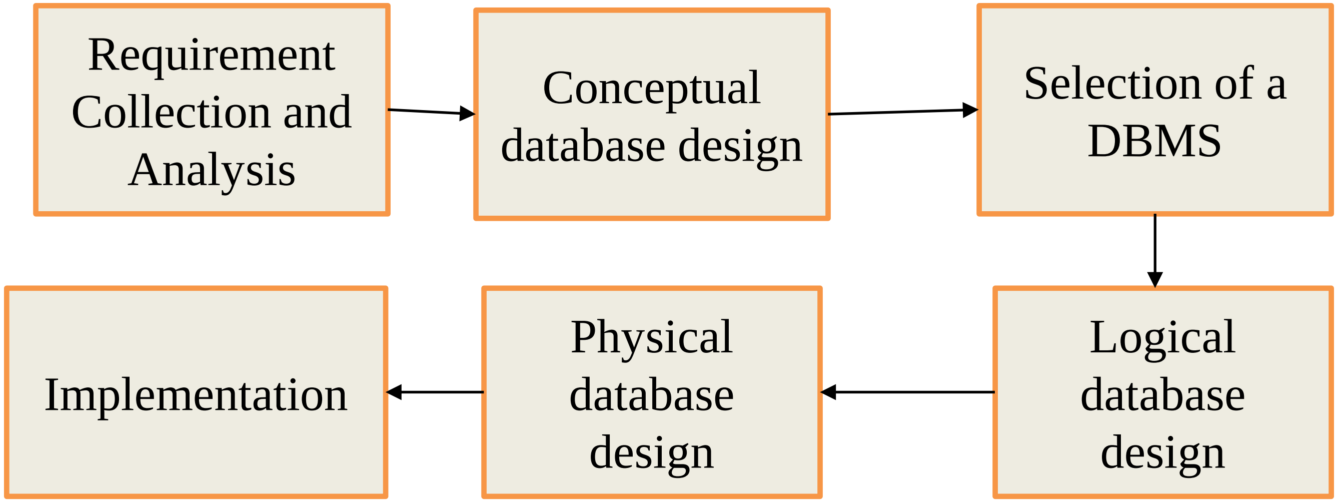
UNIT II Data Modeling using Entity-Relation Model

Database Design Process

- DBMS is designed to manage large volume of information.
- Database design is a collection of processes that facilitate the designing, development, implementation and maintenance of database systems.
- DBMS is designed to meet the requirements of the users.
- It has high performance.

Following processes are needed for database design:



Database Design Process

1. Requirements collection

- ➢ Database designers interviews to database users for their requirements.
- ➢ It also can be divided into two parts:

a. Planning: it mainly concern with planning of entire database development life cycle.

b. System definition: it defines the scope and boundaries of proposed database system.

- Functional requirements:
- ➢ It consists of user defined operations.
- ➢ Data requirements: different types of data that are required to model database.

2. Conceptual design:

- ➢ it concisely or briefly describe the data requirements and their types, relationship and constraints.
- ➢ It does not describe the implementation detail and non technical person can also understand and no redundant data.
- ➢ ER diagram is used.

Database design process

3. Logical database design

- ➢ It creates well structured tables, describes attributes ,their types and relationship among tables.
- ➢ Actual implementation of the database is done in logical database design.
- ➢ Implementation details are used.

4. Physical database design

- ➢ Main purpose of this design is to optimize performance, to ensure data integrity by avoiding redundant data.
- ➢ Describes the detail of how data are stored, defines tables and relationship among tables.

Entity Types and Entity sets

- Entity: It is something which has real existence. For example: Tuple 1 contains information about Aarav(Id, Name, and Age) which has existence in real world, so tuple 1 is an entity.

Sita

ENTITY

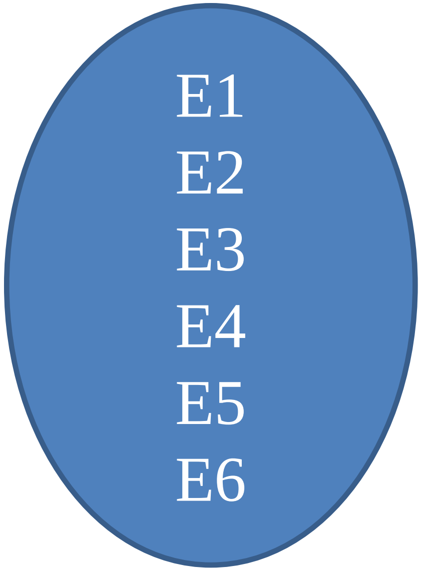
Entity Type: It is a collection of entities having common common attributes. For example: Student

Student

Entity Type

Entity Set

- It is same as an entity type, but defined at a particular point in time. For example student enrolled in a class on the first day.



E1
E2
E3
E4
E5
E6

Entity set

Attribute

- Attributes: descriptive properties possessed by an entity set. E.g. student has name, age, address

Types:

- Simple attribute − Simple attributes take atomic values. E.g. student's roll no.
- Composite attribute − Composite attributes are made of more than one simple attribute. E.g. student's complete name may have first_name and last_name.
- Single-value attribute − Single-value attributes contain single value. For example − Social_Security_Number
- Multi-value attribute − Multi-value attributes may contain more than one values. For example, a person can have more than one phone number, email_address
- Derived attribute: whose values are derived from other attribute present in the database, e.g.. Avg(salary)

Attribute

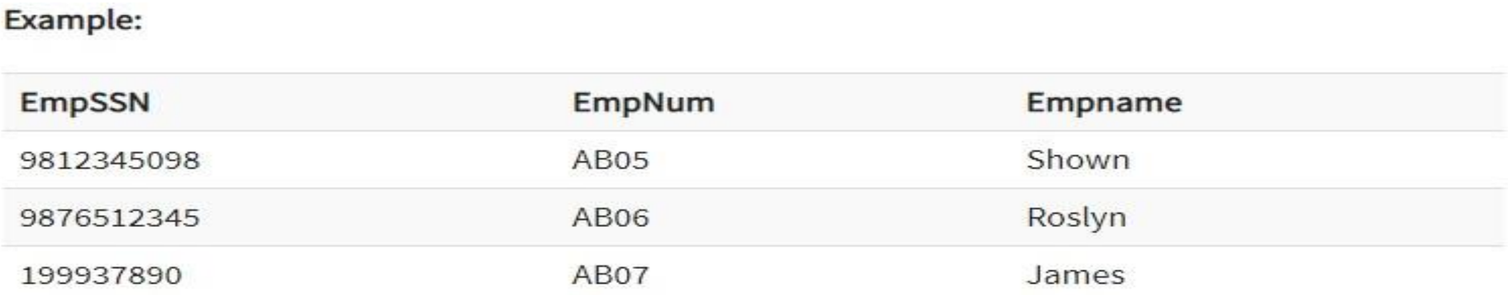
- Stored attribute: it is such type of attributes which is stored physically in the database. For example: Name, address, roll no attributes can not be derived from any other attributes.

Keys

- A **DBMS key** is an attribute or set of an attribute which helps you to identify a row(tuple) in a relation(table).
- They allow you to find the relation between two tables. Keys help you uniquely identify a row in a table by a combination of one or more columns in that table.

1. Super key:

A **superkey** is a group of single or multiple keys which identifies rows in a table. A Super key may have additional attributes that are not needed for unique identification.



Example:

EmpSSN	EmpNum	Empname
9812345098	AB05	Shown
9876512345	AB06	Roslyn
199937890	AB07	James

Keys

- It is superset and uniquely identify tuples.
- In the above-given example, {EmpSSN} and {EmpNum},{ EmpNum, EmpName}, { EmpSSN, EmpName} etc are superkeys.

2. Candidate key: Minimal subset of super key. It can not be null and can be more than one candidate key.

For example: EmpNum, EmpSSN.

Key

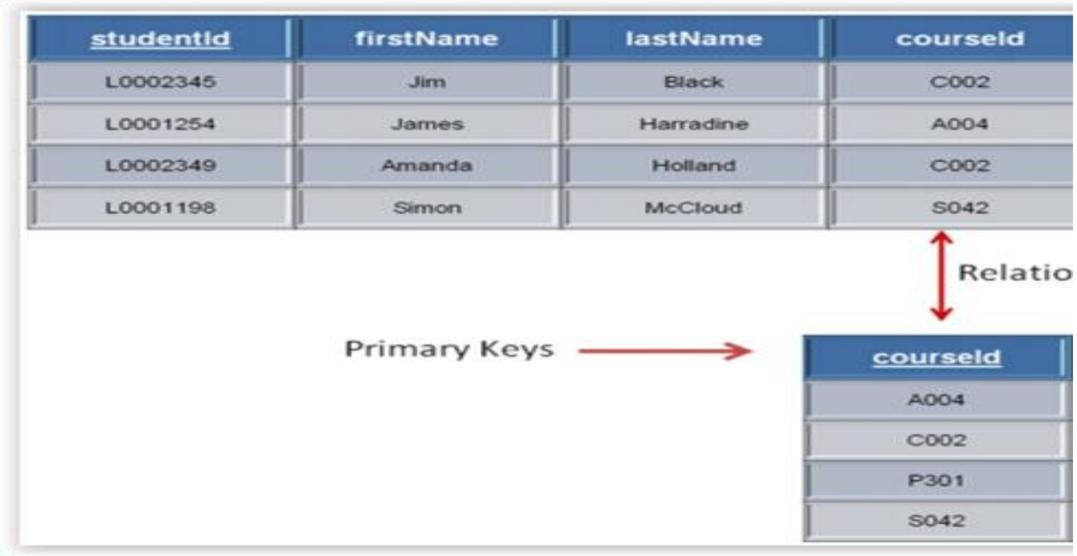
3. Primary Key: it is also candidate key which is most appropriate to become the main key. Can not be null and duplicate value.l

For example: {EmpNum}

4. Composite key: It consists of two or more attributes that uniquely identify any record in a table is composite key. For example: {**EmpNum} is primary key and { EmpNum, EmpName} is composite key.**

Key

5. Foreign key: Foreign keys are the columns of a table that points to the primary key of another table.



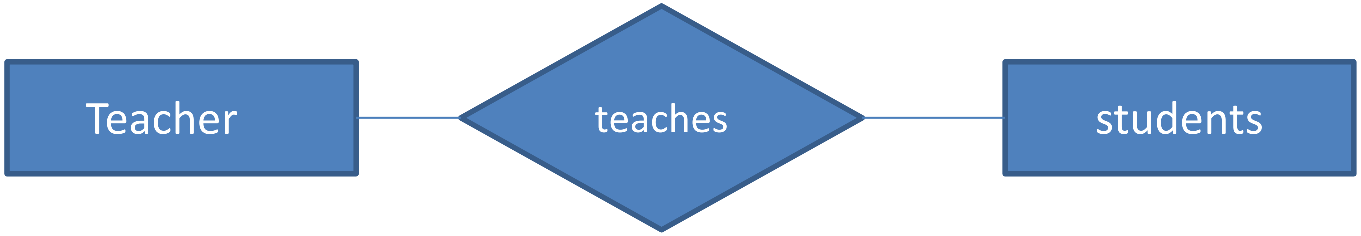
studentId	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042

Relatio
Primary Keys

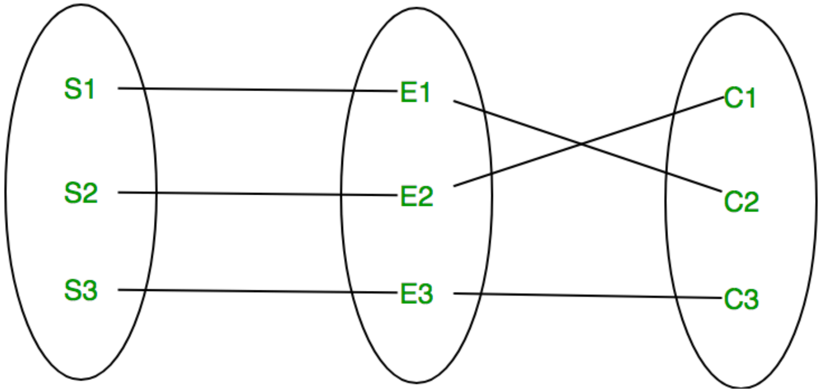
courseId
A004
C002
P301
S042

Relationship Type

- Relationship type: It represents the association between entity type. For example, ‘teaches’ represents relationship type. In ER diagram, relationship type is represented by diamond shape and connecting the entities with lines.

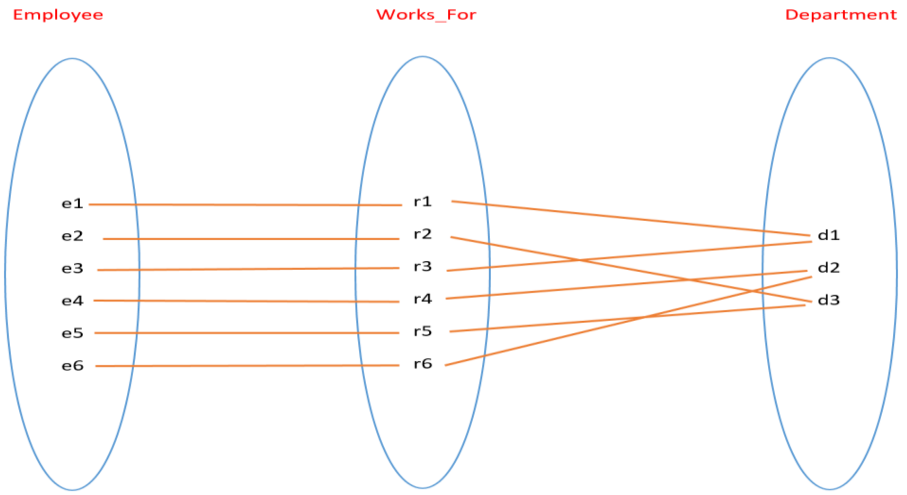


- Relationship set: It is the association between two entities sets is called relationship type.



Relationship instance

- A relationship instance in R is an association of entities, where the association includes exactly one entity from each participating entity type.



Relationship instance and degree of a relationship type

Degree Of a relationship

- Number of entities-sets that participate in a relationship set is called degree of the relationship set. On the basis of degree, relationships can be divided as below:

1. Unary relationship
2. Binary relationship
3. N-ary relationship