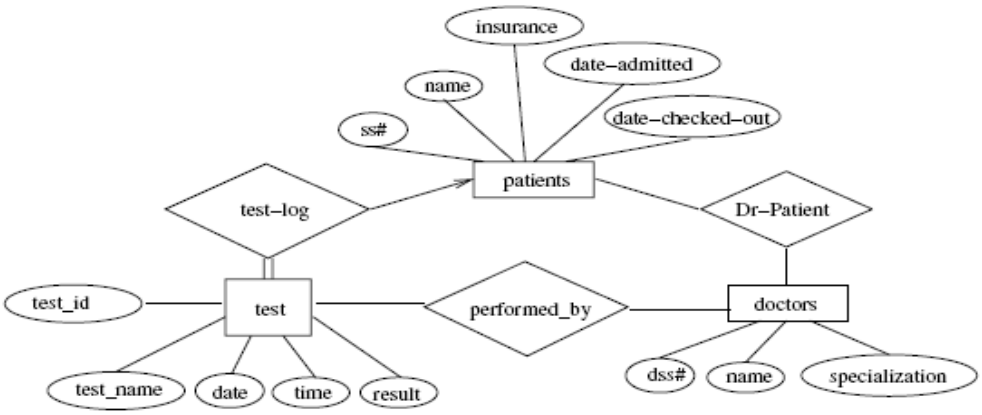



Figure 2.2 E-R diagram for a hospital.

A university registrar's office maintains data about the following entities: (a) courses, including number, title, credits, syllabus, and prerequisites; (b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom; (c) students, including student-id, name, and program; and (d) instructors, including identification number, name, department, and title. Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled.

Construct an E-R diagram for the registrar's office. Document all assumptions that you make about the mapping constraints.

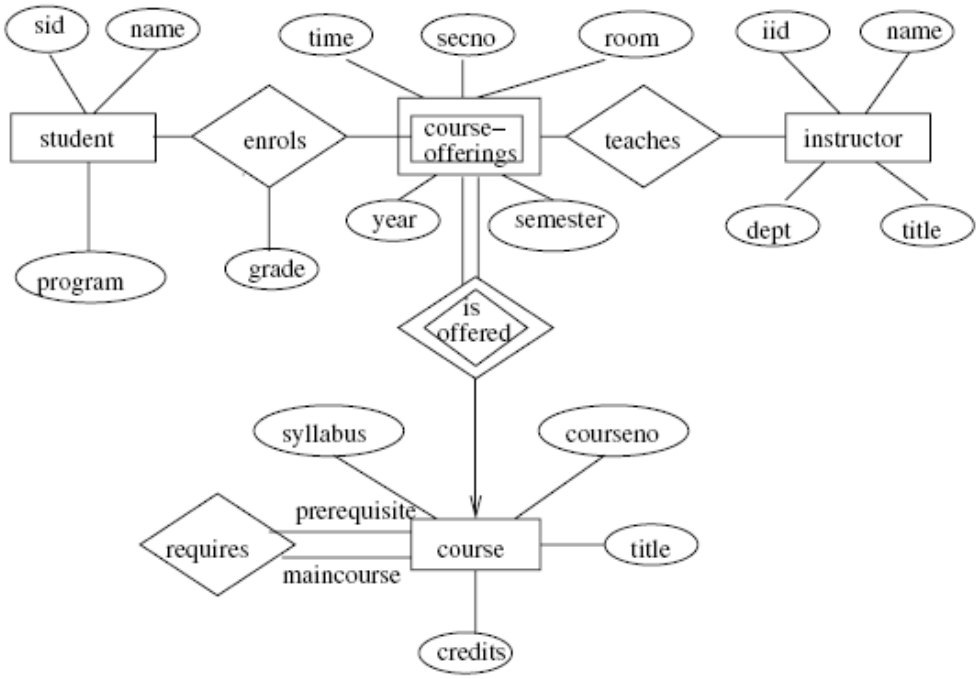



Figure 2.3 E-R diagram for a university.

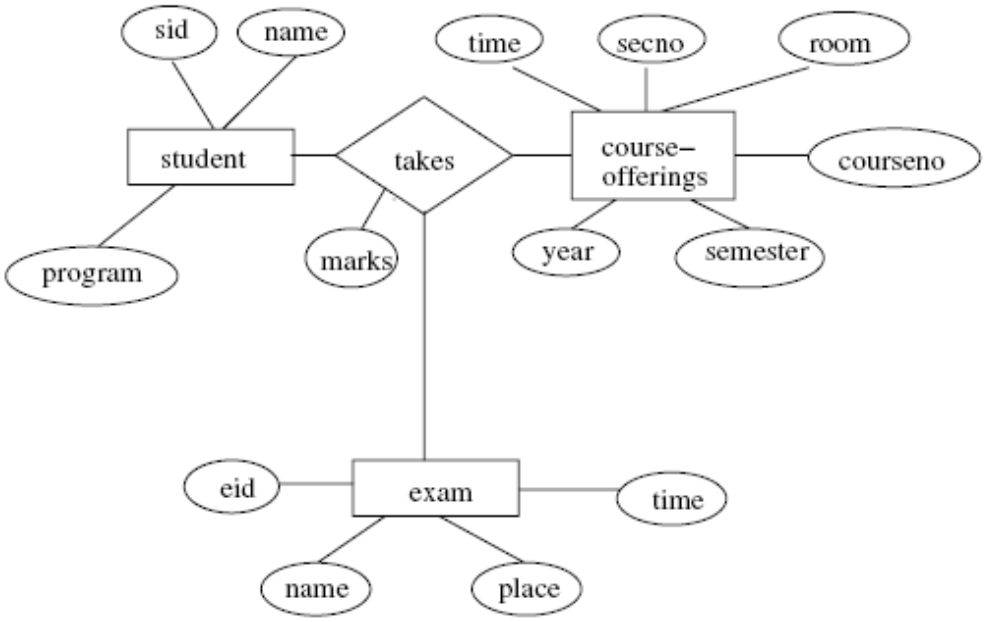



Figure 2.4 E-R diagram for marks database.

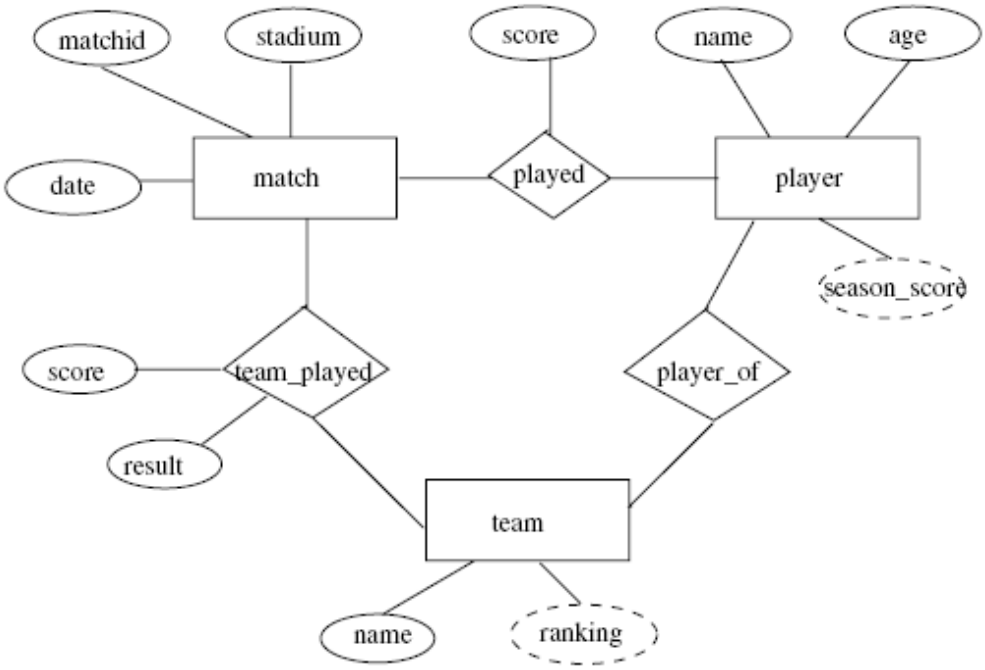



Figure 2.7 E-R diagram for all teams statistics.

Consider a university database for the scheduling of classrooms for final exams. This database could be modeled as the single entity set *exam*, with attributes *course-name*, *section-number*, *room-number*, and *time*. Alternatively, one or more additional entity sets could be defined, along with relationship sets to replace some of the attributes of the *exam* entity set, as

- *course* with attributes *name*, *department*, and *c-number*
- *section* with attributes *s-number* and *enrollment*, and dependent as a weak entity set on *course*
- *room* with attributes *r-number*, *capacity*, and *building*

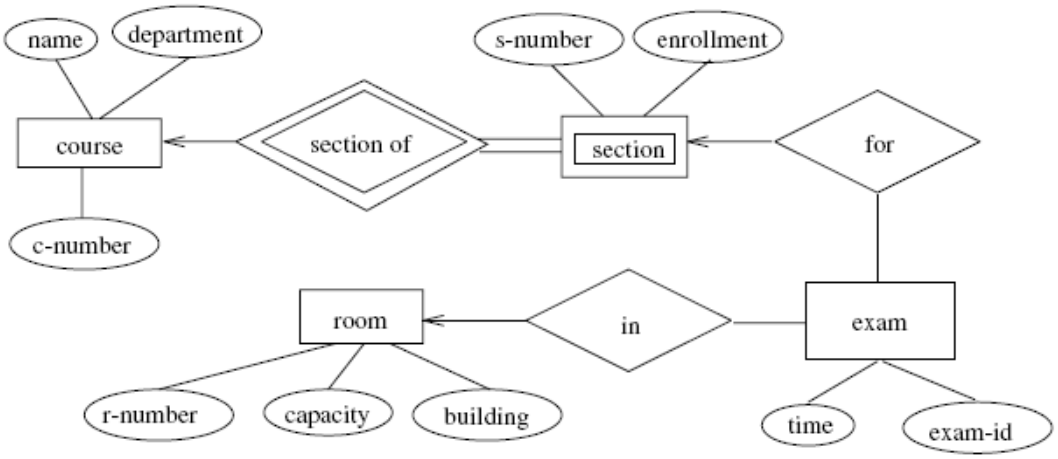
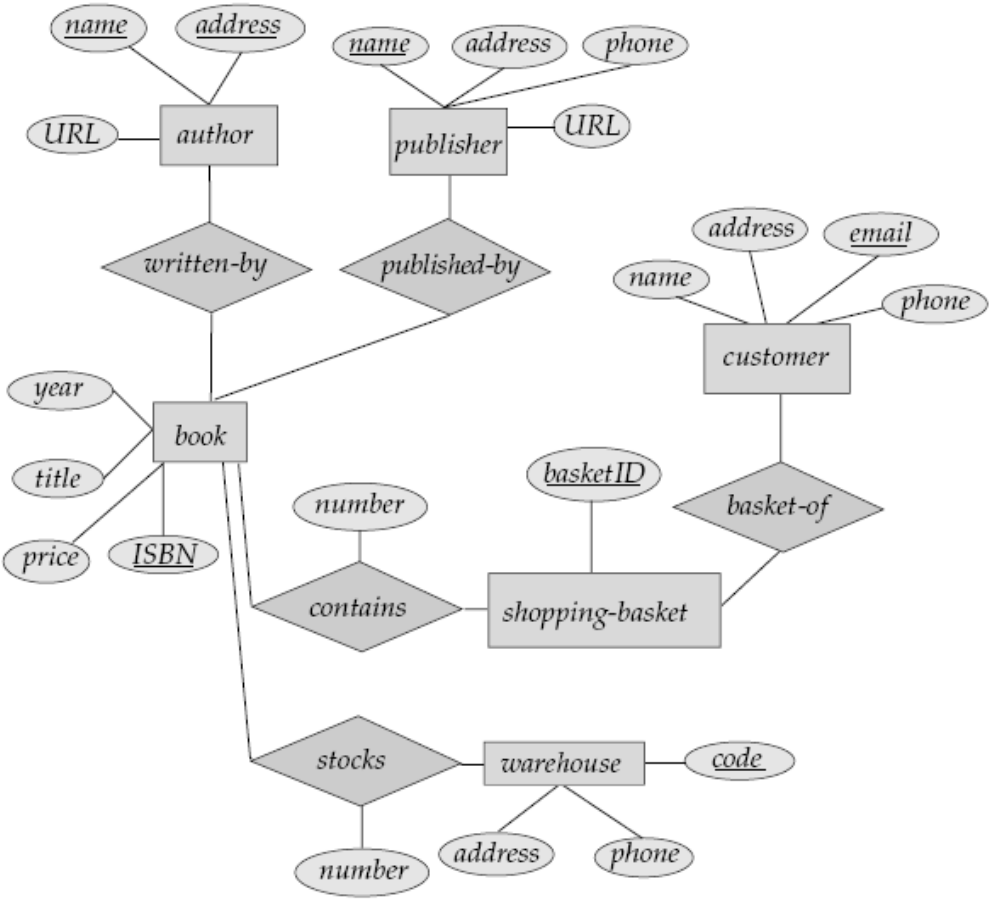



Figure 2.12 E-R diagram for exam scheduling.



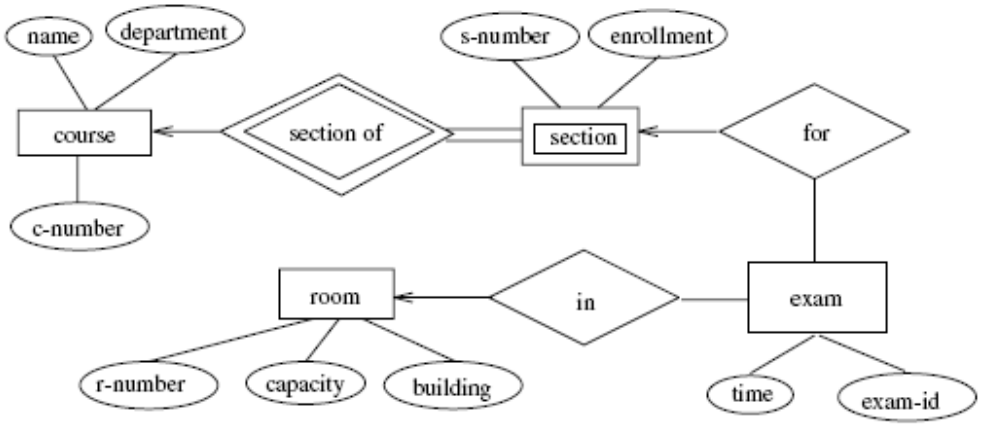
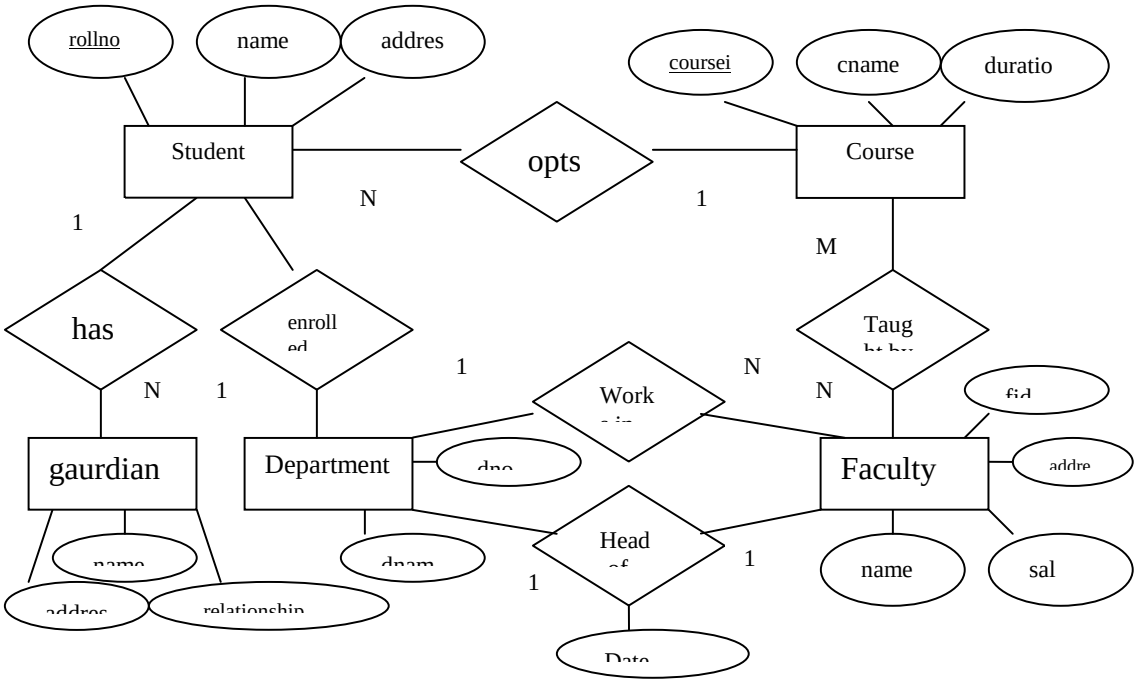



Figure 2.12 E-R diagram for exam scheduling.

**ER- Diagram For College Database**




**Conversion of ER-diagram to relational database**

**Conversion of entity sets:**

1. For each strong entity type E in the ER diagram, we create a relation R containing all the single attributes of E. The primary key of the relation R will be one of the key attribute of R.

   **STUDENT**(rollno (primary key),name, address)
   **FACULTY**(id(primary key),name ,address, salary)
   **COURSE**(course-id,(primary key),course_name,duration)
   **DEPARTMENT**(dno(primary key),dname)