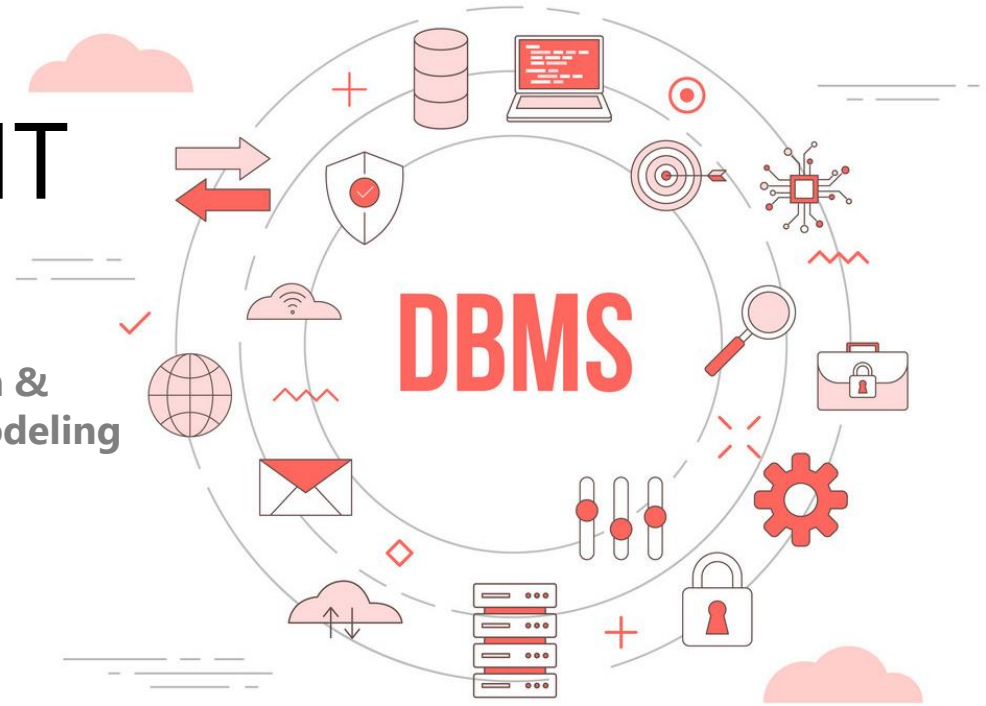
DATABASE MANAGEMENT SYSTEMS

Database Design & ER Modeling



Dr. Jay Sarraf

School of Computer Engineering

KIIT Deemed to be University

Overview

What is Database Design Process?

Database design is a set of methods that aid in the creation, development, deployment, and upkeep of corporate data management systems. A well-designed database is simple to manage, increases data consistency, and saves money on disc storage space. The database designer determines how the data pieces must be related and what information must be saved.

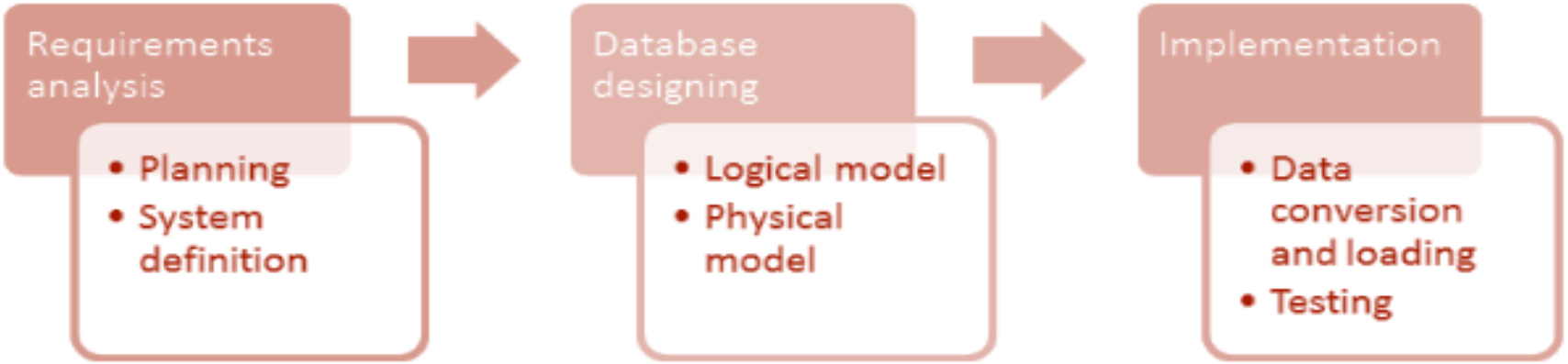
Why it is important?

- To meet the requirements of the users
- To have high performance.

Note: DBMS design process is crucial for high performance database system.

Cont...

The database development life cycle has a number of stages that are followed when developing database systems.



Database development life cycle

Cont…

1 **Requirements analysis**

- Planning - This stage of database design ideas is focused with the overall Database Development Life Cycle planning. It takes into account the organization's Information Systems strategy.
- System definition - At this step, the scope and bounds of the proposed database system are defined.

2 **Database designing**

- Conceptual database design: to build the conceptual representation of the database, which has the identification of the important entities, relationships, and attributes (ER Model)
- Logical database design: to convert the conceptual representation to the logical structure of the database, which includes designing the relations (Relational Data Model)
- Physical database design: to decide how the logical structure is to be physically implemented (as base relations) in DBMS.

Cont…

3 **Implementation**

- Loading and converting data: from the old system into the new database is the focus of the data conversion and loading stage of relational database architecture.

- Testing: During this phase, errors in the recently put into place system are to be found. It compares the database to the requirements laid down.

ER Model

In 1971, Peter Chen proposed the ER diagram as a way to visualise the ER model. By enabling the creation of an enterprise schema, which defines the overall logical structure of a database, the ER model was created to simplify database design. The ER model is quite helpful for conceptually mapping the relationships and meaning of real-world organisations.

Component of ER Diagram

The ER diagram consists of three basic concepts:

- Entities
- Attributes
- Relationship

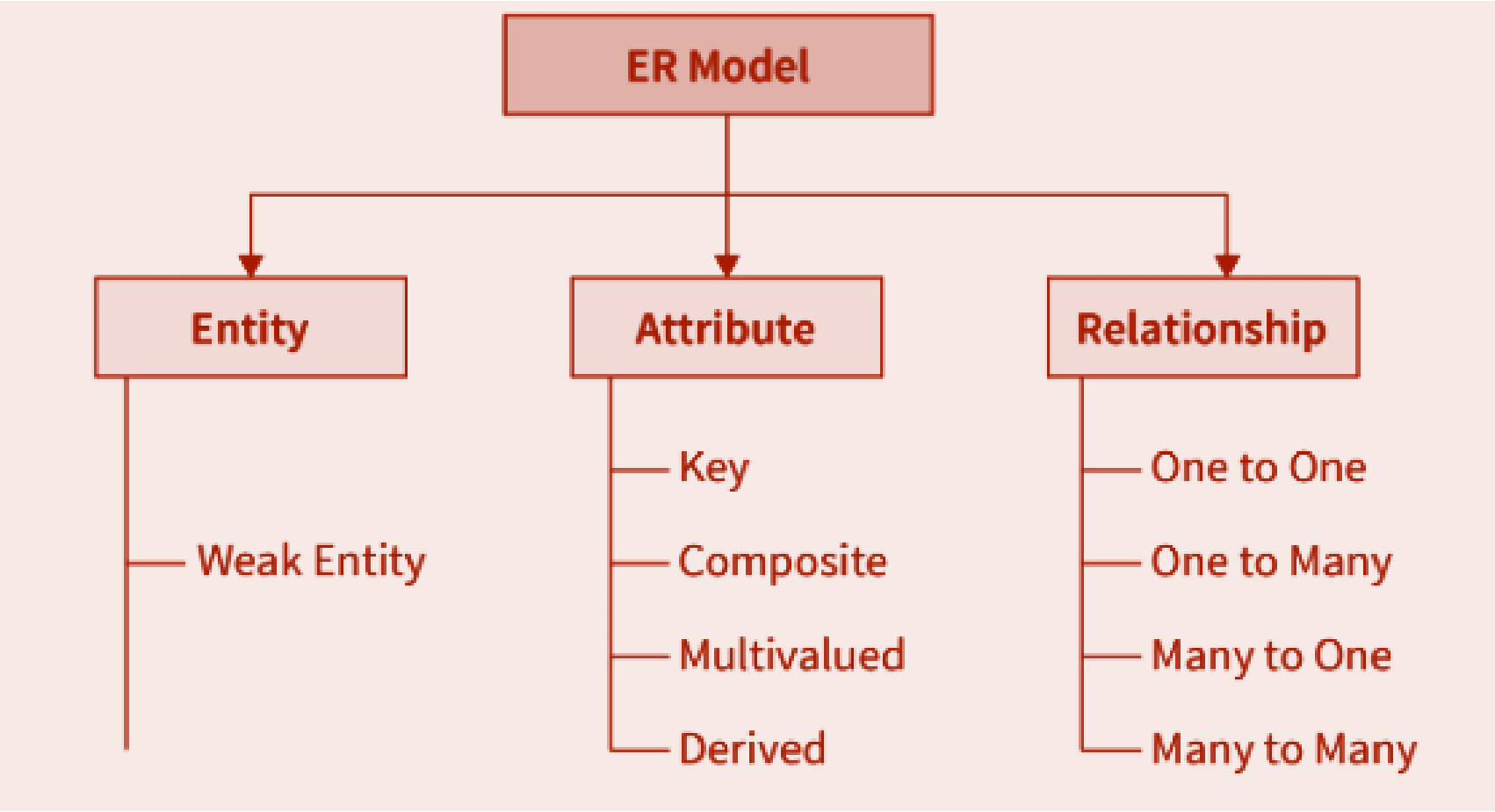
The ER model also has an associated diagrammatic representation, the ER diagram, which can express the overall logical structure of a database graphically.

Facts about ER model

Why ER diagram is used?

- Helps in the definition of terminologies used in entity relationship modelling.
- Give a glimpse of how your tables should connect and what fields each table will have.
- Description of entities, properties, and relationships is aided by.
- Because ER diagrams can be converted into relational tables, building databases fast is possible.
- Database designers might utilize ER diagrams as a guide for incorporating data in certain software applications.
- With the aid of an ERP diagram, the database designer is better able to comprehend the data that will be stored in the database.
- You may communicate with users about the logical structure of the database using an ERD Diagram.

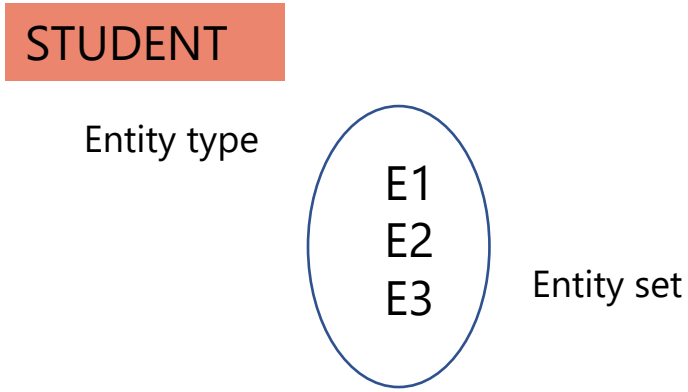
Components of ER model



Entities

Entities

An entity, which contains data in the database, might be a place, person, item, event, or idea. Entities must possess an attribute and a distinct key as part of their characteristics. Each "attribute" that makes up an entity serves to represent it. It is represented by a rectangle containing the entity set's name. The entity set name, a noun, is usually written in all capital letters.



Cont…

Two types of entities:

Strong Entity

A strong entity is a sort of entity that possesses a crucial quality. It is independent of other schema entities. A single rectangle in the ER diagram serves as the representation of a strong entity, which always has a primary key.

As an illustration, we may claim that the student is a strong entity type since each student in the organisation is uniquely identified by their roll number.

Weak Entity

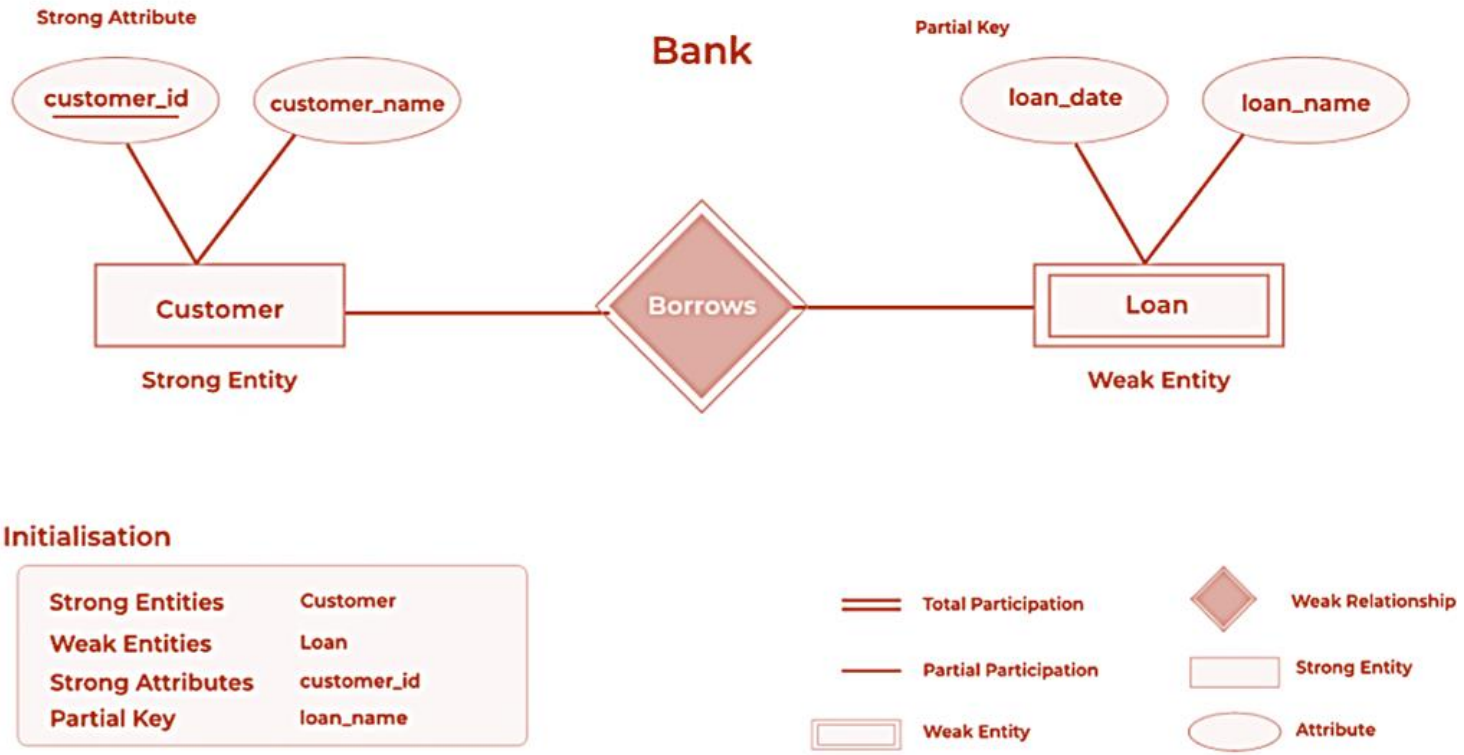
Weak entity types lack a key characteristic, making it impossible to distinguish them solely based on their properties. Therefore, in order to produce a primary key, a foreign key must be combined with its properties.

They are referred to be weak entity types since they are unable to recognize on their own. Its distinct identity is dependent on another strong entity. In ER diagrams, a double-outlined rectangle represents a weak entity.

Cont…

For instance, there may be several customer from the same loan date, the loan name cannot be used to identify customers in a specific way.

Therefore, in order to specifically identify entities of the Loan Entity Type, we require a property of the Strong Entity Type, namely “Customer."



Cont…

Strong Entity		Weak Entity
• Strong entity always **has a primary key** • It is **not dependent** on any other entity • Represented by a **single rectangle** • Relationship between two strong entities is represented by a **single diamond** • A strong entity **may or may not have total participation**	VS	• Will **not have a primary key but it has partial discriminator key** • Which entity is **dependent** on the strong entity • Represented by **double rectangle** relationship between a strong entity and the weak entity is represented by **double Diamond** • It has always **total participation**

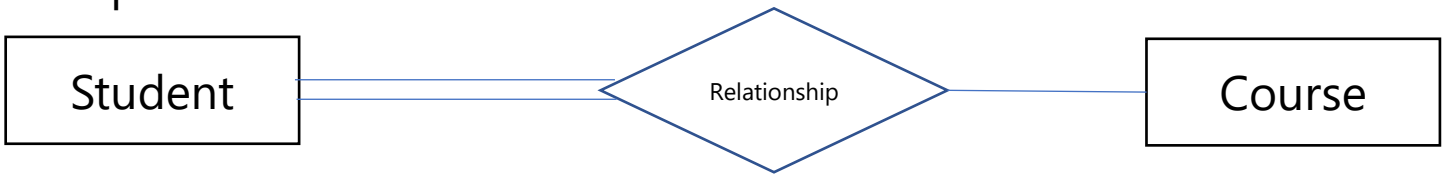
Participation Constraints

Total Participation

Each entity in the entity set is required to engage in a minimum of one relationship instance in that relationship set, according to the clause. Because of this, it is also referred to as a required activity.

A double line between the entity set and relationship set shows the total participation.

Example:



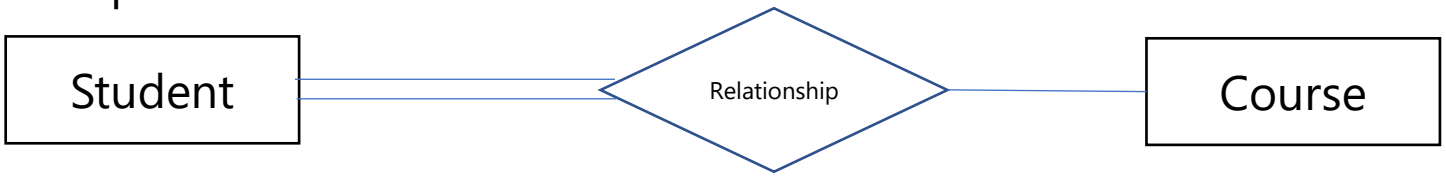
It specifies that there might exist some courses for which no enrolments are made.

Partial Participation

It clearly states whether or not each entity in the entity set may take part in a given relationship instance in the relationship set. It is also known as voluntary participation because of this.

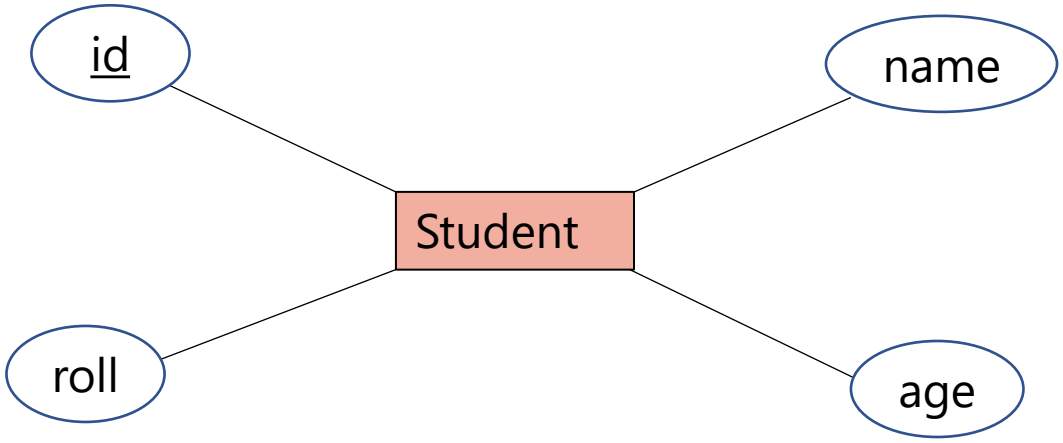
A single line connecting the entity set and relationship set represents partial involvement.

Example:



It specifies that there might exist some courses for which no enrolments are made.

Attributes



Attributes

The characteristics or features that characterise an entity type are known as its attributes. The attribute is represented by an ellipse in the ER diagram.

As an illustration, the attributes id, Name, Age, and roll no in this case form the entity type Student.

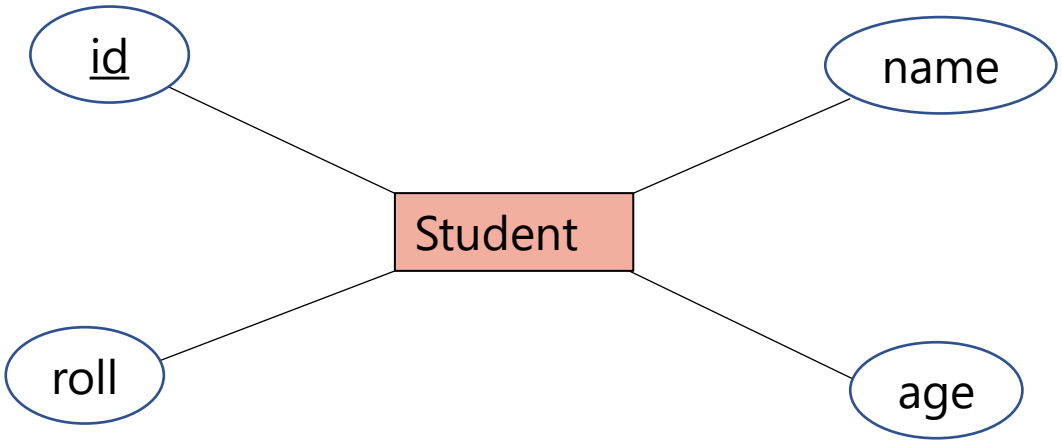
Cont…

Different types of attributes:

Simple attribute:

Simple attributes are attributes that cannot be further broken down into sub-attributes.

It is also referred to as the key attribute and is an atomic value. In ER diagrams, the basic attributes are shown as an oval with the name of the attribute in bold.



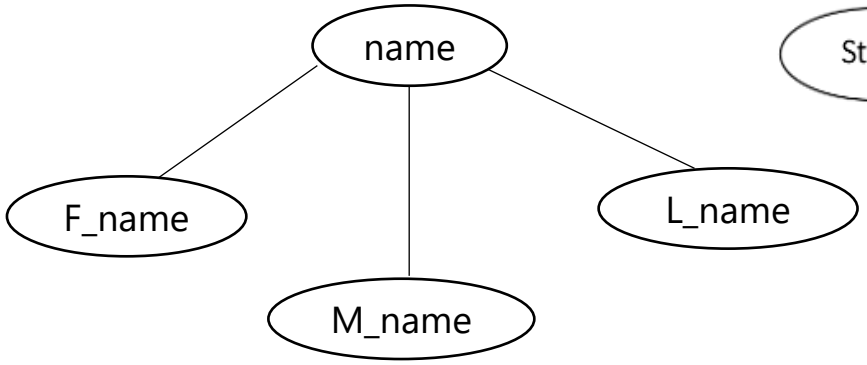
Cont…

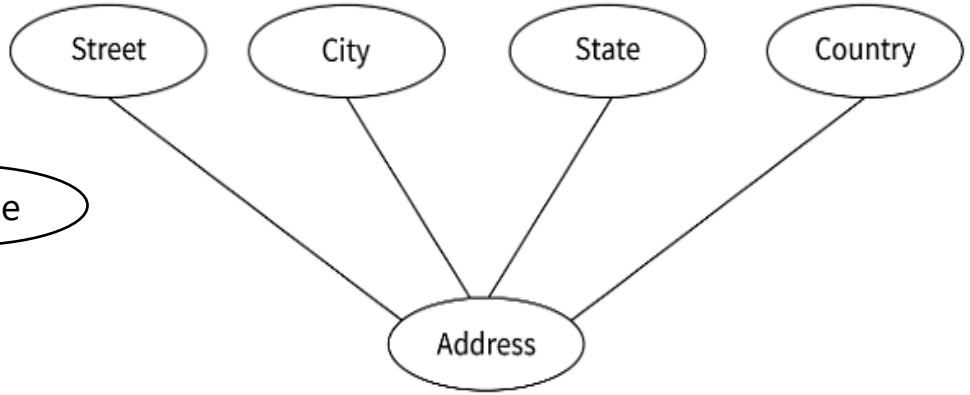
Different types of attributes:

Composite attribute:

A composite characteristic is one that may be broken down into basic attributes and is made up of several additional qualities.

An ellipse is used to symbolize the composite property.





Different types of attributes:

Multivalued attribute:

Attributes with several values are multivalued attributes. A multivalued attribute is represented by the double oval.

One student may have many mobile numbers, hence the student's mobile number is an example of a multivalued attribute.

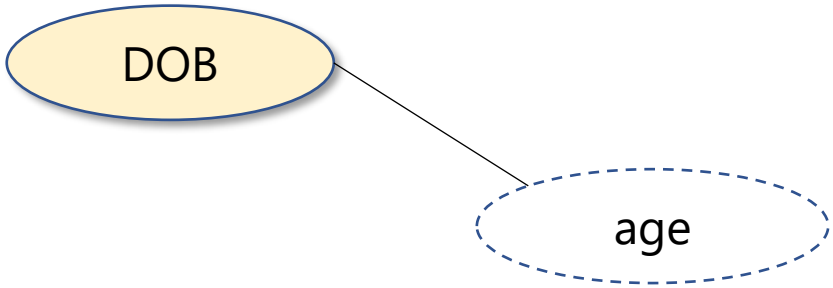
Cont…

Different types of attributes:

Derived attribute:

The qualities that can be derived from other attributes of an entity type are known as derived attributes. In the ER diagram, a dashed oval symbol stands in for the derived characteristics.

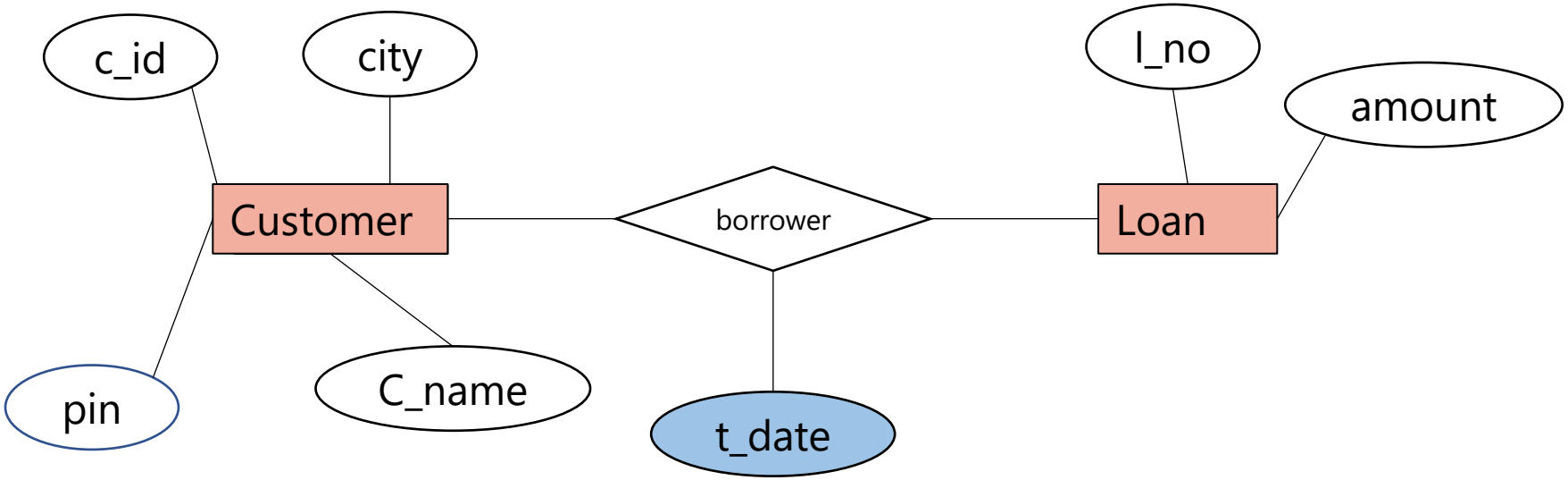
For instance, the date of birth (DOB) information may be used to determine the age attribute. It is therefore a derived characteristic.



Different types of attributes:

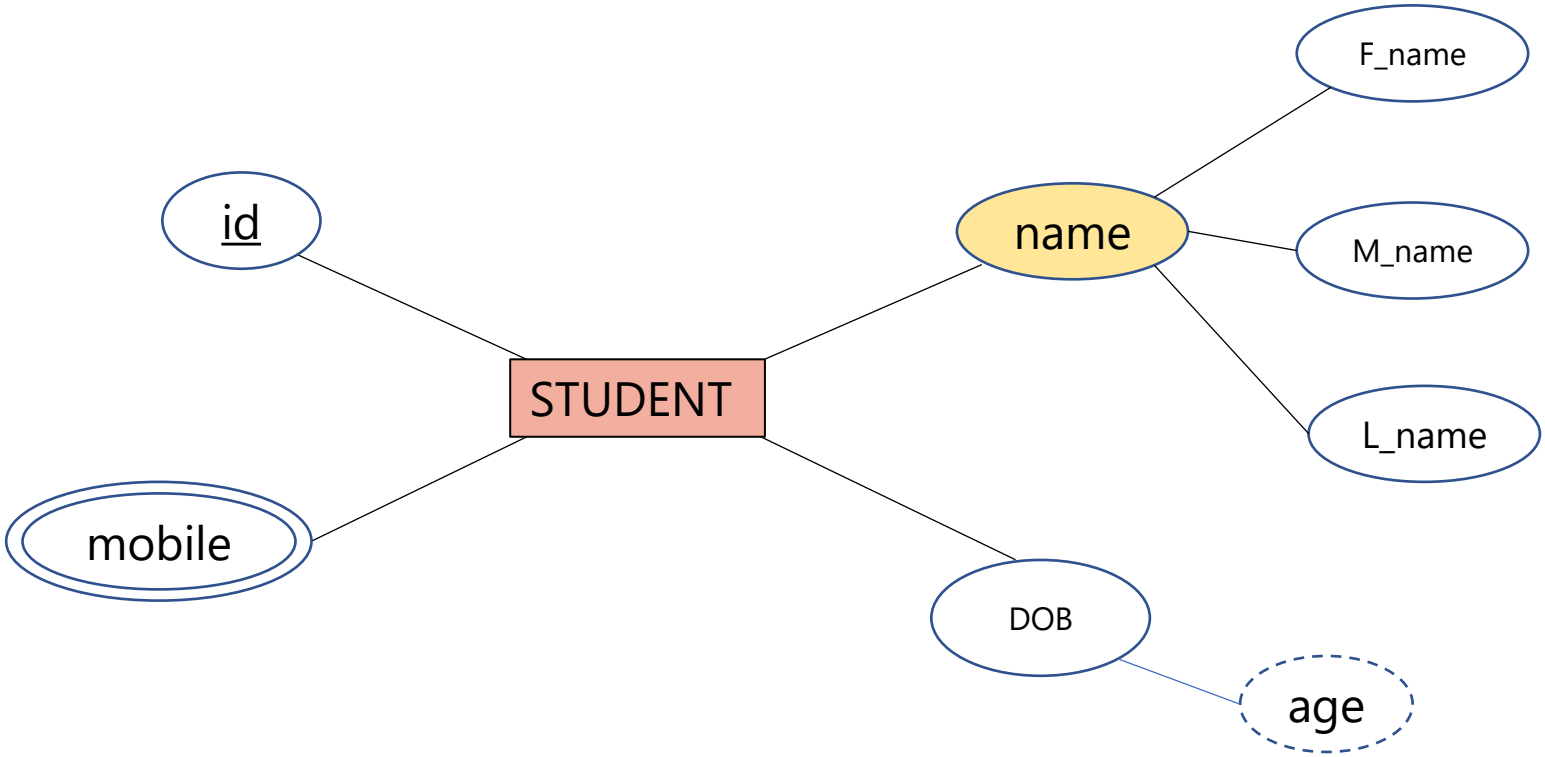
Descriptive attribute:

A relationship may also have attributes called descriptive attributes for representing the description about the association.



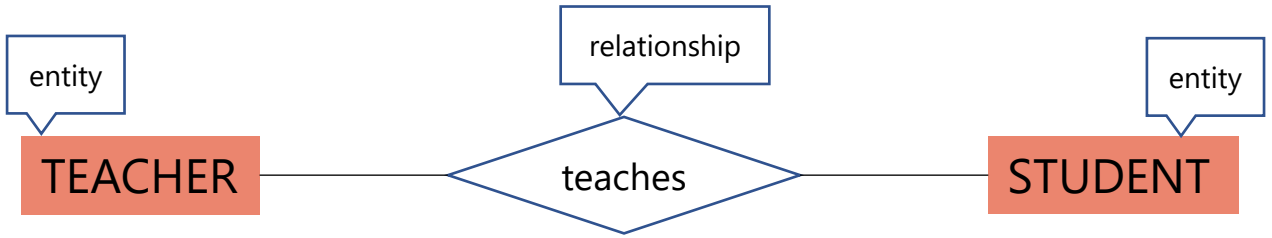
Cont…

The following figure illustrates the whole student entity type (represented by the rectangular shape) with its many properties, including address as a composite attribute, age as a derived attribute, phone number as a multivalued attribute, and roll number as the key attribute.



Relationships

The relationship between several things is described using the idea of relationship. This is indicated with a rhombus or a diamond symbol. For instance, the relationship between the instructor and student entity types is shown by a diamond shape.



The degree of relationship in DBMS can be classified as **Unary, Binary, Ternary, or N-ary relationship**.

Cont…

Degree of Relationships:

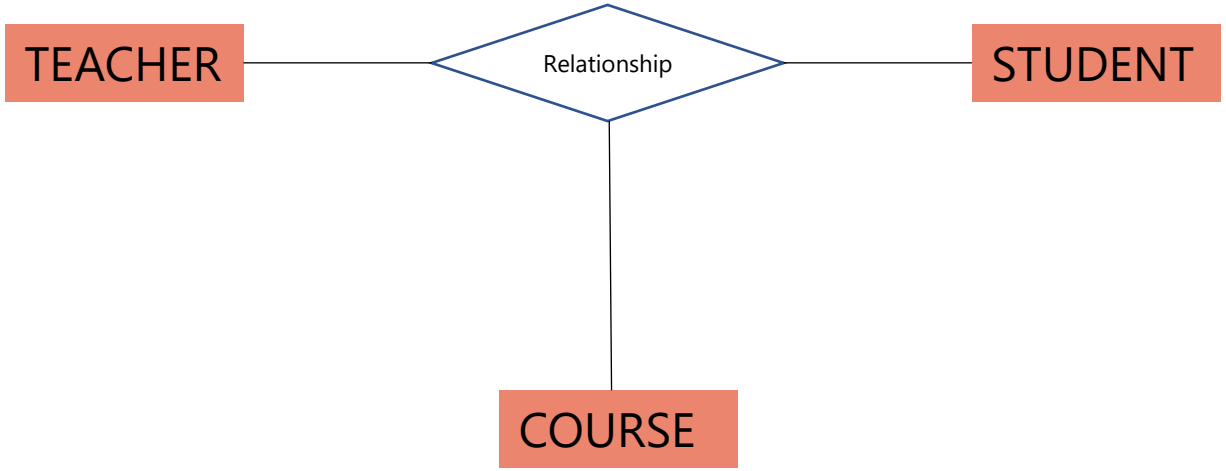
1 Binary Relationships

A relationship is referred to be a binary relationship when there are two entity set engaged in the relationship.

Degree of Relationships:

2 Ternary Relationship

A relationship is referred to be a binary relationship when there are exactly three entity set engaged in the relationship.



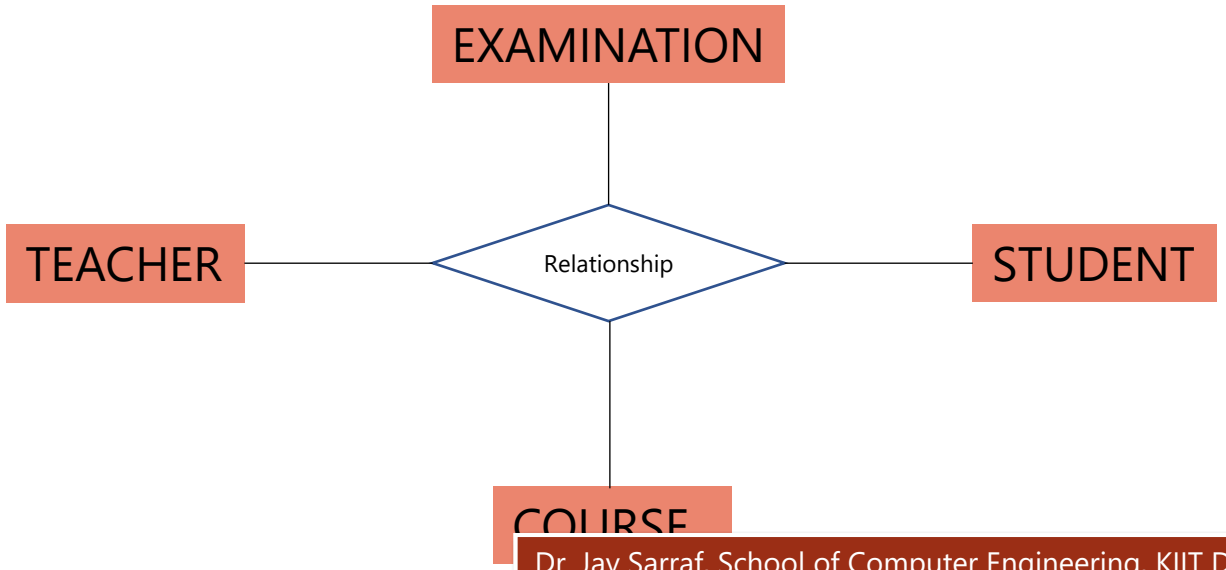
Cont…

Degree of Relationships:

3 **N-ary Relationship**

A relationship is referred to be a binary relationship when there are more than three entity set engaged in the relationship.

Additionally, the relationship illustrates the various entity sets involved, making it extremely valuable for analysing how the system was designed.

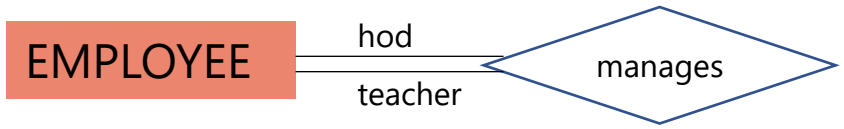


Degree of Relationships:

4 Unary / Recursive Relationships

A relationship is referred to be a unary or recursive relationship when there is just one entity set engaging in the relationship.

The role name of an entity set can be used to identify that entity's participation in a relationship type. Role names in recursive relationship types indicate how the participation functions.



Cardinality Constraints

Mapping Cardinality Constraints

A mapping constraint is a type of data constraint that specifies how many other entities one object may be connected to through a relationship set.

The description of relationship sets including more than two entity sets is where it is most helpful.

For a binary relationship set (2 entities) the mapping cardinality must be one of the following types:

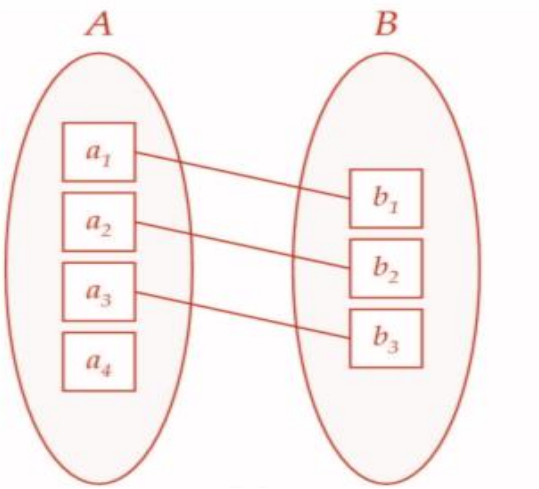
- One to one (1:1)
- One to many (1:M)
- Many to one (M:1)
- Many to many (M:M)

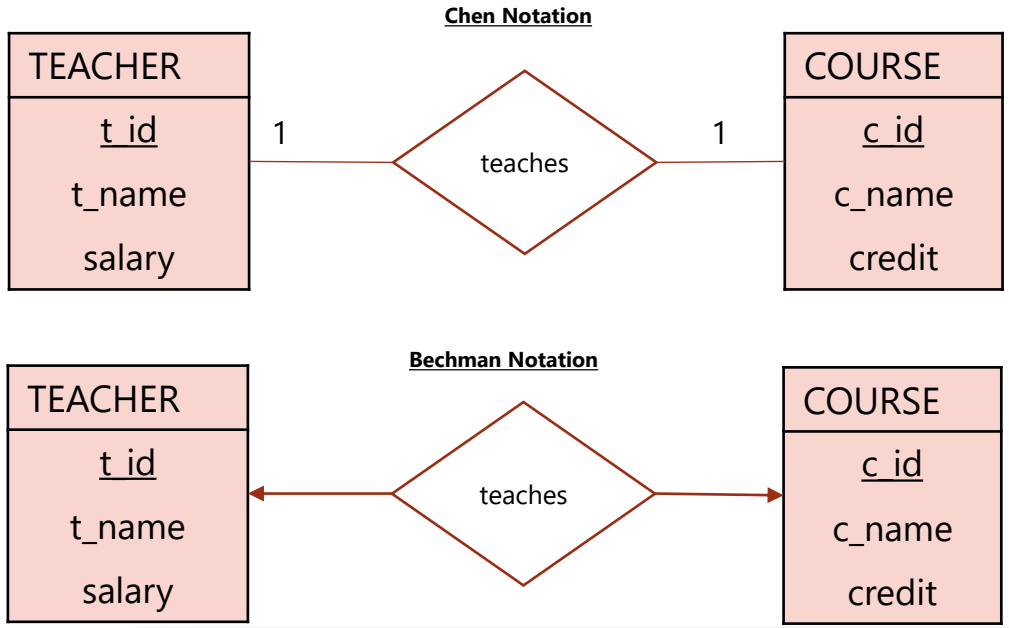
Cont…

One to one (1:1)

As the name suggests, it translates a single entity from the first entity set to a single entity from the second entity set.

One entity from entity set A is connected with no more than one entity from entity set B, as seen in the graphic below.



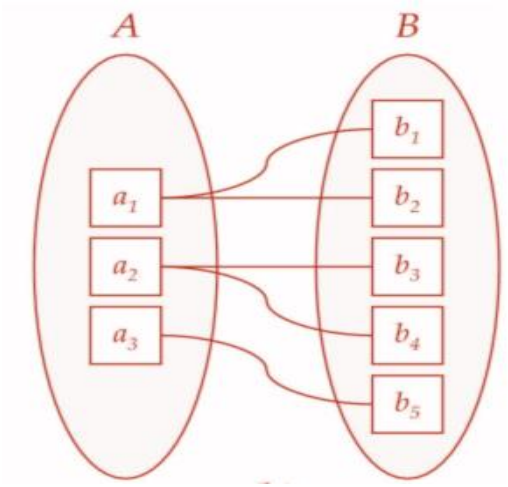


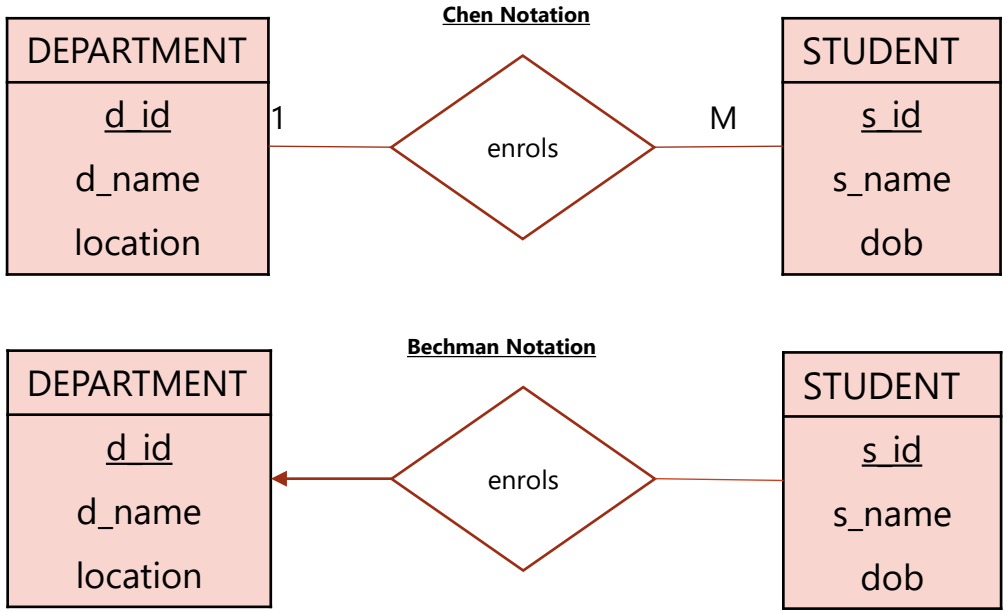
Cont…

One to Many (1:M)

It associates a single entity from the first entity set to many entities from the second entity set.

One entity from entity set A is connected with more than one entity from entity set B, as seen in the graphic below.



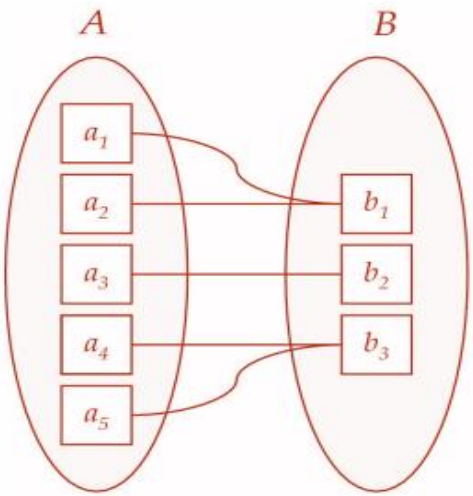


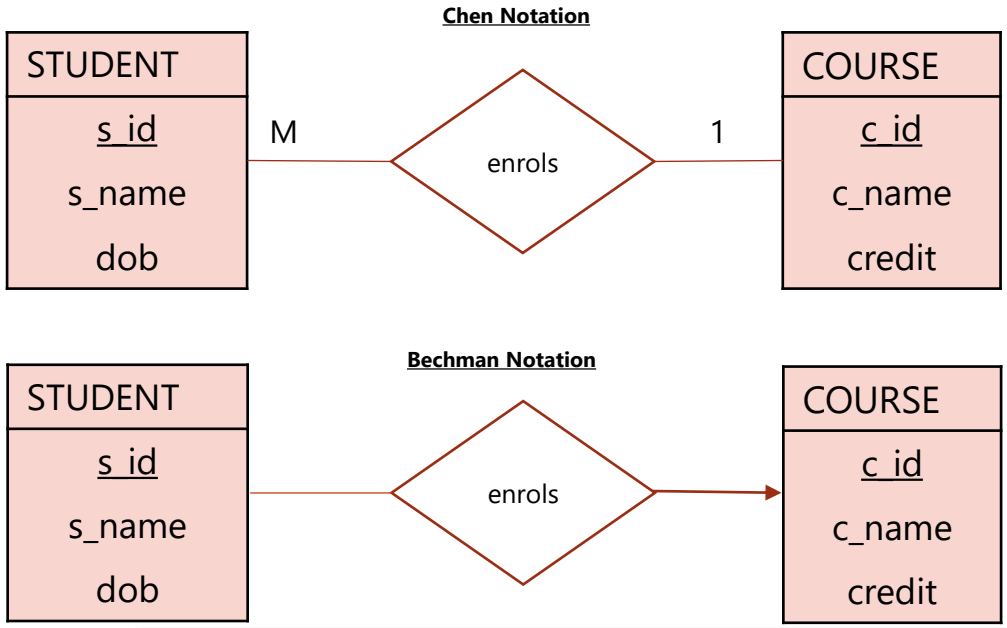
Cont…

Many to One (M:1)

It associates many entity from the first entity set to one entity from the second entity set.

Many entity from entities set A are connected with one entity from entity set B, as seen in the graphic below.



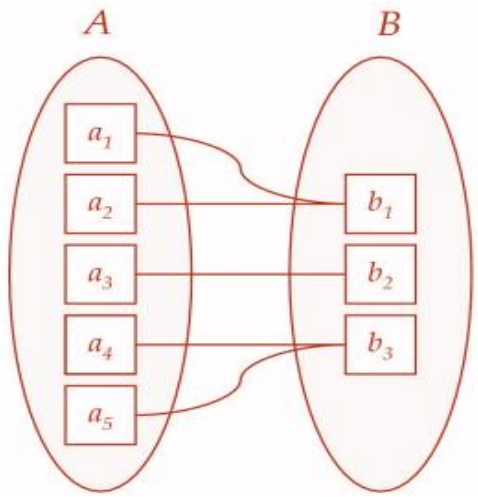


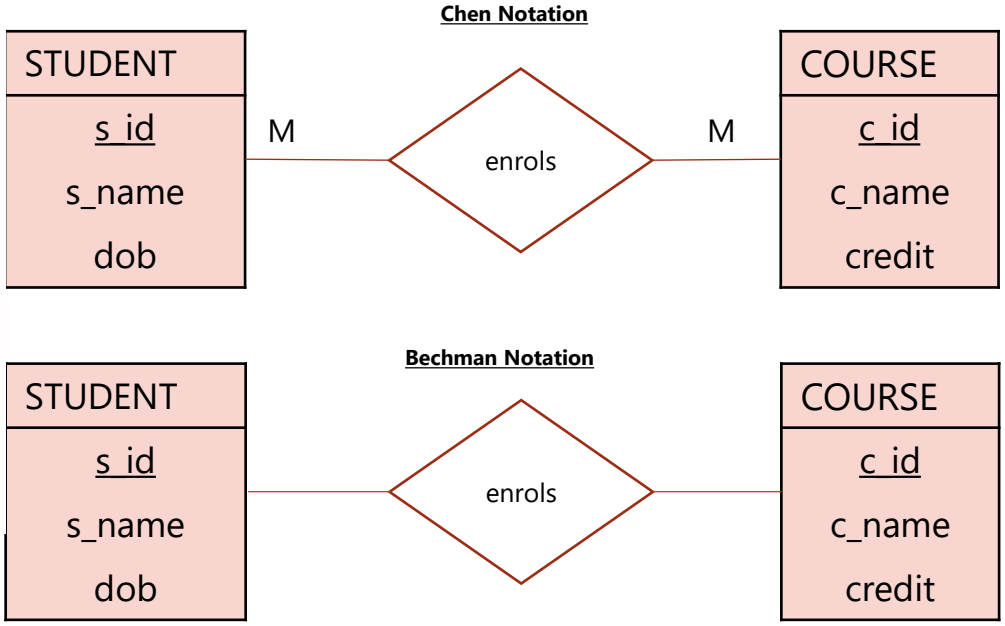
Cont…

Many to Many (M:M)

It associates many entity from the first entity set to one entity from the second entity set.

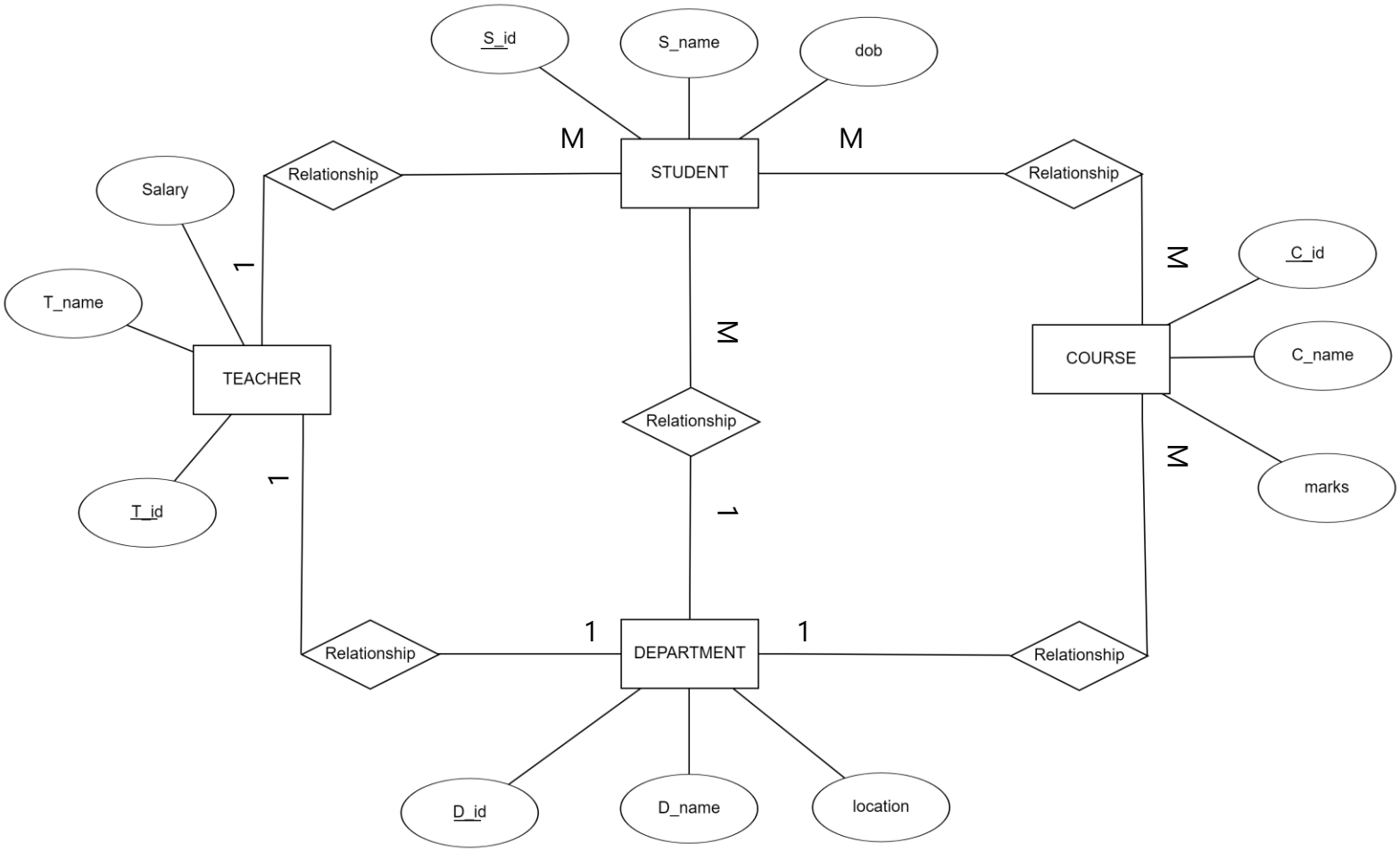
Many entities from entity set A are connected with many entities from entity set B, as seen in the graphic below.





Cont…

Illustration of relationships and their cardinalities.



Keys

A **key** is use to locate a collection of characteristics that are sufficient to differentiate one entity from another.
Rather than being a property of individual entities itself, a key distinct entities.

Primary key: Each entity in an entity set can be uniquely identified by an attribute or group of attributes. The important key in the previous ER diagram is *S_id, T_id, D_id & C_id*. The attribute name is underlined to show what it looks like.

Super key: It is a set of fields that a RDBMS table may use to uniquely identify any row.

Candidate key: A very similar idea is a candidate key, where the superkey is reduced to the minimum number of columns necessary to separately identify each row.

The remaining candidate keys except the primary key are called as **alternate** keys.

Cont…

Foreign key: A foreign key is distinct from a super key, candidate key, or main key since it is used to join or link two tables together, as opposed to any of the other three.

Two tables are connected by a foreign key, which is utilized as the primary key. It means that any attribute defined as a primary key will function as a foreign key attribute in any other table. However, one should be aware that a foreign key is unrelated to the primary key.

An attribute must meet the following requirements to qualify as a foreign key:

- The Strong Entity Set's primary key's domain must match that of the foreign key attribute's values.
- Any entity in an entity set has a foreign key value that is either null or has to match the primary key data values of another entity in the primary key entity set.

Foreign keys cannot be represented in ER diagrams.

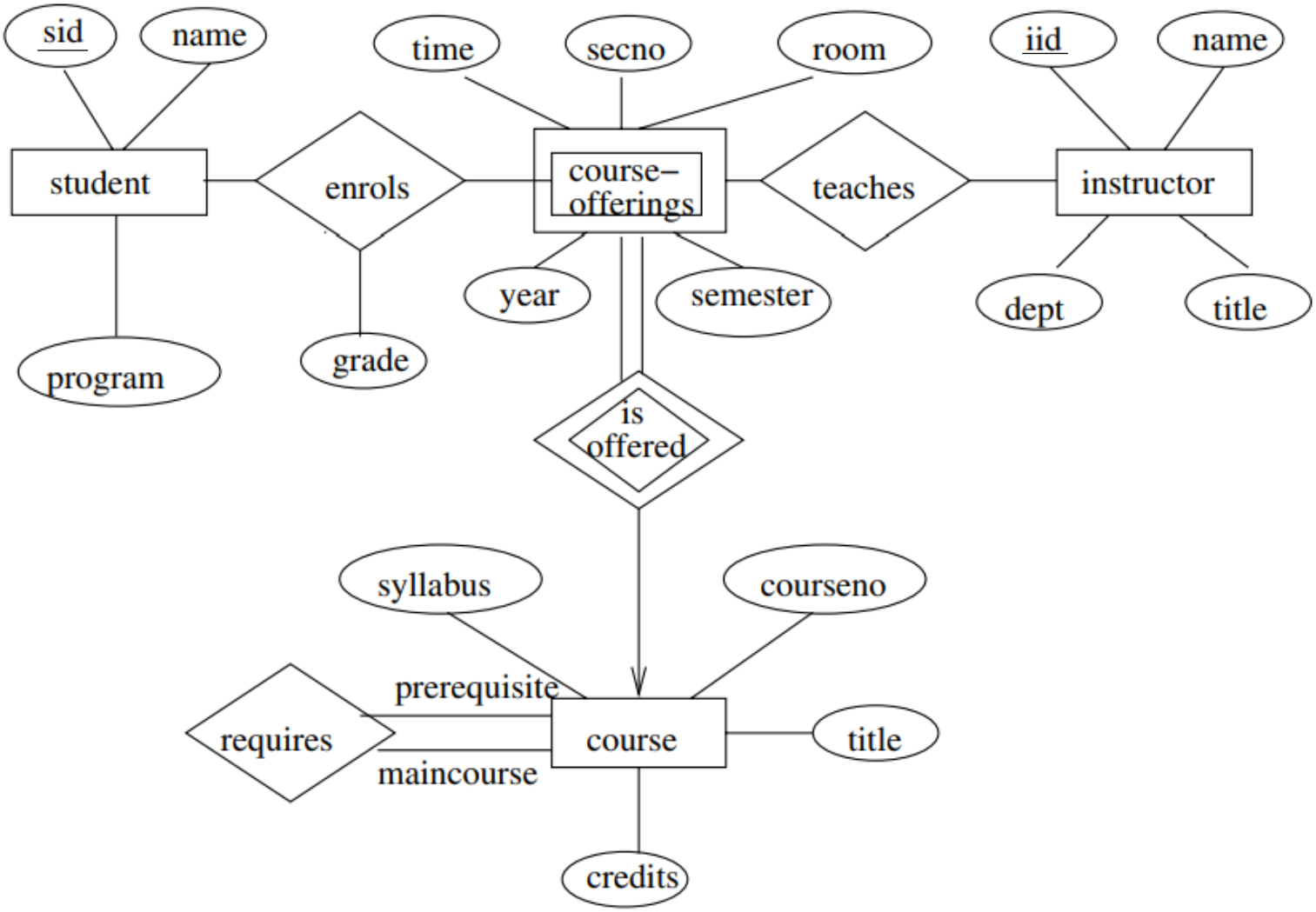
Question

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.

Answer

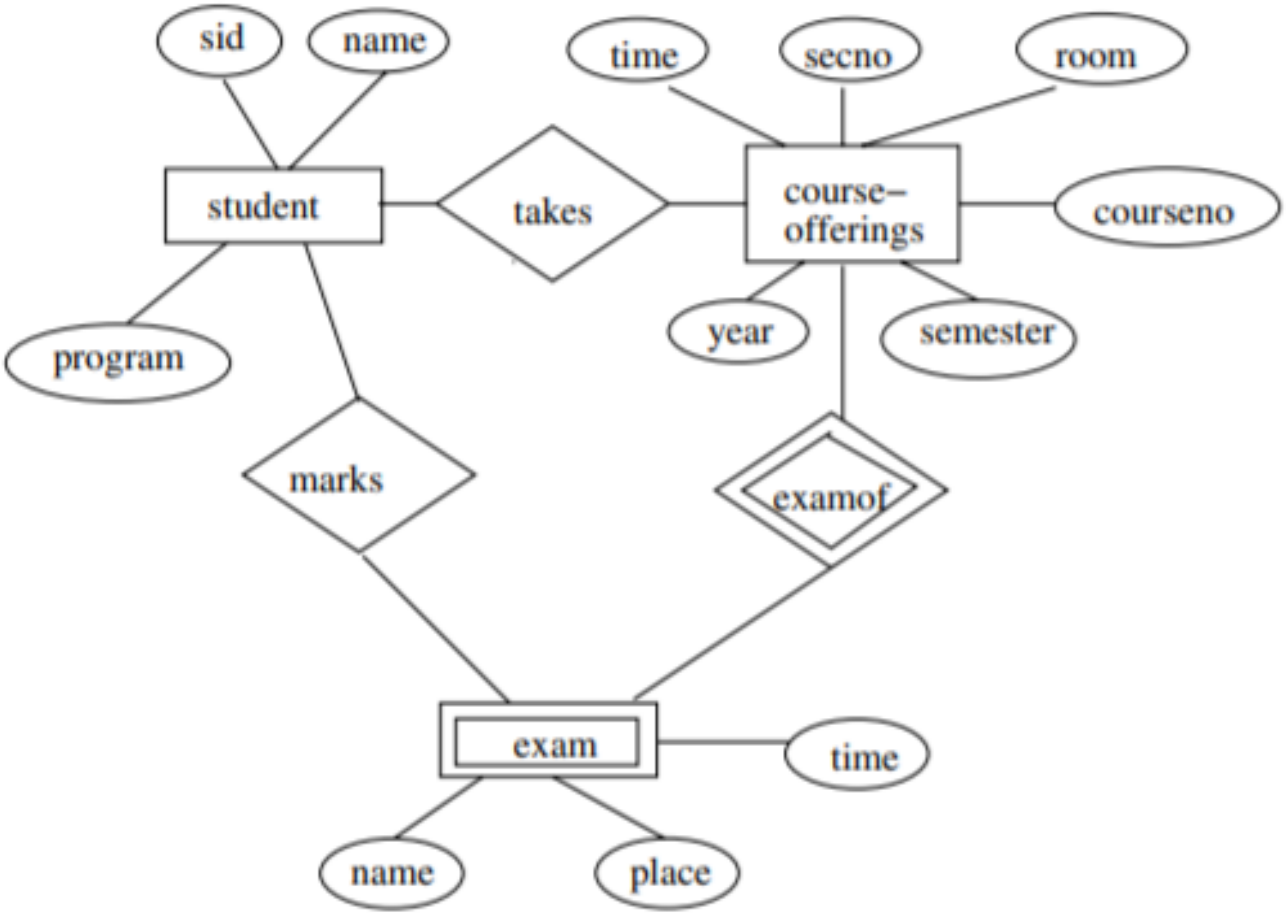

sid
name
time
secno
room
iid
name
student
enrols
course–
offerings
teaches
instructor
year
semester
dept
title
program
grade
is
offered
syllabus
courseno
prerequisite
requires
course
title
maincourse
credits

Question

Consider a database used to record the marks that students get in different exams of different course offerings.

Construct an E-R diagram that uses only a binary relationship between students and course-offerings. Make sure that only one relationship exists between a particular student and course offering pair, yet you can represent the marks that a student gets in different exams of a course offering.

Answer

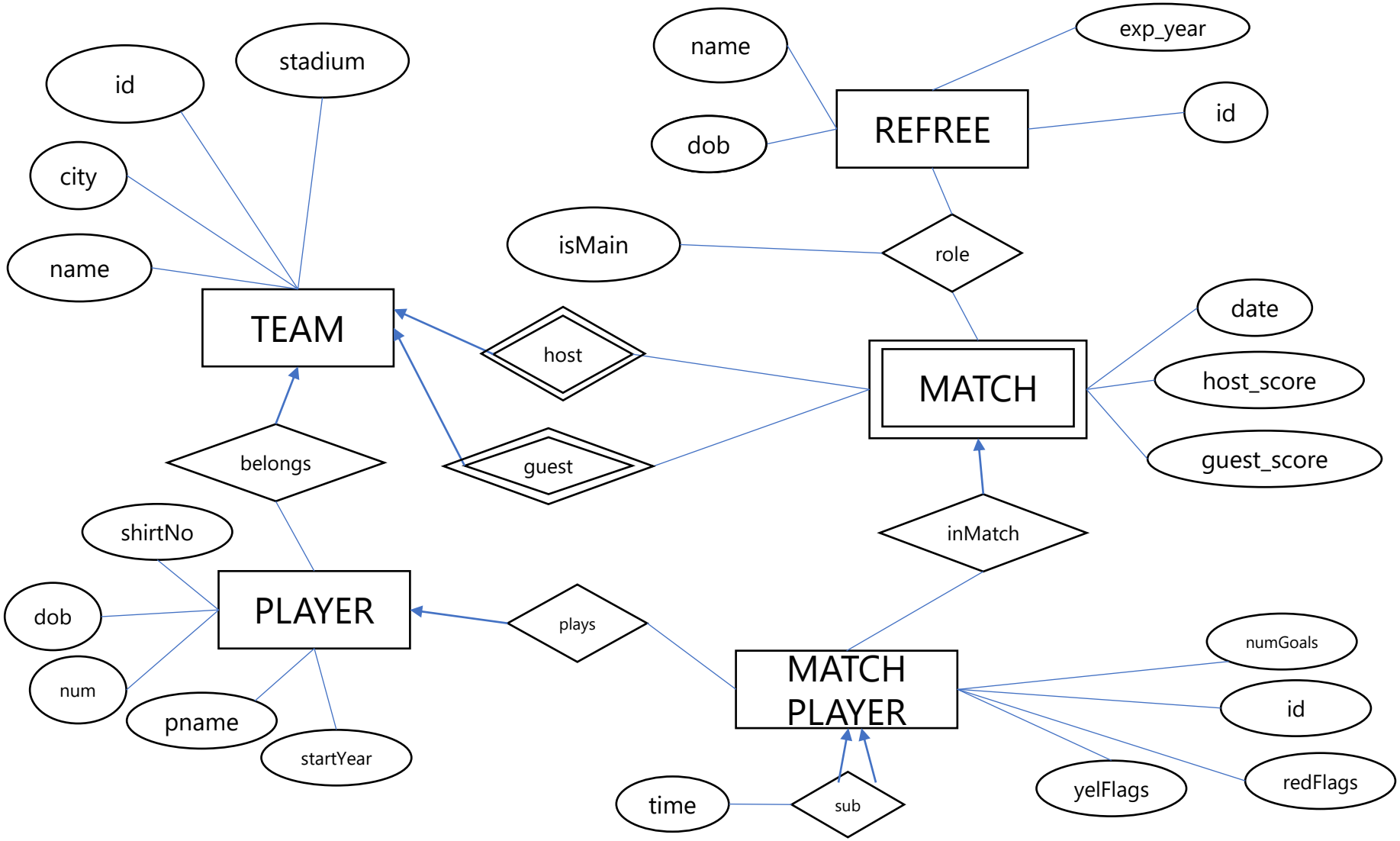

sid
name
time
secno
room
student
takes
course–
offerings
courseno
program
year
semester
marks
examof
exam
time
name
place

Question

There are many teams playing matches among themselves. In each match, there is a host team and a guest team. The match takes place in the stadium of host team. Each team has an unique **TID** along with **Tname**, **Home_stadium** and **Home_city** as attributes. Each team has many players and each player belongs to only one team.

Each player is uniquely identified by **PID** and has **Pname**, **Birth_date**, **Debut_year** and **Jersy_no**. For each match, we are interested to record the **date of play**, **game result**, **players participate**, **number of red cards issued**, **number of yellow cards issued, substitute players' list and the times of substitutions**. Each match has exactly three **referees**. Out of these referees, one referee is the main referee and the other two are assistant referees. Each referee has as **RID** (unique), **Rname**, **Birth_date**, **Years_of_experience**. Construct a ER diagram for the above scenario.

Answer


id
stadium
city
name
TEAM
name
dob
REFREE
exp_year
id
isMain
role
host
guest
belongs
MATCH
date
host_score
guest_score
inMatch
shirtNo
dob
num
pname
startYear
PLAYER
plays
MATCH PLAYER
numGoals
id
redFlags
yelFlags
time
sub

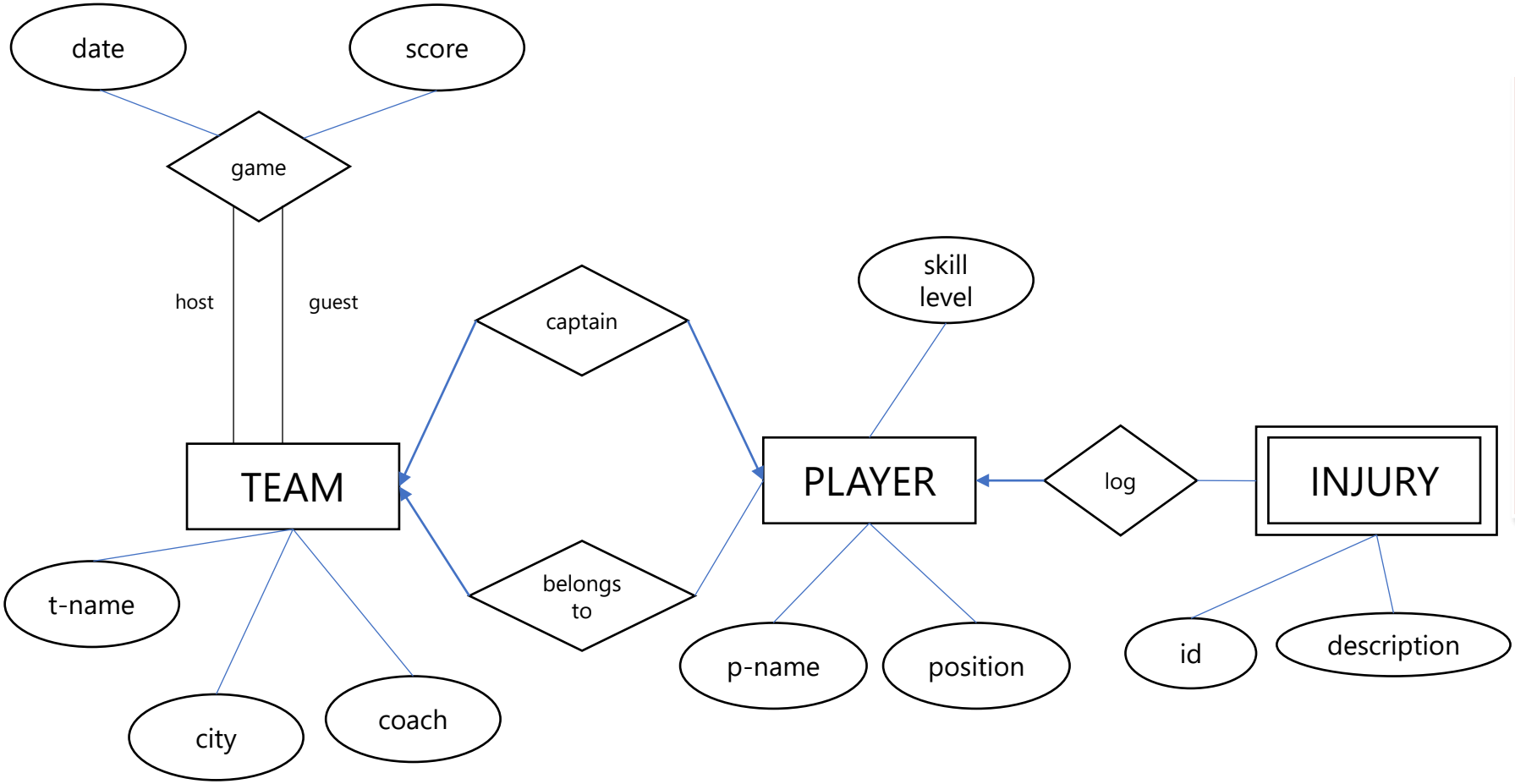
Question

Suppose you are given the following requirements for a simple database for the National Hockey League (NHL):

- the NHL has many teams,
- each team has a name, a city, a coach, a captain, and a set of players,
- each player belongs to only one team,
- each player has a name, a position (such as left wing or goalie), a skill level, and a set of injury records,
- a team captain is also a player,
- a game is played between two teams (referred to as host_team and guest_team) and has a date (such as May 11th, 1999) and a score (such as 4 to 2)

Construct a clean and concise ER diagram for the NHL database.

Answer


date
score
game
host
guest
captain
skill level
TEAM
PLAYER
log
INJURY
t-name
belongs to
p-name
position
id
description
city
coach