Module 2

Entity –Relationship Modelling

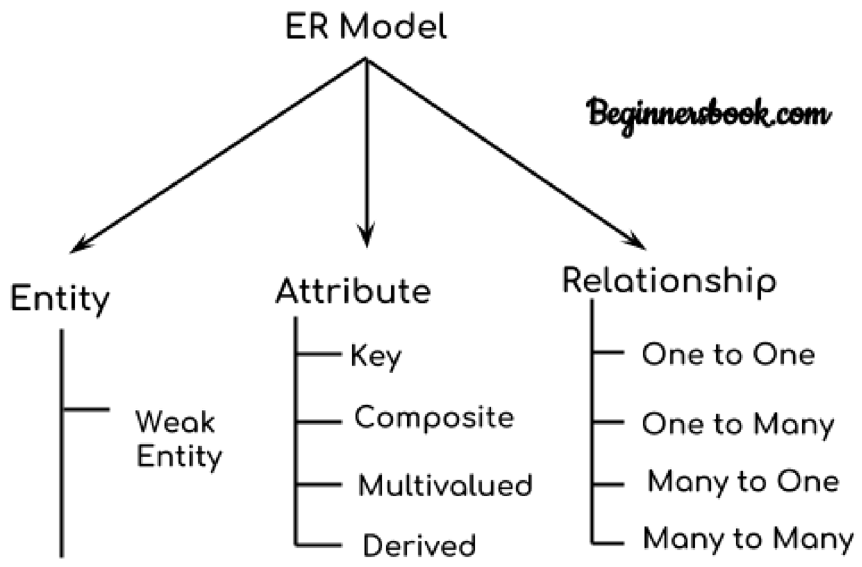
ER-Model [Entity-Relationship Model]

It is a popular high level conceptual data model. This model is used for the conceptual design of database applications

An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database.

ER Diagram

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.



Components of ER Diagram

ER model describes entities, relationships and attributes

Entity

An Entity may be an object with a physical existence – a particular person, car, house, or employee – or it may be an object with a conceptual existence – a company, a job, or a university course.

Attributes

Each entity has attributes. An attribute defines set of properties that are used to describe an entity. For example, an employee entity has attributes like employeeid, employee_name, designation, salary etc,,

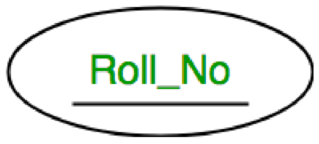
There are different types of attributes in ER diagram

- Key attributes
- Simple versus composite attributes
- Single valued versus multivalued attributes
- Stored versus derived attributes

Key Attributes

The attribute which **uniquely identifies each entity** in the entity set is called key attribute. Key attribute uniquely identify an entity, For example, Roll_No will be unique for each student.

In ER diagram, key attribute is represented by an oval with underlying lines.

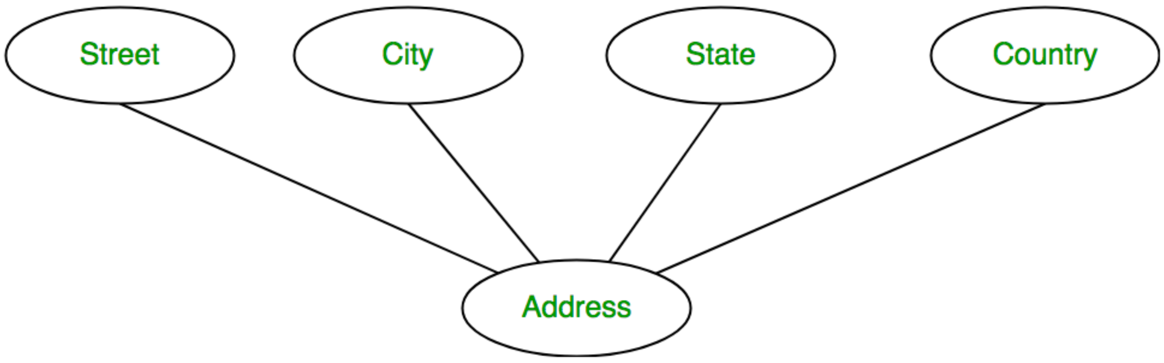


Simple versus composite attributes

Simple attribute is also called atomic attributes. The attributes that are not divisible are called simple attributes. Examples of simple attributes include registerno, voterid, aadharno etc

The attributes that are divisible are called composite attributes. Ie, the composite attributes can be divided into smaller sub-parts. The value of the composite attribute is the concatenation of the value of its component simple attributes

Example of composite attribute is the **address**. It can be further divided into house name, city name, district and pin code etc..student name is also a composite attribute. It can be divided into first name, middle name and last name. In ER diagram, composite attribute is represented by an oval comprising of ovals.

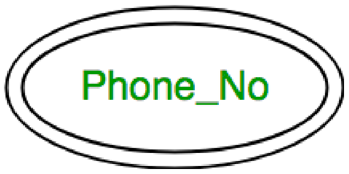


Single valued versus multi-valued attributes

Most attributes have a single value for a particular entity. Such attributes are called single-valued attributes, For example, age is a single valued attribute of a person, and Date-of-birth is also a single valued attribute

Some attributes have multiple values foe a particular entity. Such attributes are called multi-valued attributes. For example, degree is a multi-valued attribute of a person.ie some person may not have a degree, but someone has more than one degrees or others have only one degree. Phone number is also a multi-valued attribute because some people have more than one phone numbers

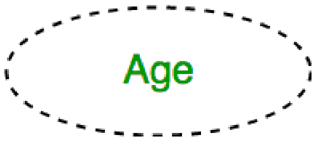
In ER diagram, multi-valued attribute is represented by double oval.



Stored versus derived attributes

In some cases, two or more attribute values are related. For example, age and date-of-birth of a person. Ie, we can determine the age of a person by knowing their date of birth and current date. Thus, the age attribute is called derived attribute and the date-of-birth is called stored attribute.

The derived attribute is derived from the stored attribute. In ER diagram, derived attribute is represented by dashed oval.



Complex Attributes

Composite and multi-valued attributes can be nested. We can represent such attributes by grouping the components using parenthesis () and separated by commas (,) and displaying between curly braces {}. Such attributes are called complex attributes

For eg if a person has more than one residence with different address

{

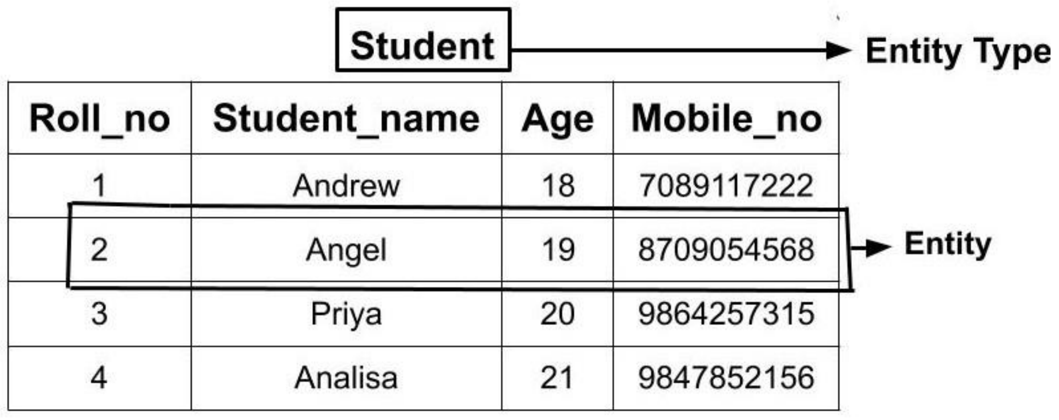
(housename1, place 1),(house name2,place2),...

}

Entity type and Entity sets

A database usually contains group of entities that are similar. For example, a company contains hundreds of employees that share the same attributes such as name, age, salary, designation etc.

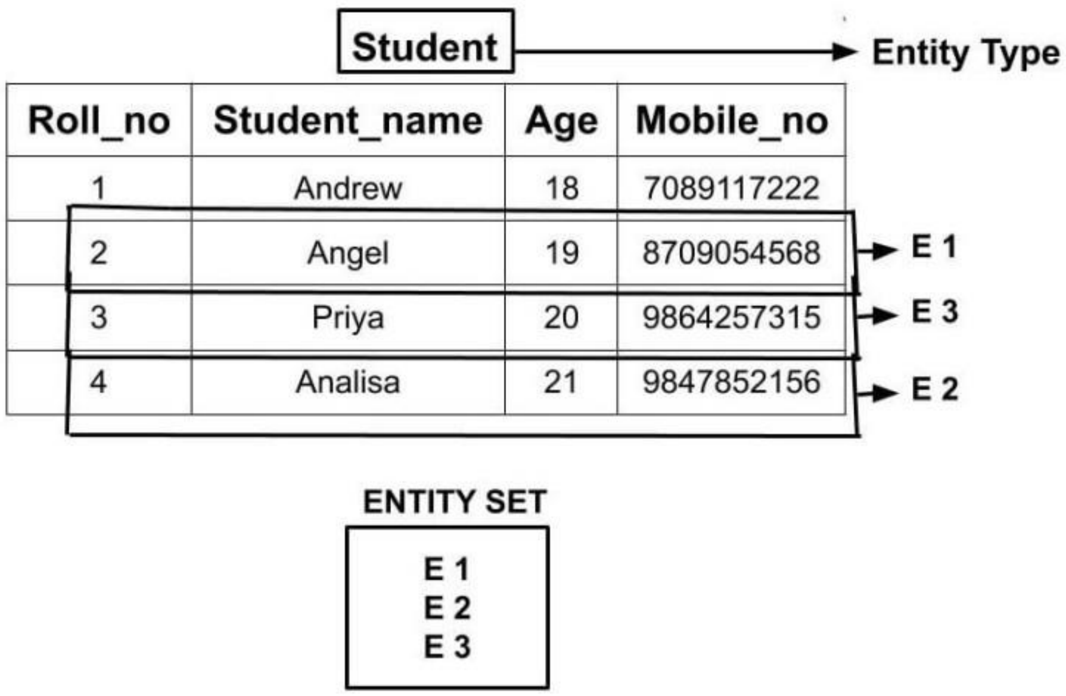
Collection of entities that have the same attribute is called entity types. Each entity type in the database is described by it's name and attributes



Roll_no	Student_name	Age	Mobile_no
1	Andrew	18	7089117222
2	Angel	19	8709054568
3	Priya	20	9864257315
4	Analisa	21	9847852156

The collection of all entities of particular entity type in the database at any point in time is called an entity set. The entity set is usually referred using the same name as the entity type

In the above example of STUDENT entity type, a collection of entities from the Student entity type would form an entity set. **We can say that entity type is a superset of the entity set as all the entities are included in the entity type**



Roll_no	Student_name	Age	Mobile_no
1	Andrew	18	7089117222
2	Angel	19	8709054568
3	Priya	20	9864257315
4	Analisa	21	9847852156

Weak Entity Types

The entity types that do not have key attributes are called weak entity types. In contrast, regular entity type must have key attributes. They are also called strong entity types

Entities belonging to weak entity types are identified by relating to specific entities from another entity type in combination with one of their attribute values. We call this other entity type as the identifying or owner entity type and the relationship between them is called identifying relationship

Relationship types and Relationship sets

Relationship

A relationship is an association between several entities. For example, consider the two entity types ‘customer’ and ‘account’. We can define a relationship in between these entities named ‘Owns’. It is a relationship that denote the association between ‘customer’ and their ‘account’.

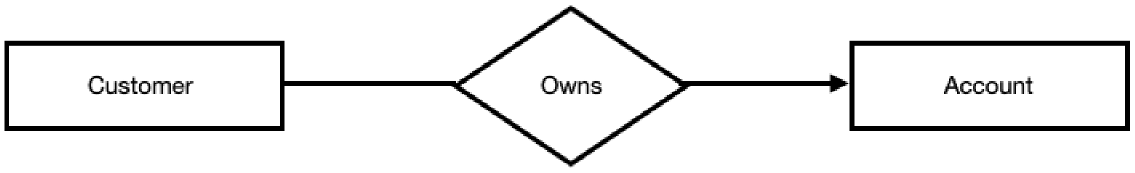
A relationship may also have descriptive attributes (composite attributes).

For example, ‘Owns’ may contain ‘last date’ attribute to denote the last date of account access

Relationship types

A relationship type defines a relationship set among the entities of certain entity types. The relationship type is illustrated in ER diagram with diamond symbol.

Following figure shows the relationship among two entity type ‘customer’ and ‘account’



Many to One Relationship

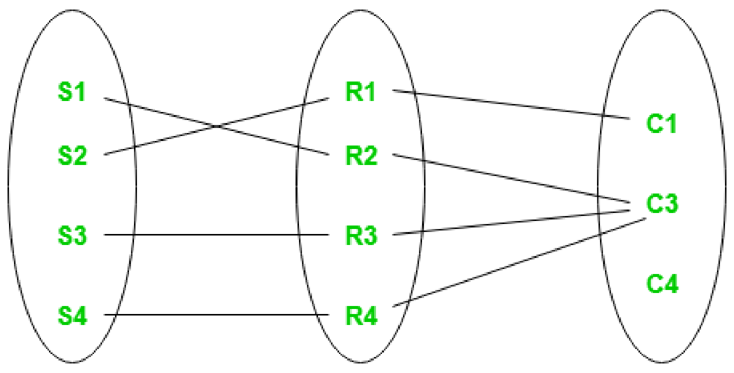
Relationship sets

Relationship set is a set of relationships of the same type. It is a collection of relationships all belonging to one relation type

Relationship Instances

The relationship set R of particular relationship type contains a set of relationship instances r_i, Where each r_i associates n individual entities e1,e2,e3 ….

For example, in a relationship type 'Enrolled in', the each enrolment of a student in a course is an instance of that relationship. Where 'student' and 'course' are entity types



Student Enrolled in course

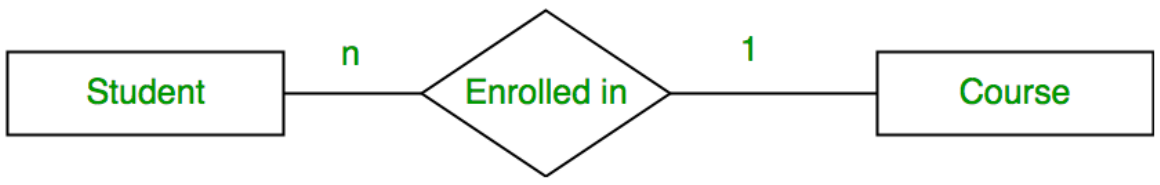
Relationship instances

S2----R1----→ c1

S1----R2----→c3

S3-----R3---→c3

S4-----R4--→c3



<u>Degree of relationship types</u>

The degree of relationship type is the number of participating entity type in a relationship. From the above example, the 'Enrolled in' relationship has the degree two. Ie, two entity types are participating 'student' and 'course'.

A relationship of degree two is called binary relationship and a relationship of degree three is called ternary relationship. Most commonly used relationship is the relationship with degree two. Higher degree relationships are generally complex than binary relationship

Constraints on Binary Relationship

There are two types of constraints

1. Participation constraints
2. Cardinality ratio

PARTICIPATION CONSTRAINTS

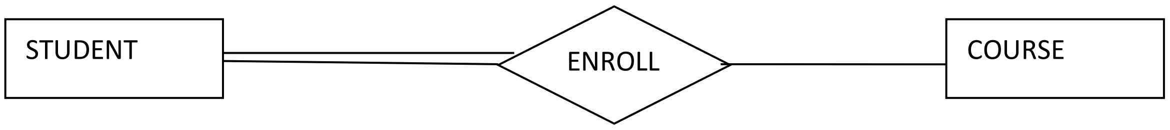
There are two kinds of participation constraints

1. Total participation
2. Partial participation

Total participation

When each entity in the entity set occurs in at least one relationship, then it is called total participation. In ER diagrams, total participation is displayed as a double line connecting the participating entities to the relationship.

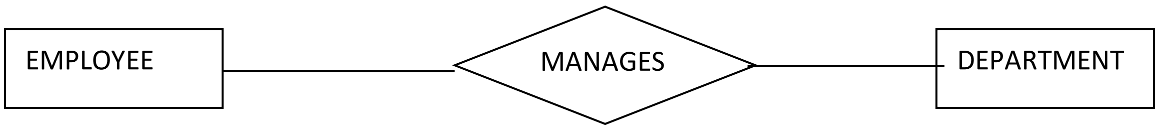
Total participation is also called existence dependency



Every student should enrolled in at least one course

Partial participation

When each entity in the entity set may not occurs in a relationship, and then it is called total participation. It is represented by single line in the ER diagram



Not all employees are the manages the department

Cardinality Ratios for binary relationship

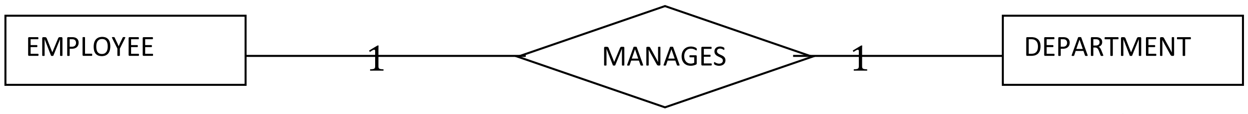
The cardinality ratios for a binary relationship specifies the maximum number of relationship instances that an entity can participate in.

Possible cardinality ratios for binary relationship are

1. One-to-one(1:1)
2. One-to-many(1:M)
3. Many-to-one(M:1)
4. Many-to-many(M:M)

One-to-one (1: 1)

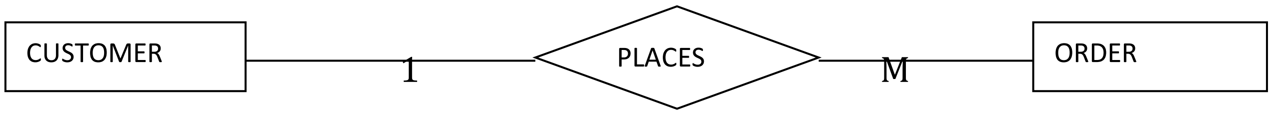
An entity is associated with at most one entity is called one to one relationship



One employee manages one department. There will be one manager for each department

One-to-many (1: 1)

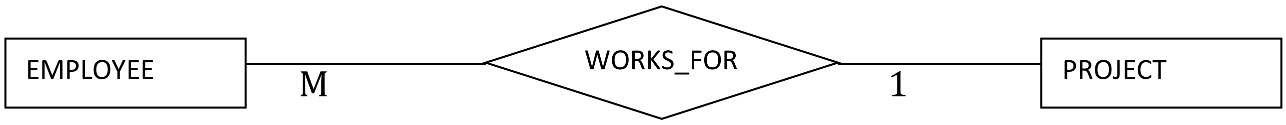
An entity is associated with any number of entities



An order is related to one customer and a customer can have any number of orders

Many-to-one(M:1)

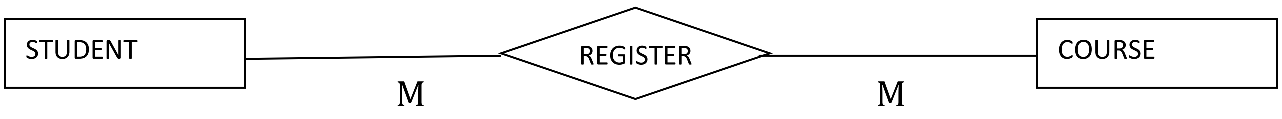
Here many entities are associated with one entity



Many employees are associated with one project

Many-to-many(M:M)

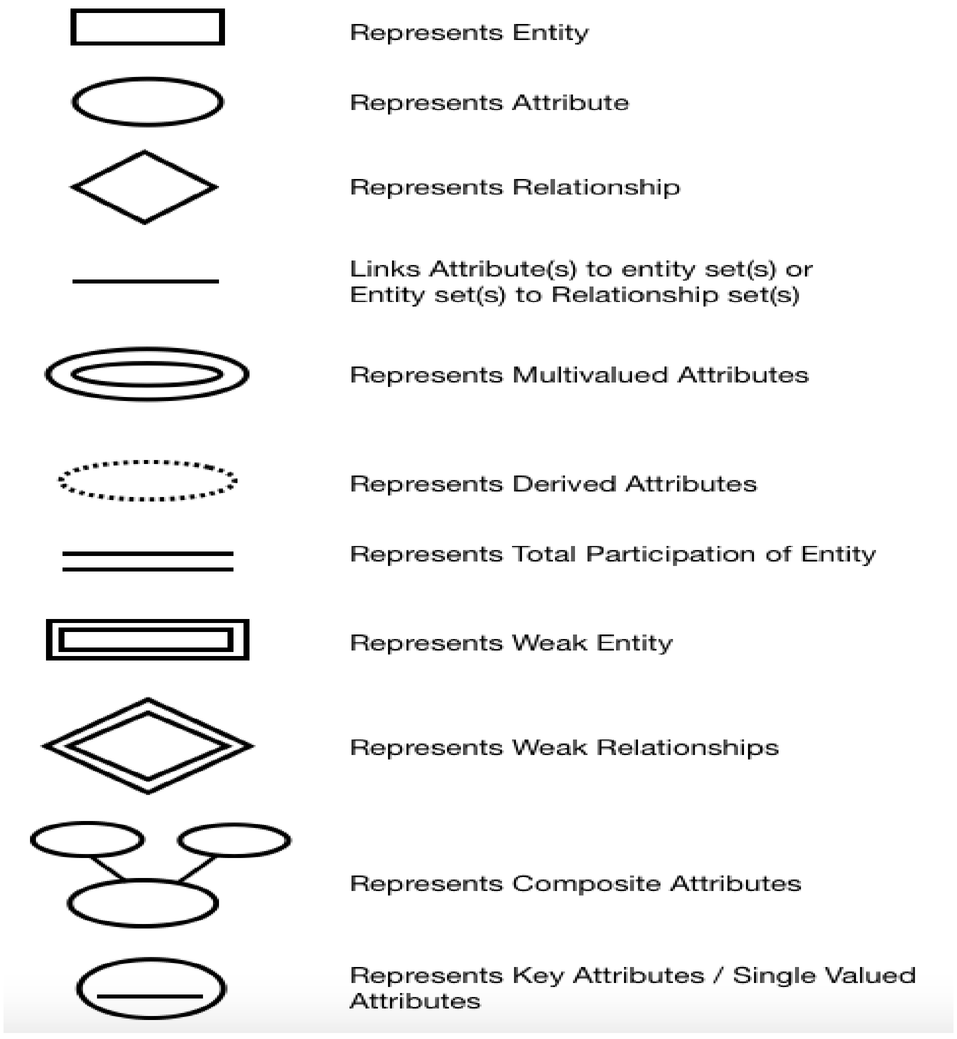
Many occurrences in an entity is related to many occurrences of another entity



Many student can register for many courses. Also a course contains many students

ER Diagram and its symbols

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.



Represents Entity

Represents Attribute

Represents Relationship

Links Attribute(s) to entity set(s) or Entity set(s) to Relationship set(s)

Represents Multivalued Attributes

Represents Derived Attributes

Represents Total Participation of Entity

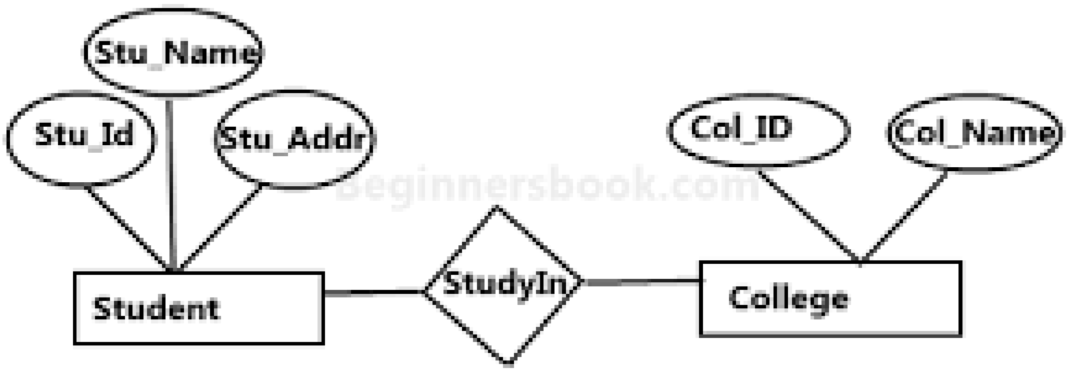
Represents Weak Entity

Represents Weak Relationships

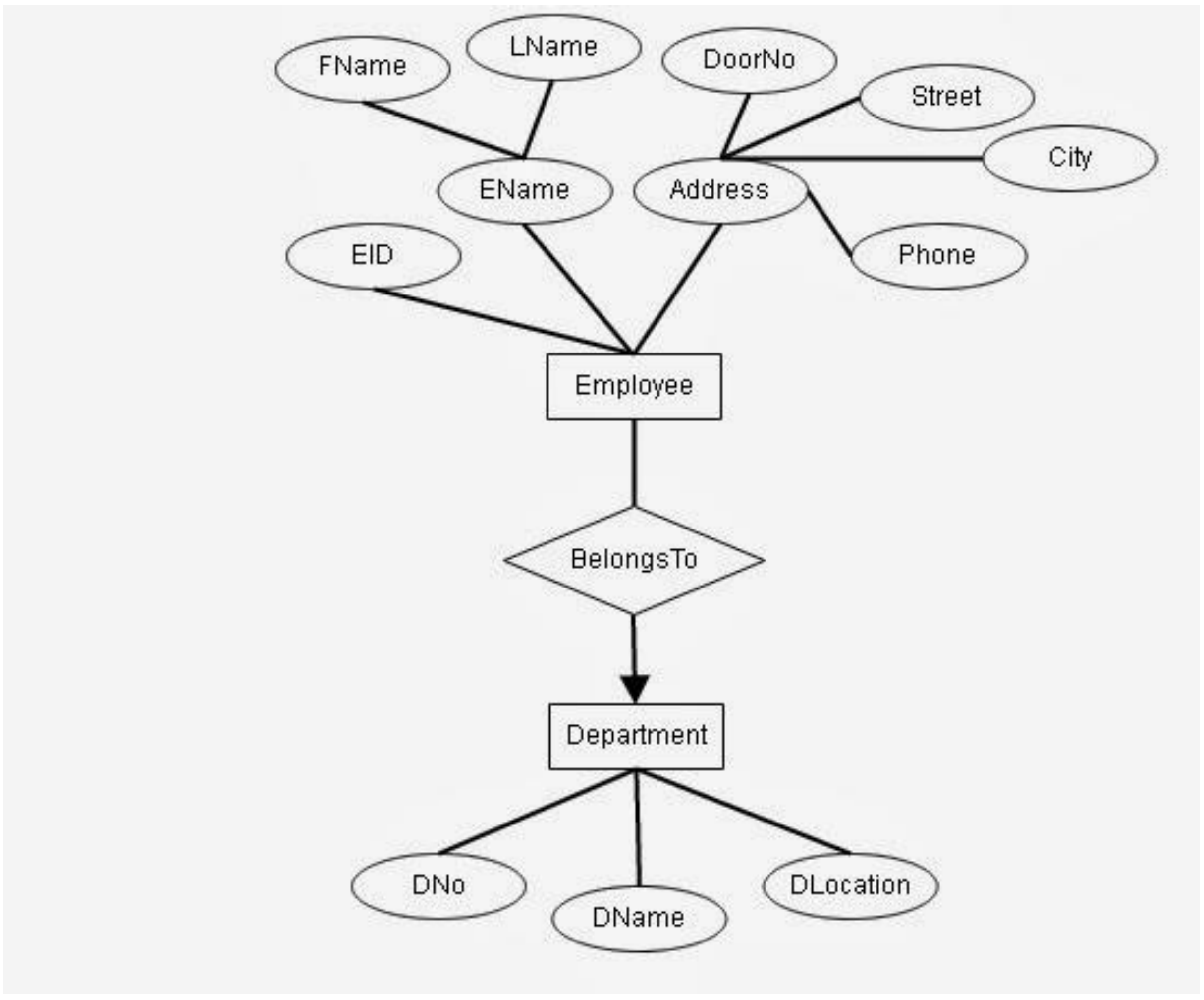
Represents Composite Attributes

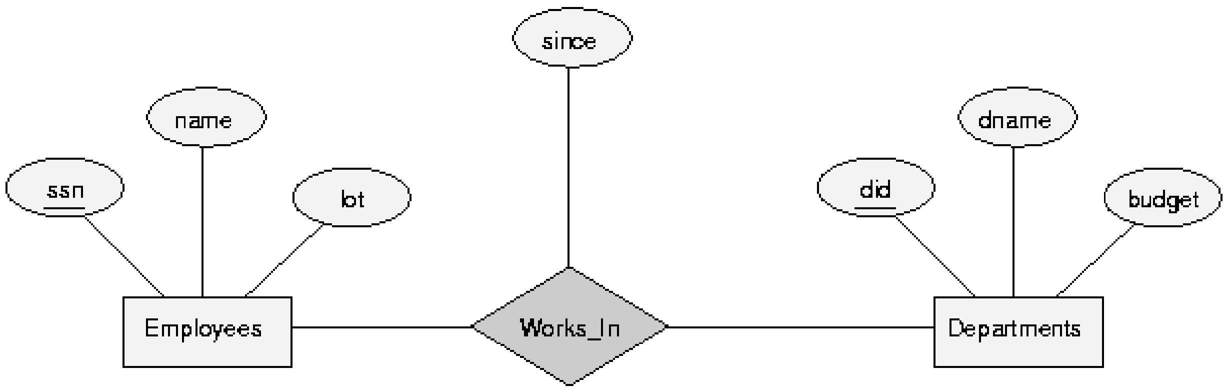
Represents Key Attributes / Single Valued Attributes

Examples of ER Diagrams



Sample E-R Diagram





Relational model concepts

Relational database consists of collection of tables. The relational model represents the database as a collection of relations. A relation is thought of as a table of values. Each row in the table represents a collection of related data values

Relational model VS Database concepts

- Relation ←→Table
- Tuple ←→row or record in the table
- Attribute ←→ column or field in a table
- Cardinality ←→ number of rows in a table
- Degree ←→ number of columns in a table
- Primary key ←→ unique identifier
- Domain ←→ pool of legal values for an attribute

Domain

A domain D is a set of atomic values. By atomic, mean that each value in the domain is indivisible. Domain specifies a data type. Examples of domain include

- Mob_no -- set of 10 digit phone number are valid
- Names – the set of character strings
- Emp-ages ---possible age of employee in a company must be between 18 and 60(example only)

Attributes

In the context of relational database, attribute is the name of the role played by some value. Ex: name,designation etc

Definition

The domain of attribute A is denoted as dom(A).

A relation schema R, denoted by R (A1,A2…..An) is made up of a relation name R and a list of attributes A1,A2,……An. Each attribute Ai is the name of a role played by some domain D. Thus D is called domain of Ai denoted by dom(Ai).The degree or arity of a relation is the number of attributes in a relation

Tuple

One row of information in a table is called tuple, ie, one record

Table name: employee $\longrightarrow$ relation name

attribute

Empid	Emp_name	Experience	Salary
1	Akhil	2	20000
2	Meena	3	27000
3	Wilfred	1	14000

tuples

Characteristics of Relations

1. **Ordering of tuples**

   A relation is a set of tuples. There is no order associated with the tuples in the relation. Ie ordering of tuples within the relation indicates first, second , third ……. Up to the last record in the table

2. **Ordering of values with in a tuple**
   A relation schema R={ A1,A2…….An} is a set of attributes and a relation state r(R) is a finite set of mappings r={t1,t2,……..tn} where t1,t2….tn is a set of tuples

   According to the above definition a tuple can be considered as a sety of (<attribute>,<value>) pairs, where each pair gives the value of the mappings from an attribute Ai to a value Vi from dom(Ai)

3. **Values and NULLs in the tuples**

   NULL values are used to represent the values of attribute that may be unknown or may not apply to a tuple.
4. **Interpretation of a relation**

   Each relation can be viewed as a predicate and each tuple in that relation can be viewed as an assertion for which that predicate is satisfied for the combination of values in it

<u>Relational model constraints and relational database schemas</u>

Relational model constraints are mainly divided into three main categories

1. Constraints that are inherent in the data model are called inherent model based constraints or implicit constraints
2. Constraints that can be directly expressed in the schemas of the data model is called schema based constraints or explicit constraints
3. Constraints that cannot be directly expressed in the schemas of the data model and hence must be expressed and enforced by the application programs. These are called application based or semantic constraints or business rules

Schema based constraints include

- Domain constraints
- Key constraints
- Entity integrity constraints
- Referential integrity constraints

<u>Domain constraints</u>

Domain constraints specify that within each tuple, the value of each attribute A must be an atomic value from the domain dom(A)

It specify the data types associated with domains. They include

- Numeric data type for integers. Ie short integer, integer, long integer etc
- Data type for real numbers. Ie float and double
- Data type for characters
- Data type for fixed and variable length strings
- Boolean type

Key constraints

Key constraints include key,s that are associated with the database tables. Key's are used to easily identify any row of data in a table

Student

roll_no	Sname	Phone	age
1	Anu	9847123456	17
2	Anju	9895983451	18
3	Varun	9991008112	18
4	Appu	9961002345	17

1. **Super key**

   A super key (SK) specifies a uniqueness constraints that no two distinct tuples in any state of relation R can have the same value for SK
   Or
   Super key is the combination of more than one attribute that is used to uniquely identify every tuple in the relation

In the above table super key include

Sk = { roll_no,(roll_no,sname),phone}

A super key also have redundant attributes. Here in (roll_no, sname) , sname is a redundant attribute

2. Candidate key

Candidate keys are defined as the minimal set of fields which can uniquely identify each record in the table

The value of the candidate key is unique. There can be more than one candidate key in a relation

In the above example

Candidate key = {roll_no,phone}

The set of keys which have a chance to become the primary key are called candidate keys

3. Primary key

There can be more than one candidate key in a relation, out of which one can be chosen as the primary key

Primary key uniquely identifies each record in a table. It must have unique values and cannot include NULL values

In the above example, roll_no and phone are treated as candidate keys. But from them, roll_no is chosen as a primary key of that relation

4. Composite key

If a primary key contains two or more attributes which are used to uniquely identify any record in a table is called composite key

5. Alternative key

The candidate key which are not selected as primary key are called alternative keys or secondary keys

In the above example, roll_no and phone are treated as candidate keys. But from them, roll_no is chosen as a primary key of that relation. Then the phone attribute is called alternative key

6. Non-key attributes

The attributes of a table other than the candidate key attributes are called non-key attributes

7. Non-prime attributes

The attributes of a table other than the primary key attribute is called non-prime attributes

Entity integrity and referential integrity constraints

Entity Integrity

The entity integrity constraints states that no primary key value can be NULL. This is because the primary key value is used to identify individual tuples in a relation. Having NULL values for the primary key implies that we cannot identify that tuple

Referential Integrity constraint

Referential Integrity constraint is specified between two relations and is used to maintain the consistency among tuples in the two relations. Foreign key is used for this purpose

Foreign key

Foreign key is an attribute of one relation R1 whose values are required to match those of the primary key of some relation R2

We refer to the relation that contains the foreign key as the referencing relation and the relation that contains the corresponding primary key is called referenced relation

For example consider the following table

Employee & department

Department

Did	Dname	Dfloor
101	Sales	First floor
102	Promotion	Second floor
103	HR	First floor

Employee

Eno	Ename	Salary	Did
1	wilfred	20000	101
2	arvind	15000	102
3	meena	18000	103
4	anu	18000	101

In employee table, the attribute Did is used to represent the department in which that employee belongs. Here in this table, Did is called foreign key. Because Did is the primary key of department table.

The relation employee is called referencing relation and the relation department is called referenced relation