DBMS

DBMS (+8 marks)

ER model, Relational model - 1 problem

Relational algebra, Tuple calculus - 1 problem

SQL Integrity constraint - 1 problem

Transactional & concurrency control - 1 problem

File organization, indexing, B, B^+ trees - 1 problem

→ Time consuming

→ Multi-level indexing can also be problematic

Data

↓

Related data

for ex: s(Ram, 101, 21) ⇒ This is a record

↓

A collection of such records forms a relation / table

Eg: (Ram, 101, 21)
(Shyam, 102, 21)
(Gita, 103, 20)

↓

R_1 (Name, Rollno, Age) ⇒ This representation is called relational schema

Name of the table

↓

A collection of such relations forms a database

Eg: I can have another table with schema R_2(student, courses)

↓

The above examples look like a database for a college. We'll call it college.db

Note: In the above example,

These are known as attributes/columns

Name	Roll no.	Age
Ram	101	21

This is known as a tuple/row.

Also, in a relation schema, the unique attribute is underlined. In the above case, it is roll.no.

A row differs from a tuple in such a way that a row always means the entire record while a tuple can be a subset of the record as well.

The relation is a set:

$R_1 \{ (\underline{Ram, 101, 21}), (Shyam, 102, 21), \vdots \}$

Order cannot be changed

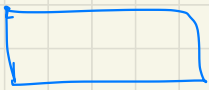
The number of elements in this set is called the quality of the set?

ER-Model

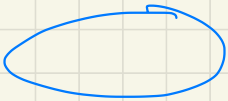
→ Diagrammatic representation of collection of objects (called entities) and relationship among the objects and their entities.

→ ER Model Components

- Entity Set
- Entity attribute
- Relationship b/w the entity



(An entity set is represented by a rectangle)



→ We represent entity sets and not single entities (although the terms are used interchangeably)

Entity

→ Any real world object that is distinguishable from other objects based on their value (attributes) is an entity.

→ The diff. b/w an entity and an object is the distinguishability part.

→ Every entity needs to have atleast one attribute so it can be distinguished from other entities.

→ Entity is of 2 types

Entity

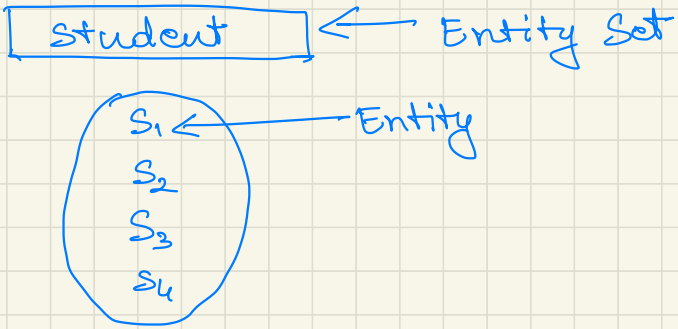
Tangible
- → Can touch
- → Physically exists
- → Eg: chalk

Intangible
- → Cannot touch
- → Logically exists i.e. logical entity.
- → Eg: Account number 1413279763

Entity Set

→ A collection of entities of the same type is called an entity set.

→ Entity sets are represented in the ER Model as:



Entity Attribute

→ Characteristics or properties of an entity which distinguish one entity from another.

→ Eg: Students have attributes:
- ⇒ Name
- ⇒ Roll no.
- ⇒ Mobile no.

→ Attributes are called columns in Table

→ In relational schema, we call them attributes and not columns.

→ In the foll. section, we define the diff. types of attribute.

→ An attribute can be of multiple types.

For ex: An attribute can be simple, single-valued and stored. Like a roll. no.

→ But an attribute cannot belong to conflicting types.

Eg: An attribute cannot be simple and composite.

Attribute types

Simple attribute

→ It cannot be broken down further

→ It is also called an atomic attr.

Eg: Gender.

Represented by oval

Gender

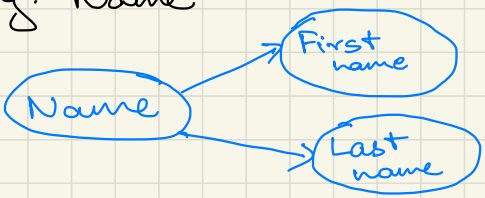
Composite attribute

→ Which can be broken down further

→ Also called inatomic

→ Represented by oval connecting with more ovals.

→ Eg: Name



Single value

→ Have only one value.

→ Represented by oval

→ Eg:

Aadhaar no.

Multivalued

→ Can have more than one value in the attribute.

→ Eg: mobile no.s, email ids

→ Represented by double ovals.

mobile no.

email

Stored value

→ Eg: DOB

→ This is value that needs to be stored.

→ Represented by oval

→ Eg: DOB

Derived value

→ Can be derived

→ Represented by dashed oval

→ Eg: Age

Age

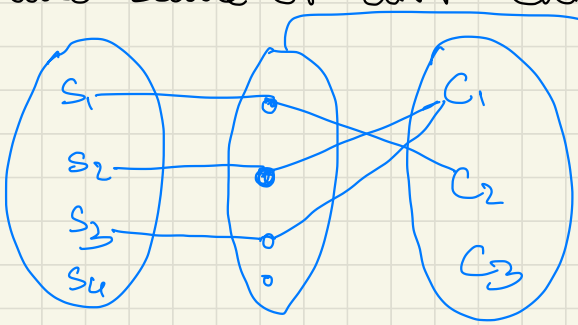
Note: Whether an attribute is simple or composite is dependant on our view of it.

Eg: Roll no. of the form 16H61A0505 can be considered atomic if we want to but if we want, we can also consider it to be composite.

Relationship b/w entities

Relation or association b/w 2 or more entities of the same or diff entity set.

Eg:



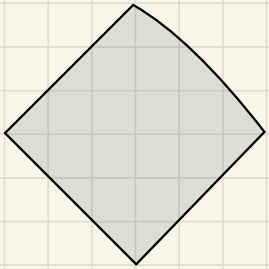
This entire thing is your relationship set while its constituent connections are your relations

Relationship set: A collection of similar type of relationships.

Note: A relationship represents the relation b/w 2 entities while a relationship set represents the relation b/w 2 entity sets. So, in ER model, we use relationship set.

Unlike in a function, in a relation it is not necessary that every element should participate in the relation. Eg: S_4 in the above example

In E-R model, relationship set is represented by diamond/rhombus.



Example ER-model

Let's construct the example ER Model for "students enrollment in courses"

A student entity set will probably have the foll:
- → Name (composite)
- → DOB
- → Age (derived)
- → Roll no.
- → Mobile no. (multivalued)

A course entity set will probably have the foll:
- → Name
- → Course Id

The relationship b/w the 2 will be through enrollment.

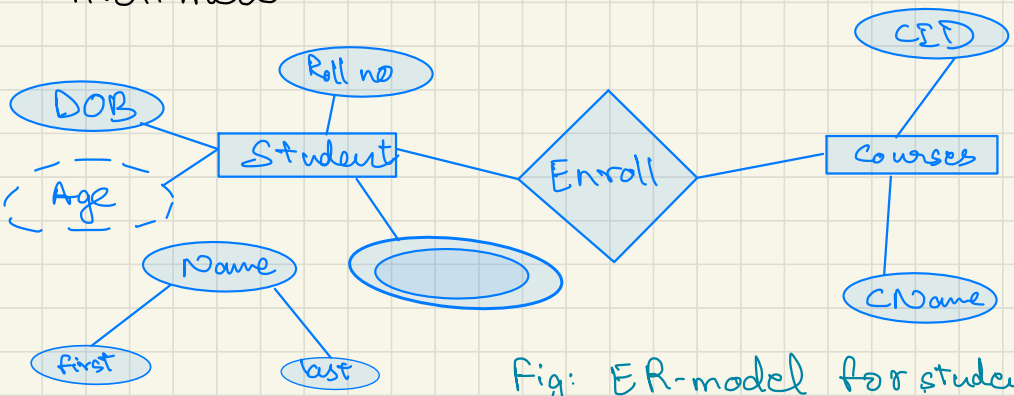
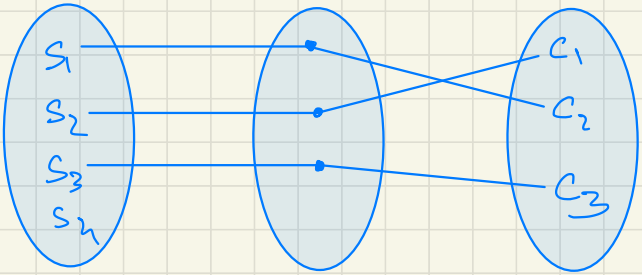


Fig: ER-model for student enrollment in courses

We can also create a set diagram for the above ER-model:



From the ER-model, a couple of things are not clear:

1) Is every entity in the "student" entity set supposed to enroll?

This issue is tackled by the participation constraint which is of 2 types:

a.) Total participation (all entities need to participate.)

b.) Partial participation (not all entities need to participate.)

2.) Can a single "student" entity enroll in multiple course entities?

This issue is addressed by cardinality ratios which have the foll. types:

a.) One to one
b.) One to Many
c.) Many to One
d.) Many to Many

Relationship set components

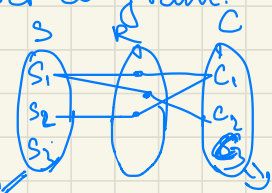
1) **Degree of relationship**: The no. of entity sets that the relation is connected to.

In the above example, the relation is connected to 2 entities. So, it's binary.

We can have unary (self join), binary, ternary .... n degree for a relationship.

2) **Cardinality ratio**: It show how many times (atmost) an entity can participate in a relationship.

set diagram:



ER model

atmost participation = 2 (many)

at most participation (1)

In ER model, we only deal with 2 values:

a) 1
b) Many

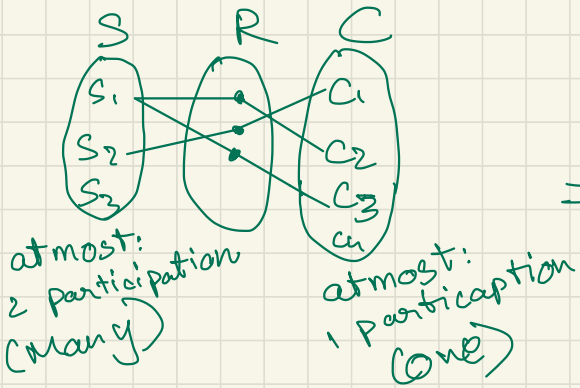
These values represent the atmost no. of relationships that an entity set's members can have.

Note that the keyword here is "atmost". It represents the maximum

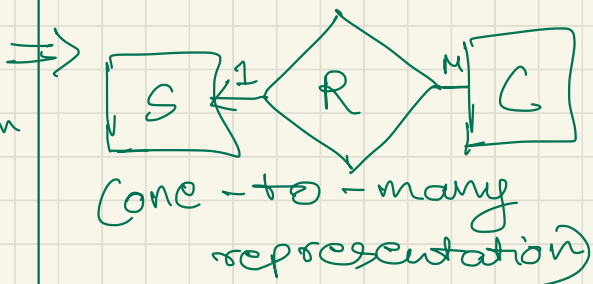
no. of relationships an entity can have. But the minimum no. of relationships is not defined. So, an entity can have 0 relationships as well.

Eg: In the above set diagram, S_3 has no relationships. Similarly, C_3 also does not have any relationships.

Note: When converting from a set diagram to an ER model, note that the representation of 1 and Many changes.



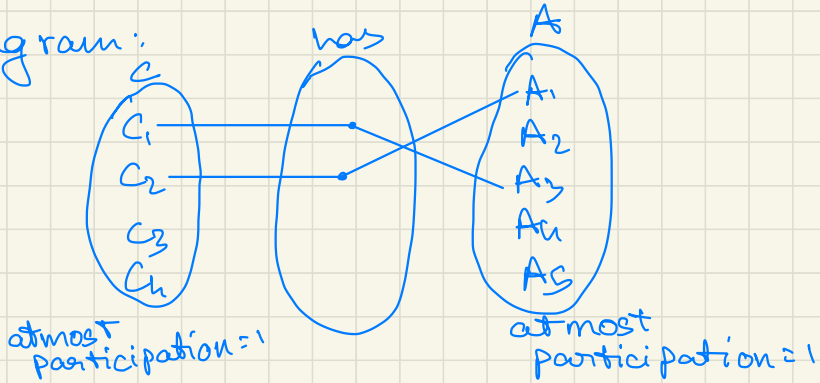
Convert to ER:



1:1 cardinality ratio:

Eg of this is a citizen-aadhar relationship

Set diagram:



Here, the atmost participation of every citizen is 1. Why? Since once a citizen has received an aadhar, they cannot have a 2nd aadhar.

It is valid that citizens like c_3, c_4 don't have aadhars (haven't applied for it) but cardinality only cares about atmost values.

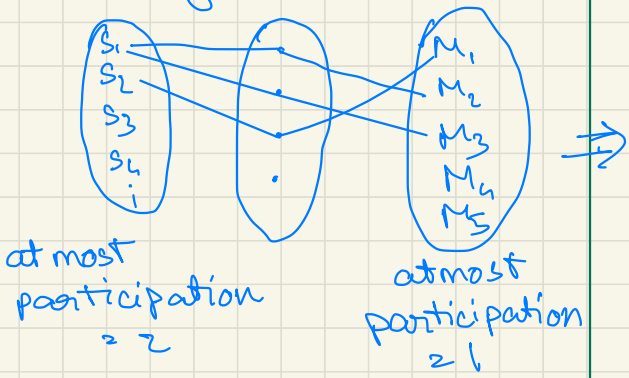
Similarly, not every aadhar that is generated needs to be assigned. But once assigned (participated), it cannot be assigned to another person as well.

ER model:

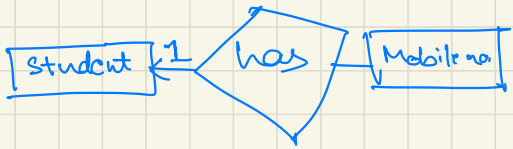
1-Many cardinality ratio:

Eg: Student have multiple mobile no.s:

Set diagram:



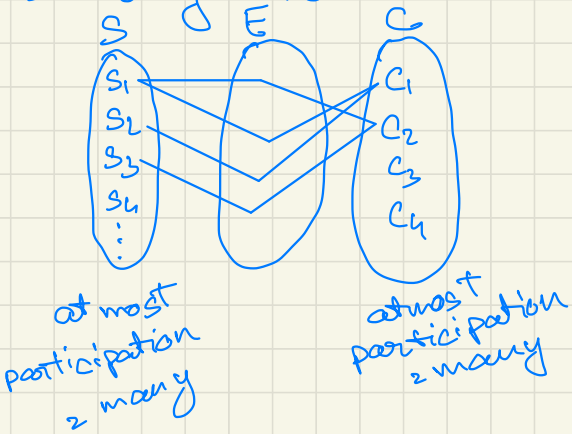
Convert to ER-model:



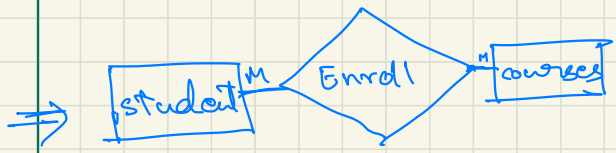
Many - Many cardinality ratio:

Eg. Student enrolled in courses

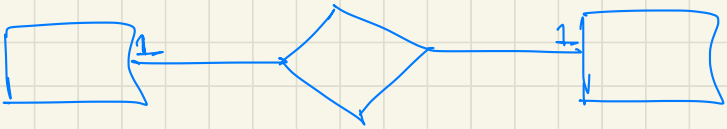
Set diagram:



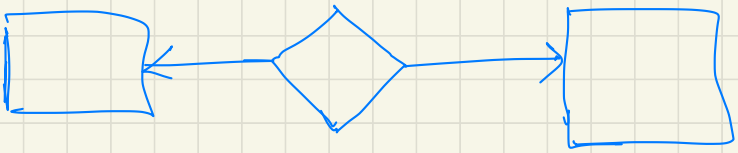
Convert to ER model:



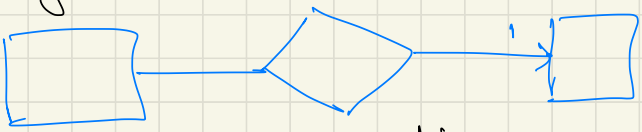
You can denote 1—1 relationship using an arrow or a 1



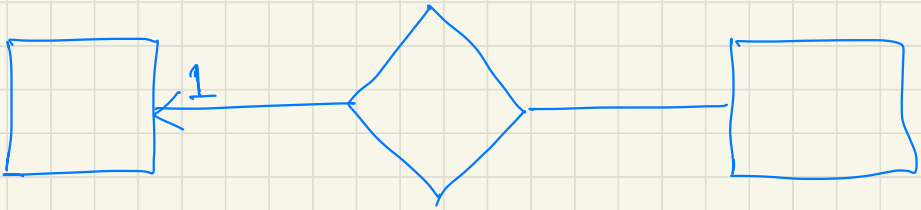
(or)



You can denote M-1 (or) 1-Many by not adding an arrow/1 on the many entity side

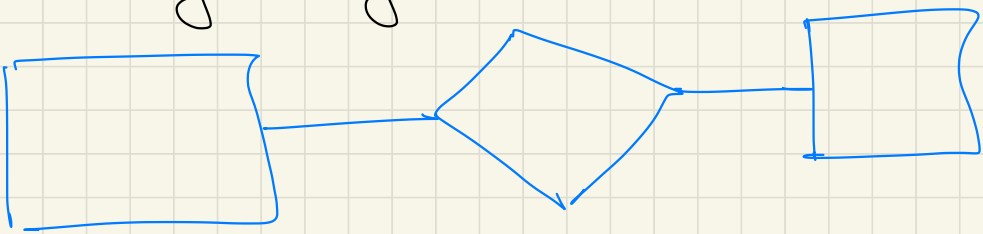


Many-1 ratio



1 - Many ratio

Many - Many can be denoted by not adding any arrows



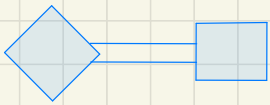
The student - courses ER model was depicting a Many - Many ratio.

Remember the notation as follows:

Many - no arrow/no.
One - add arrow/no.

3.) Participant Constraint

Total participation is denoted by double-line



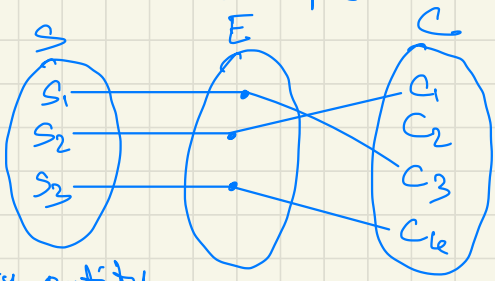
Partial participation is denoted by single line

Participation constraint tells us if every entity of entity set participates in relationship set.

If each entity has participated, it is total participation

If some entity is not participating then it is partial participation

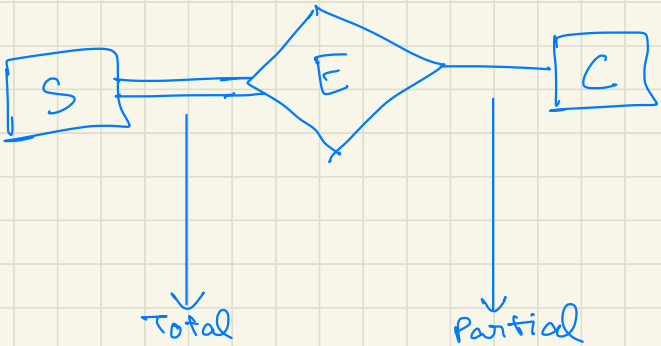
Student - course example



Every entity in S participated

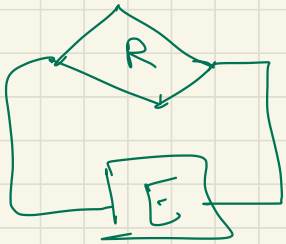
Some entity in C did not participate

Convert to ER model



Note: The degree of a relationship set is different from the degree of a relation. The degree of a relation denotes the no. of columns in it.

Unary relationship set:



Also called recursive relationship

Note: A relationship set cannot exist without an entity but an entity can exist without a relationship set.

Note: All the elements in the above ER models will be converted to tables.

Eg:

In the above ex, each entity set and the relationship set will be a table.

Since the relationship set is a table, it can also have its own attributes.
For ex: Date of enrollment.