INTRODUCTION
TO
DATABASE MANAGEMENT SYSTEMS

Ongole Campus

Rajiv Gandhi University of Knowledge Technologies

What is data?

Small named entity in the real world.

What is database?

- Database is the collection of data/ information which is interrelated.

(or)

- Database is collection of data that is managed by DBMS.

What is DBMS?

- DBMS –

  Data Base Management Systems.

- It is a software.

- Helps to manage large amount of data.

Database Management Systems

A Data Base Management System (DBMS) is software that is designed to assist in maintaining and utilizing the large collection of data/ database.

Management involves

1. Defining structure to the data/information.
2. Providing mechanism for manipulation of data/ information.

Examples:

1. University database.
2. Enterprise information.
3. Online ticket reservation systems.
4. Telecommunication.
5. Banking and Finance.
6. Online shopping systems, etc...

Ways to store and manage large amount data:

1. File systems
2. DBMS

File systems vs database systems:

For each and every requirement programmers need to write an individual program which makes the programmers difficult where as it is not the case in database systems.

Disadvantages of File system

1. Data redundancy and inconsistency.
2. Difficulty in accessing data.
3. Data isolation.

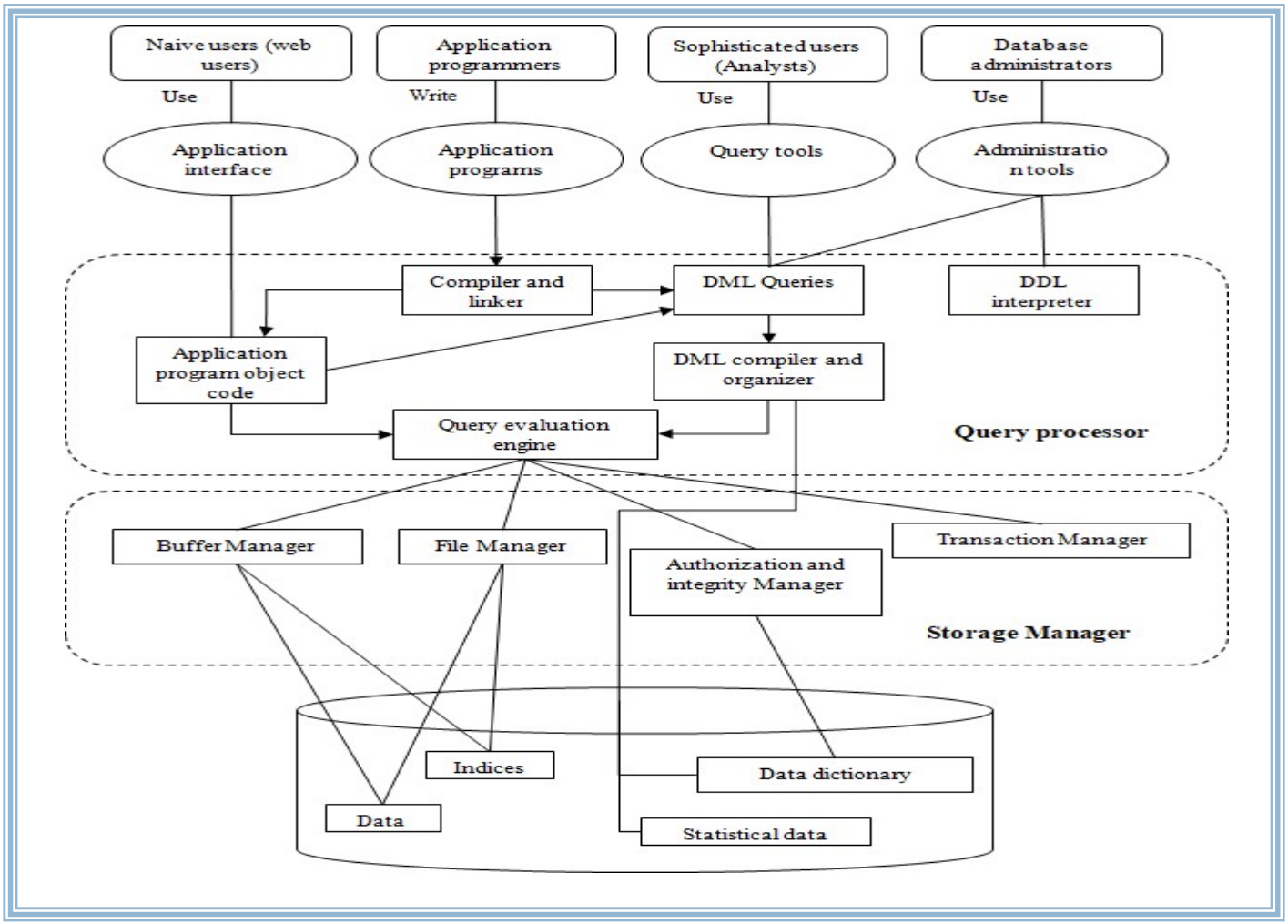
Disadvantages of File system(Cont . .)

4. Integrity problems.
5. Atomicity problems.
6. Concurrent access anomalies.
7. Security problems.

Advantages of Database system

1. Data independence – Three layers of abstraction.
2. Efficient access data.
3. Data integrity and security.
4. Concurrent access and crash recovery.
5. Reduced application development time.

Structure of Database systems



Components of Database System

- Users
- Query processor
- Storage manager
- Database

Users:

- Naive users (Web users).
- Sophisticated users (Analyst).
- Application programmers.
- Database Administrators.

Query Processors:

- Application program Object Code.
- Compiler and Linker.
- DML compiler.
- DDL interpreter.
- Query Evaluation Engine.

Storage Manager:

- File manager.
- Buffer manager.
- Authorization and integrity manager.
- Transaction manager.

Database:

- Physical / actual data storage device.
- Indices.
- Data dictionary.

View of Data

- There are three levels.
  - Physical level
  - Logical level
  - View level

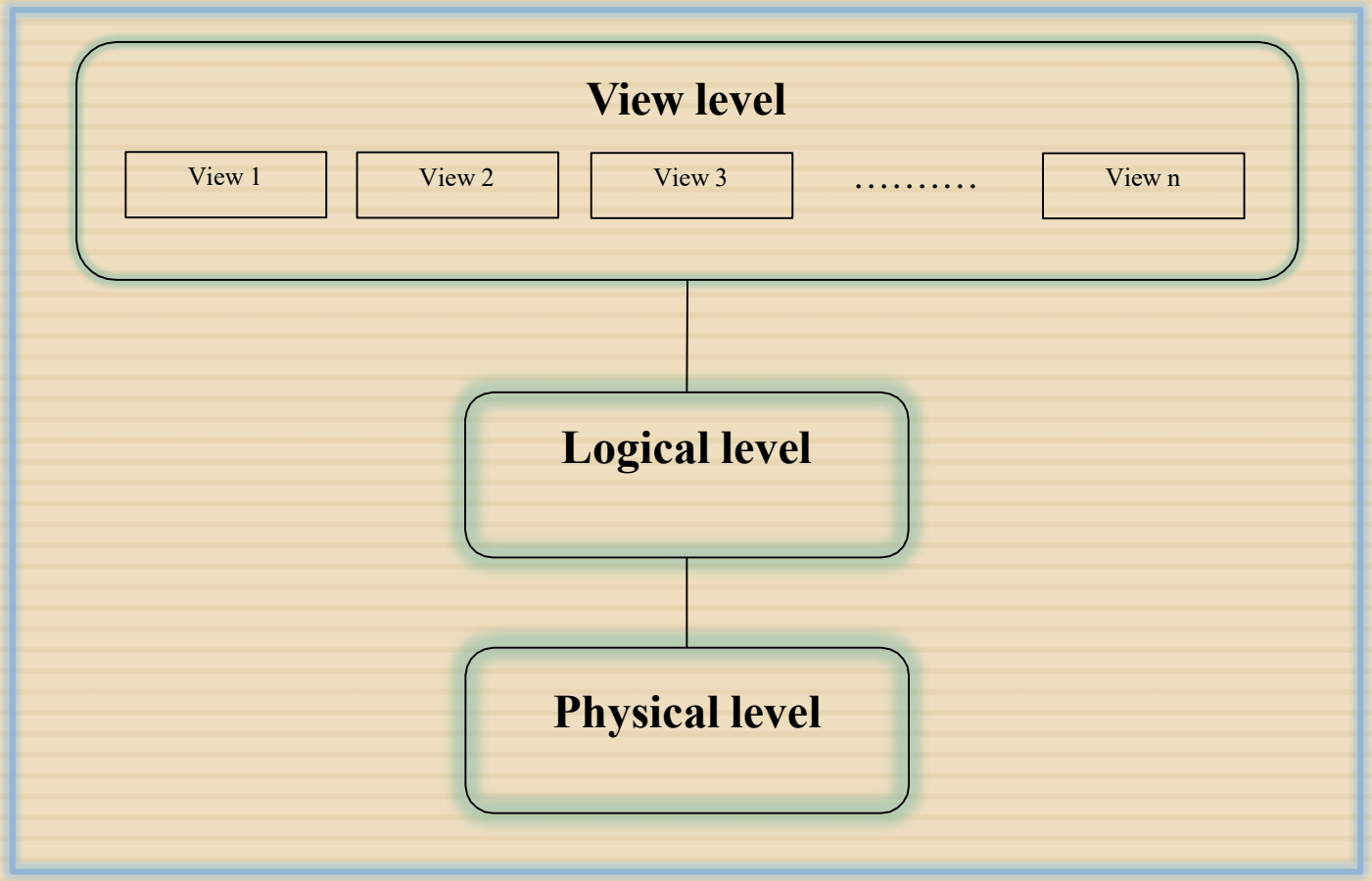


Figure - 1.2 Three level of data abstraction.

Physical level abstraction

- Lowest level of abstraction.
- Describes low-level complex data structures.
- Describes ***how*** the data is stored in memory.

Logical level abstraction

- Known as “conceptual level”.
- Builds above the physical level.
- Describes
  - ***What*** data are stored in memory.
  - ***What*** relations exist among the data.
- It incorporates physical data independence.

Physical data independence

- Changes in physical level design will not reflect logical level design.

View level abstraction

- Known as “external level”.
- Builds above the logical level.
- Describes *which* data can be viewed by the user.
- It allows data access as
  - Customized
  - Authorised

View level abstraction (Cont....)

- Makes easy interaction of system with users.
- Single database systems can have many views.
- It incorporates the logical data independence.

Logical data independence

- Changes in logical level design will not reflect view level design.

What is Data model?

The collection of conceptual tools for describing data, data relationships, data semantics and consistency constraints to design database is called as data models.

Categories of Data model

- High level/conceptual models.
- Low level/ physical data models.
- Representational/ implementation models.
- Higher level implementation model.
- Semi structured data models.

High level/conceptual models

- Provide the concepts that are used to access the data by user.
- To represent the data these models use
  - Entity – Represents real world object/concept.
  - Attribute – Property that describes an entity.
  - Relationship – Association between two or more entities.
- Example: Entity Relationship model (ER model).

Low level/ physical data models

- Provide the concepts that describe the data storing in storage devices.
- These models are meant for specialists.
- They describe that how the data is stored in computer.
- Represent the information as
  - record formats.
  - record orderings.
  - access paths.

Representational/ implementation models

- Provide the concepts which are easily understood by the end users.
- Represent the data in the form of record structures.
- Also known as record based data models.
- Frequently used in traditional and commercial DBMSs.
- Example: Relational data model, network model and hierarchical model.

Higher level implementation model

- Implementation models which are closer to high level data models.
- Example: Object based data model, object relational data models.
- Object based data model – It is extension of ER model with Object Oriented concepts like encapsulation, methods etc.
- Object relational data model – It is combination of object model with relational data model.

Semi structured data models

- These are self describing data models.
- It permits the specify the data where individual data items of the same type may have different sets of attributes.
- Example:
  - XML (Extensible Markup Language).

Database languages

- DML – Data Manipulation Language.
- DDL – Data Definition Language.
- DSL – Data Storage Language.
- VDL – View Definition Language.

Data Manipulation Language

- Enable to manipulate the data.
- manipulation includes
  - Retrieving the data.
  - Inserting the data.
  - Deleting the data.
  - Updating the data.

Data Manipulation Language (Cont...)

- Two types:
  - Procedural DML.
  - Non-procedural DML.

Data Manipulation Language (Cont...)

Procedural DML:

- Low level DML.
- User should mention *what* data need to access and *how* to access.
- It must embed with general purpose programming language.

Data Manipulation Language (Cont...)

Procedural DML (Cont...):

- It retrieves individual records and process them.
- They need looping constructs to retrieve and process each record from set of records.
- Known as record-at-a-time DML.

Data Manipulation Language (Cont...)

Non procedural DML:

- High level DML.
- Known as declarative DML.
- User need to mention *what* kind of data required, No need of how.
- Can retrieve many records at a time in a single statement.
- Known as set-at-a-time (or) set oriented DML.

Data Definition Language

- Implements the database schemas.
- Also it facilitates adds the additional properties to the data base.
- Before modifying the data the system checks the constraints.

Data Definition Language (cont...)

- Additional properties:
  - Domain constraints.
  - Referential integrity constraints.
  - Assertions.
  - Authorizations.

Data Definition Language (cont...)

Domain constraints:

- A domain of possible values must be associated with every attribute.
- these are most elementary form of integrity constraint.
- Example: integer types, date/time types.

Data Definition Language (cont...)

Referential integrity constraints:

- Value that appears in one relation for a given set of attributes also appears in a certain set of attributes in another relation.

Data Definition Language (cont...)

Referential integrity constraints (Cont...):

- Posses parent and child relation.
- The child table should contain the records which are referred from the parent tables.
- Orphan records are not entertained.

Data Definition Language (cont...)

Example for referential integrity constraints:

- Departments table (Parent table)
  - SNo, Department name, department code.
- Students table (Child table)
  - SNo, Student name, id, department name.

Data Definition Language (cont...)

Example for referential integrity constraint(Cont...):

Department table:

SNO	DEPARTMENT NAME	DEPARTMENT CODE
1	CSE	01
2	ECE	02
3	CHEM	03
4	ME	04
5	CIVIL	05
6	MME	06

Data Definition Language (cont...)

Example for referential integrity constraint(Cont...):

Student table:

SNO	ID	STUDENT NAME	DEPARTMENT NAME
1	1234	RAVI	CSE
2	1235	RANI	ECE
3	1236	ANVESH	CSE
4	1237	ANSHU	MECH
5	1238	BHANU	CIVIL
6	1239	AKASH	EEE

Data Definition Language (cont...)

Assertions:

- Condition that should always satisfied by database system.
- Domain and referential integrity constraints are special kind of assertions.

Data Definition Language (cont...)

Example for Assertions:

- Every department should offer at least 4 subjects in every semester.

Data Definition Language (cont...)

Authorisations:

- Type of access they are permitted on various data in the database.
- There are various types of authorizations for various user.

Data Definition Language (cont...)

Authorisations (Cont...) :

- Various authorizations:
  - Read authorization.
  - Insert authorization.
  - Update authorization.
  - Delete authorization.

Data Definition Language (cont...)

Authorisations (Cont...) :

- Administrator can assign any one, none or all kind of authorizations to a user.

Data Definition Language (cont...)

- The output of DDL is metadata.
- Meta data – information about the data.
- Meta data is stored in the data dictionary.
- Data dictionary – Special table that can only be accessed and updated by the database system itself.

Data Storage Language

- Used to specify internal schema.
- Relational data models have no specific language.
- The internal schema is specified by a combination of functions, parameters, and specifications related to storage of files.

View Definition Language

- Used to specify user views and their mapping to conceptual schema.
- In most of DBMSs, DDL is used define both logical schema and external schema.

Database Design

- Designing includes
  - Design of database schema.
  - Design of program that access and update the data.
  - Design of security scheme that control the data.
- Needs of users plays vital role in the design process.

Database Design (Cont...)

- Steps that involves in database design
  - Requirements gathering phase –
    - It is the initial phase.
    - Designers gathers the application needs by interacting with users.

Database Design (Cont...)

- Steps that involves in database design (Cont...)
  - Conceptual design phase –
    - Chooses a high level data models.
    - Converts the requirements in to conceptual schema with the help of conceptual tools in the chosen data models.
    - Here the main focus is on describing the data and their relationships rather than specifying physical storage details.
    - Example: ER model.

Database Design (Cont...)

- Steps that involves in database design (Cont...)
  - Specification of functional requirements –
    - Indicates functional requirements of the enterprise.
    - Users describe the operations on data.
    - Example: searching, updating, retrieving etc.
    - Designer ensures whether the functional requirements are accomplished or not.

Database Design (Cont...)

- Steps that involves in database design (Cont...)
  - Logical design phase –
    - Translating high level conceptual schema in to implementation data model.
    - Example for implementation data model is relational model.
    - i.e Translation from high level ER model to relational data model.

Database Design (Cont...)

- Steps that involves in database design (Cont...)
  - Physical design phase –
    - Resultant schema in logical design phase is used.
    - Physical features of database are specified.
    - Example features like file organization, index structures etc.

Entity Relationship model

- It is high level data model.
- Used to design the conceptual schema.
- Provides the graphical / diagrammatic representation of schema.

Entity Relationship model

- Yields Entity – Relationship diagram.
- An "entity" is a *thing* or an *object* real world which is *distinguishable* from other objects.

Entity Relationship model

- For an example in an educational institute
  - Student is an entity.
  - Department can be an entity.
  - Faculty is an entity.

Entity Relationship model

- An entity may have set of properties which describes the entity.
- These properties are also known as *attributes*.
- Example:
  - Id, name, dept are the attributes of a student.
  - Emp_id, name, dept, salary are the attributes of an employ.

Entity Relationship model

- Association among one or more entities is known as "*relationship*".
- Example:
  - Teacher is the relation that associates teaching faculty with the student.

Entity Relationship model

- Basic Elements –
  - Entity sets.
  - Relationship sets.
  - Attributes.

Entity Relationship model

Entity Set:

- An "*entity set*" is a set of entities of the same type that shares same properties/attributes.
- Example:
  - Set of *students* are entity sets.
  - Set of *courses* are entity sets.

Entity Relationship model (Cont...)

Entity Set (Cont...):

- Entity sets need not to be disjoint.
- Example:
  - College system may have following entity sets
    - Invigilators entity set
    - teaching_faculty entity set
    - non_teaching_faculty entity set

  All these entity sets may have common entities.

Entity Relationship model

Relationship Set:

- The set of relationships of same type of same types is known as "*relationship set*".

Entity Relationship model

Attributes:

- The property which describes an entity is known as attribute.
- Example:
  - *id_number, name, department are the* attributes for the entity student.
  - Emp_id, emp_name, department, exp are attributes for an employ.

Entity Relationship model

Attributes (Cont...):

- Attributes are descriptive properties of individual members in entity sets.
- They represent or gives information about the entities.

Entity Relationship model

Attributes (Cont...):

- Attributes can be ***simple*** or ***composite***.
- Simple – cannot be sub divided.
  - Example: id_number, department
- Composite – can be divided further.
  - Example: name, address, etc.
    - Name - first_name, middle_name, last_name.
    - Address - dno, street, town, mandal,district,state,pin.

Entity Relationship model

Attributes (Cont...):

- Attributes can be ***single valued*** or ***multi valued***.
- Single valued – it has only one value.
  - Example: "id_number" for a student.
- Multi valued – can have many values.
  - Example: "teaching_subjects" in teaching faculty.

A faculty may teach multiple subjects.

Entity Relationship model

Attributes (Cont...):

- Attributes can be ***derived***.
- The values of these attributes can be derived from other relative attributes.
- Example:
  - No_of_students is an attribute that represents how many students a teaching faculty teaches. This value can be found by counting no.of student entities associated with her.

Entity Relationship model

Attributes (Cont...):

- Attributes can be ***derived***.
- The values of these attributes can be derived from other relative attributes.
- Example:
  - If Service and joing_date are the attributes of an employ, then the *service* can be derived from joing_date by coutning the no. of years from current date..

Entity Relationship model

Attributes (Cont...):

- Attributes can be ***derived***.
- The values of these attributes can be derived from other relative attributes.
- A derived attribute is not stored, it just computed whenever required.

Entity Relationship model

Attributes (Cont...):

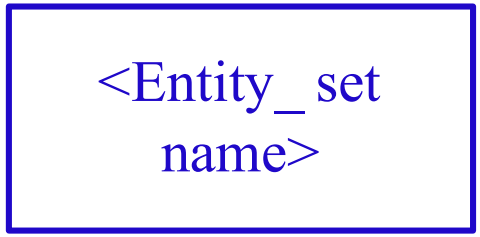
- Attributes may takes **NULL** value whenever
  - The entity doesn't have any value for it.

    (or)
  - Not applicable
- Example: For a student phone_number may not be available.

Constructing ER diagrams

Basic Components representation:

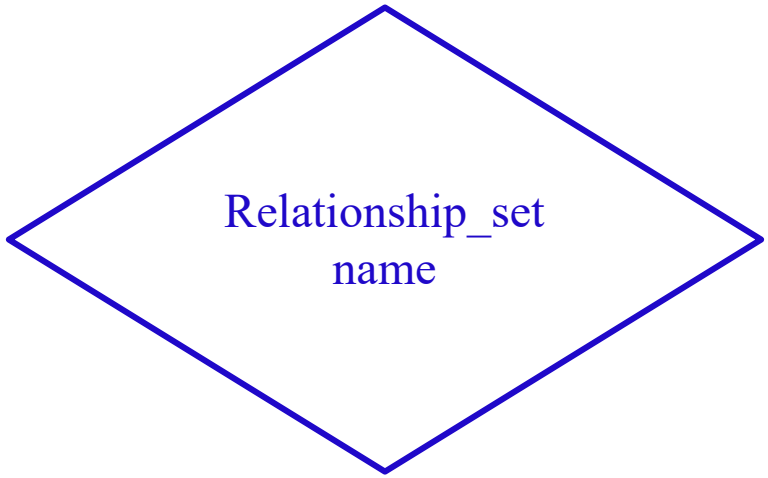
- Entity set – Rectangle.



Constructing ER diagrams

Basic Components representation:

- Relationship set – Rhombus.



Constructing ER diagrams

Basic Components representation:

- Attribute – Oval.

Constructing ER diagrams

Basic Components representation:

- Derived Attribute – Oval.

Constructing ER diagrams

Basic Components representation:

- Multi valued Attribute – Oval.

Constructing ER diagrams

Basic Components representation:

- Linking – Straight line.

Constructing ER diagrams

Example:

Draw an ER diagram to represent a ***students*** entity set that has Id_number, Name, Dept, Contact as attributes.

Constructing ER diagrams

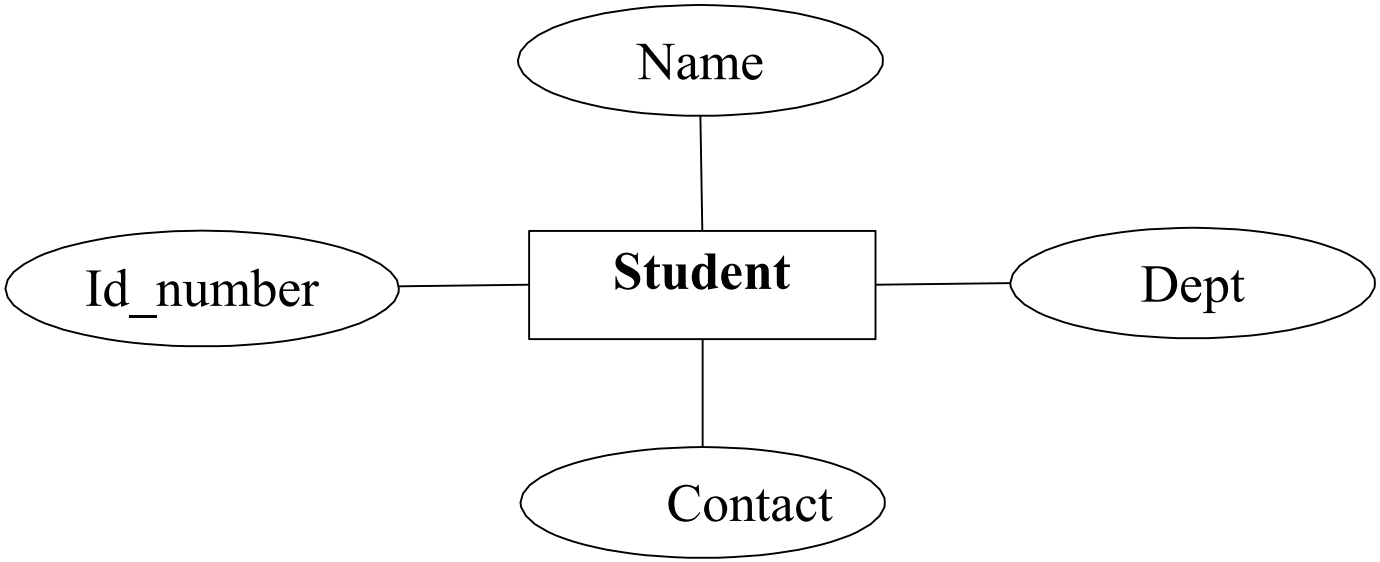


Figure - 1.3 E-R diagram for student entity set

Constructing ER diagrams

Example:

Draw an ER diagram to represent ***employs*** entity set that has Emp_id, Name, Dept, Contact, Service, Joing_date, Salary as attributes.

Constructing ER diagrams

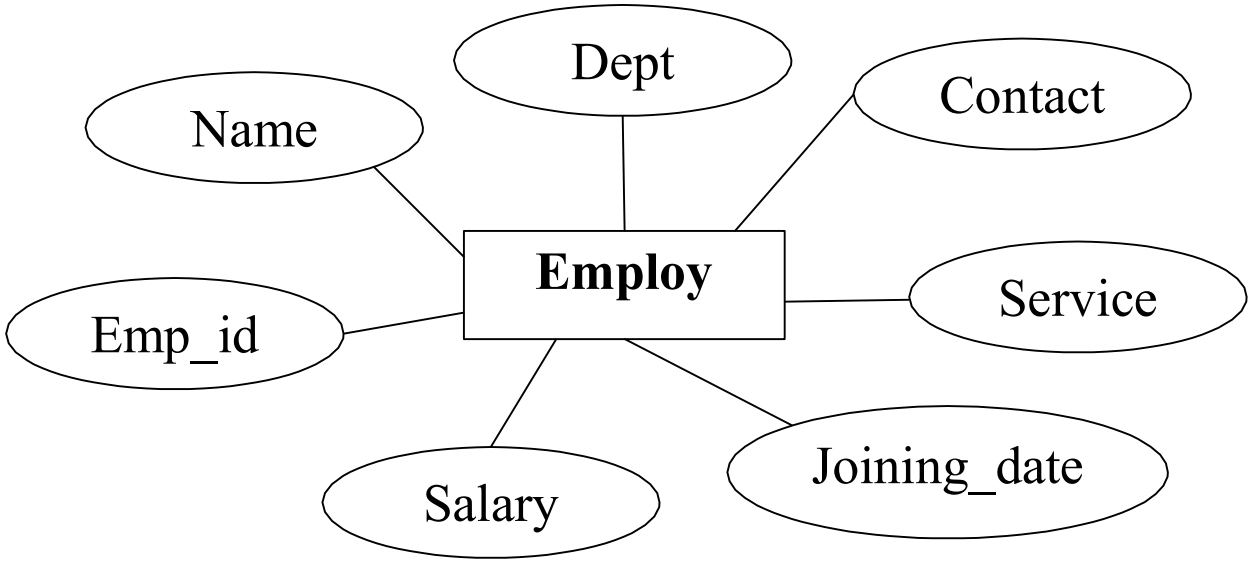


Figure – 1.4 E-R diagram for employs entity set

Constructing ER diagrams

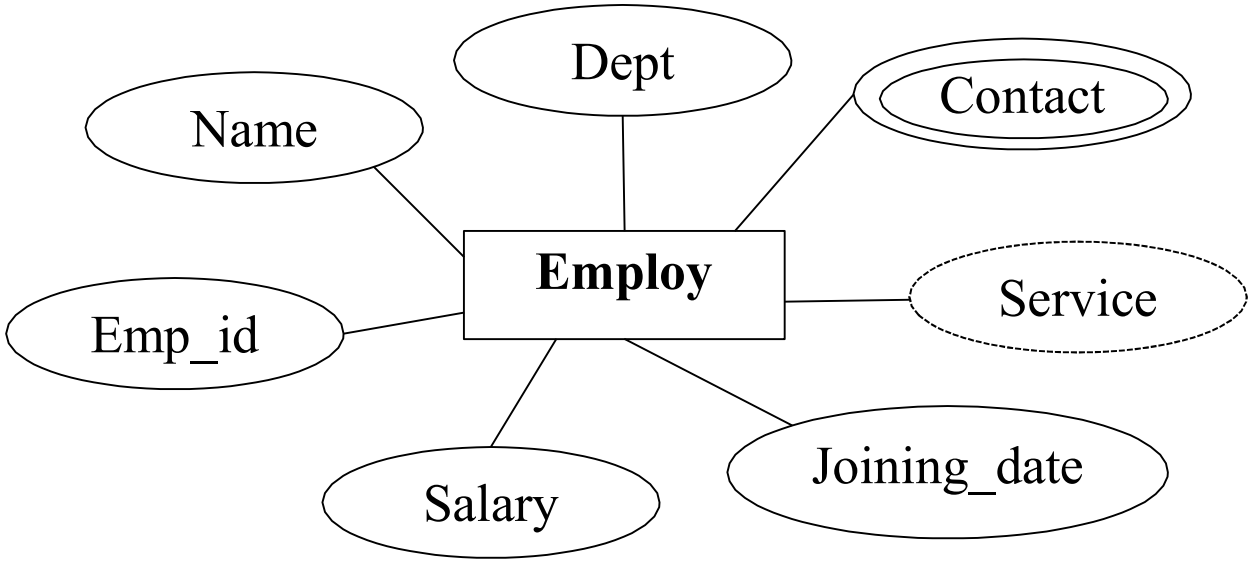


Figure – 1.5 E-R diagram for employs entity set

Constructing ER diagrams

Example:

Draw an ER diagram that represents a ***faculty*** entity set which has the attributes namely Emp_id, Name, Dept, Subject has ***teacher*** relationship set with ***students*** entity set which has the attributes namely Id_number, Name, Dept, Subject.

Constructing ER diagrams

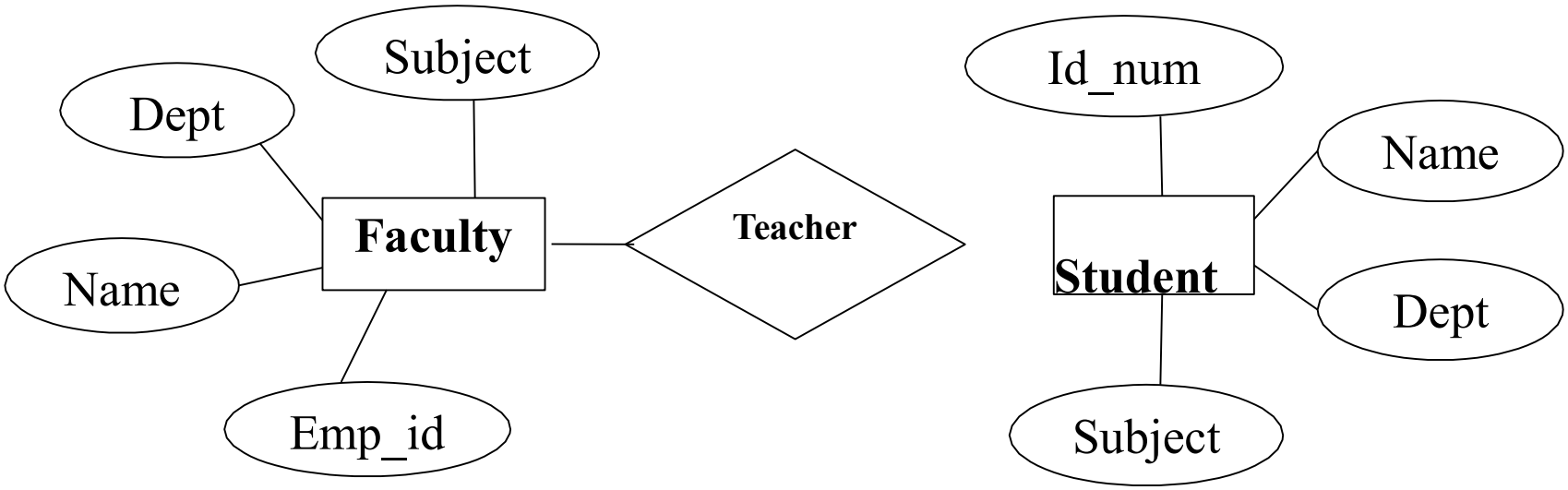


Figure -1.6 E-R diagram that represents the relationship between faculty and student.

Constructing ER diagrams

- The function that an entity plays in relationship is called entity's ***role***.
- Example: In the above example faculty plays ***teacher role*** and student plays ***listener role.***
- Of course these roles are implicit roles which are need not to be mentioned.

Constructing ER diagrams

- ***Recursive relationship sets*** – same entity set participates in a relationship set more than once in different roles.
- Example: A Staff entity set reports to another Staff entity set. Here the two Staff entities are in same entity set but one Staff entity is superior to another Staff entity.

Constructing ER diagrams

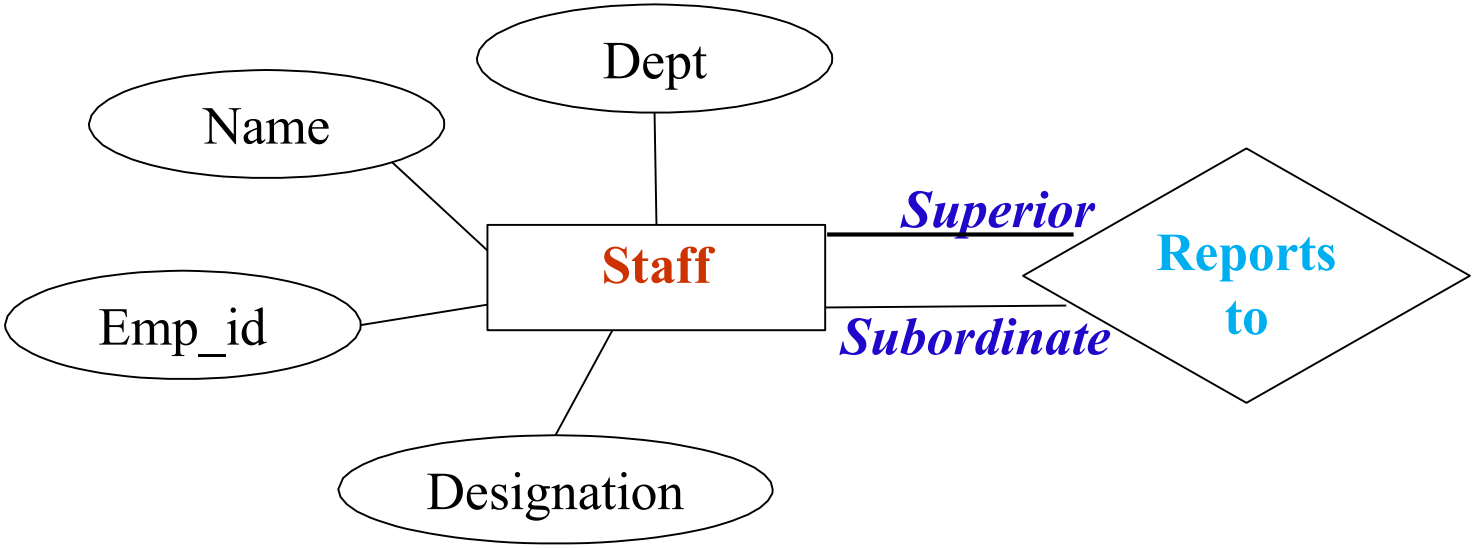


Figure – 1.7 E-R diagram for recursive relationship sets

Constructing ER diagrams

- A relationship may also have attributes called ***descriptive attributes.***

Constructing ER diagrams

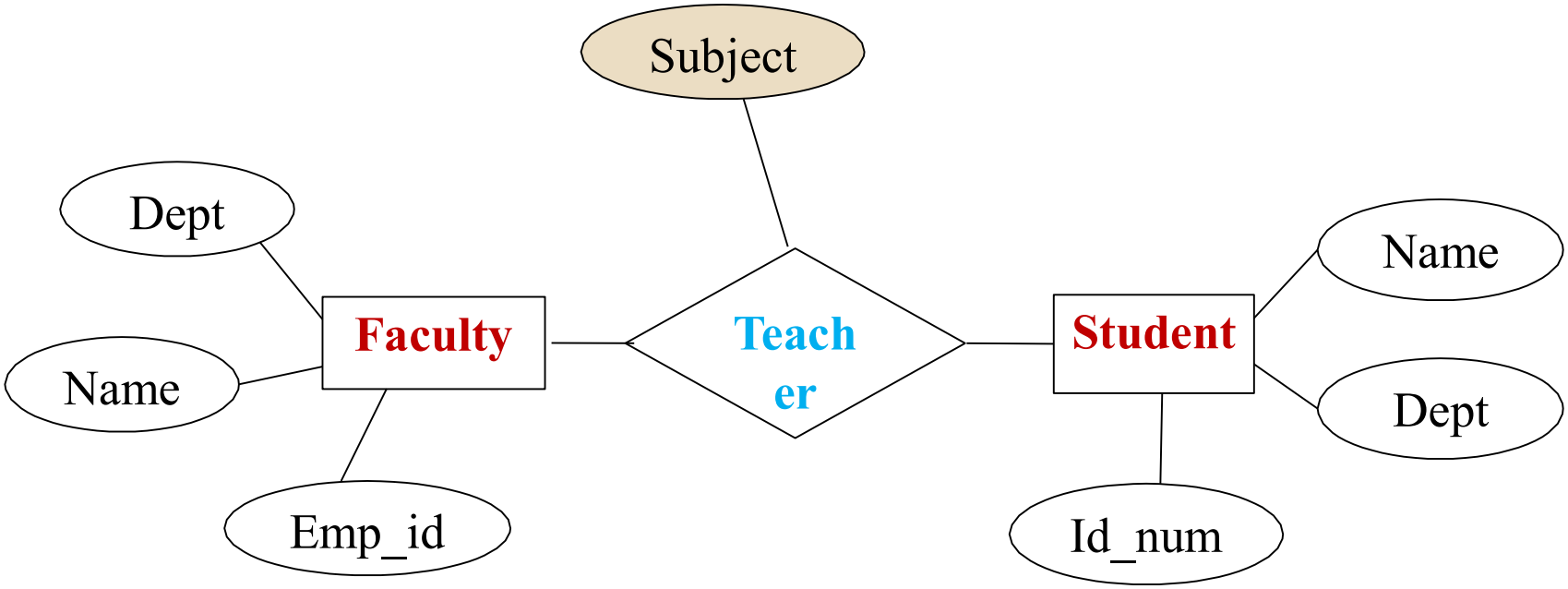


Figure -1.8 E-R diagram that represents attribute to the relationship set.

Constructing ER diagrams

- ***Degree of the relationship sets*** – No. Of entity sets participating in a relationship set.
- ***Binary relationship sets*** – two entity sets are participating in the relationship sets.
- ***Ternary relationship sets*** – three entity sets are participating in the

Constructing ER diagrams

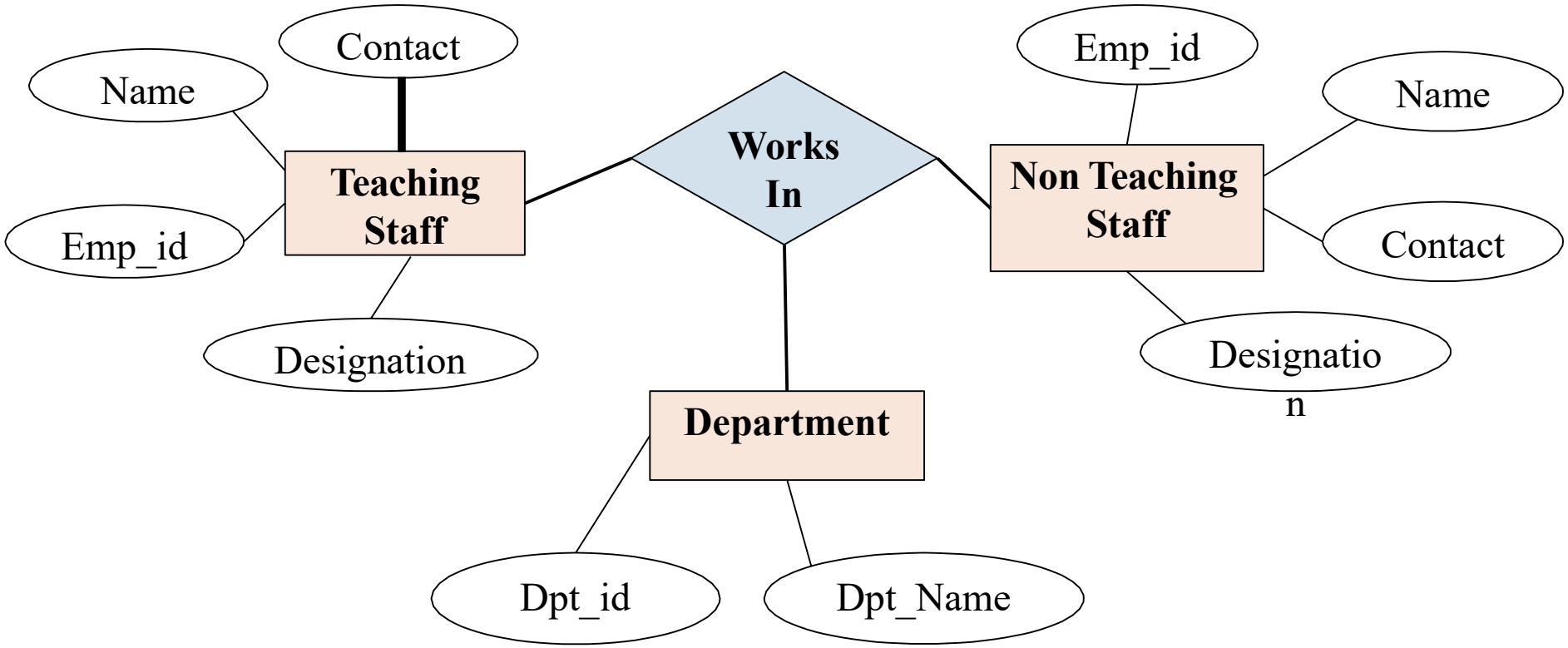


Figure -1.9 E-R diagram that represents ternary relationship set.

Modelling of Constraints

- Mapping cardinalities.
- Participation constraints.
- Keys.

Modelling of Constraints

Mapping cardinalities:

- Also known as ***Cardinality ratios***.
- Number of entities to which another entity can be associated via a relationship set.
- Useful to describe binary relationship sets.

Modelling of Constraints

Mapping cardinalities:

- Types of cardinalities – **Four**
  - **One – to – one.**
  - **One – to – many.**
  - **Many – to – one.**
  - **Many – to – many.**

Modelling of Constraints

- ‘A’ & ‘B’ are two entity sets.

- ‘R’ is a binary relationship set between A & B.

Mapping cardinalities:

- **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.

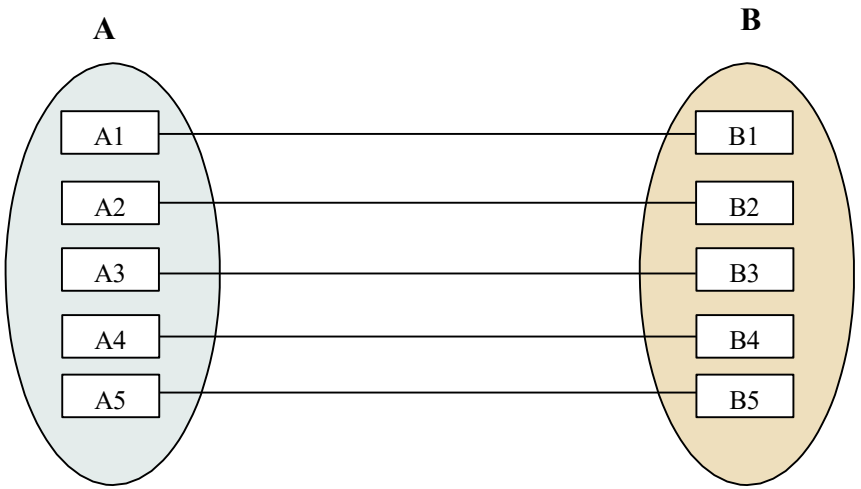
Modelling of Constraints

- ‘A’ & ‘B’ are two entity sets.

- ‘R’ is a binary relationship set between A & B.

Mapping cardinalities:

- **One – to – one**



Modelling of Constraints

Mapping cardinalities:

- **One – to – one**

  Example:

  An *invigilator,* can invigilate maximum in *one room.*

Modelling of Constraints

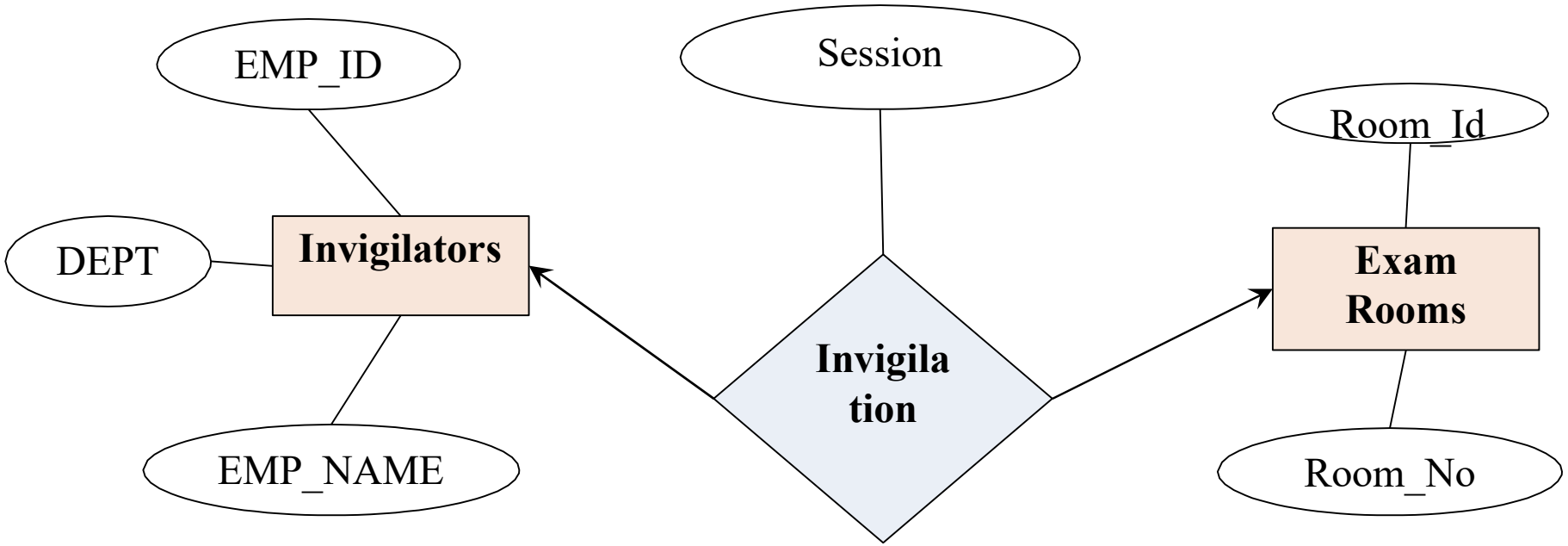


Figure -1.10 E-R diagram that represents one – to -one.

Modelling of Constraints

- ‘A’ & ‘B’ are two entity sets.

- ‘R’ is a binary relationship set between A & B.

Mapping cardinalities:

- **One – to – many**

An entity in A is associated with ***any number*** of entities in B, and an entity in B is associated with ***at most*** one entity in A.

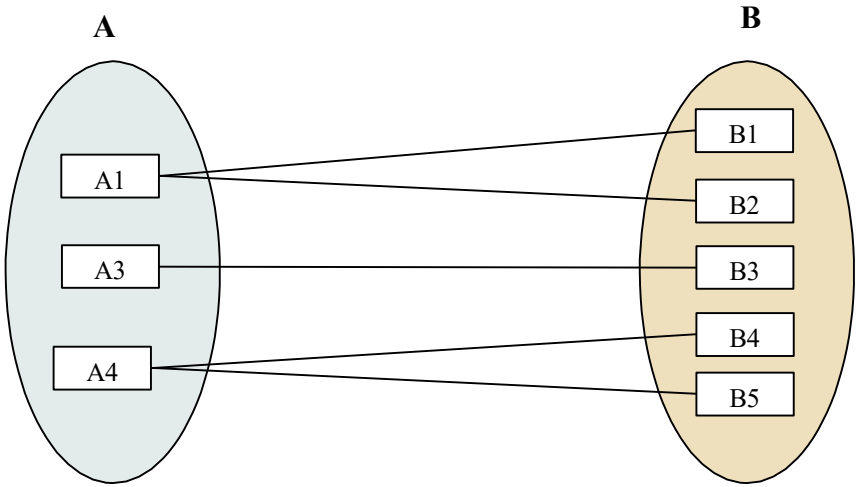
Modelling of Constraints

- ‘A’ & ‘B’ are two entity sets.

- ‘R’ is a binary relationship set between A & B.

Mapping cardinalities:

- **One – to – many**



Modelling of Constraints

Mapping cardinalities:

- **One – to – many**

  Example:

  A *teacher,* can teach more than one subjects.

Modelling of Constraints

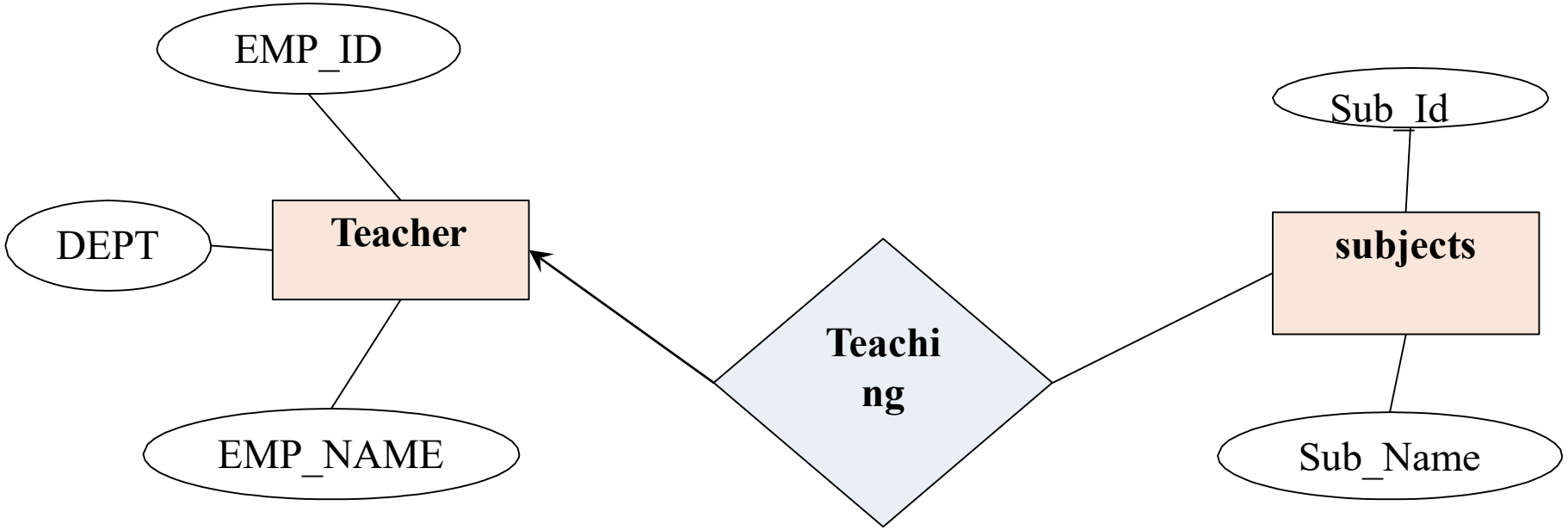


Figure -1.11 E-R diagram that represents one – to -many.

Modelling of Constraints

- ‘A’ & ‘B’ are two entity sets.

- ‘R’ is a binary relationship set between A & B.

Mapping cardinalities:

- **Many – to – one**

An entity in A is associated with ***at most*** one entity in B, and an entity in B is associated with ***any number of*** entities in A.

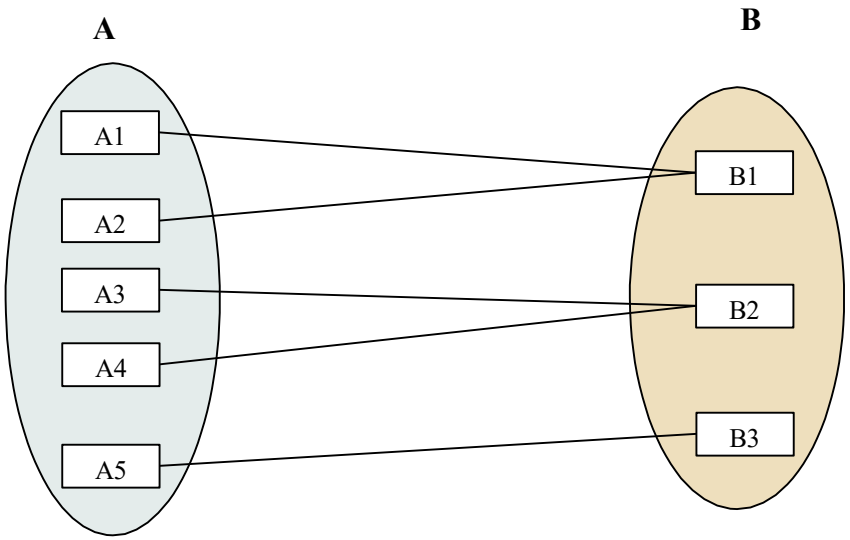
Modelling of Constraints

- ‘A’ & ‘B’ are two entity sets.

- ‘R’ is a binary relationship set between A & B.

Mapping cardinalities:

- **Many – to – one**



Modelling of Constraints

Mapping cardinalities:

- **Many – to – one**

  Example:

  Many *teaching faculty* can work in a *department.*

Modelling of Constraints

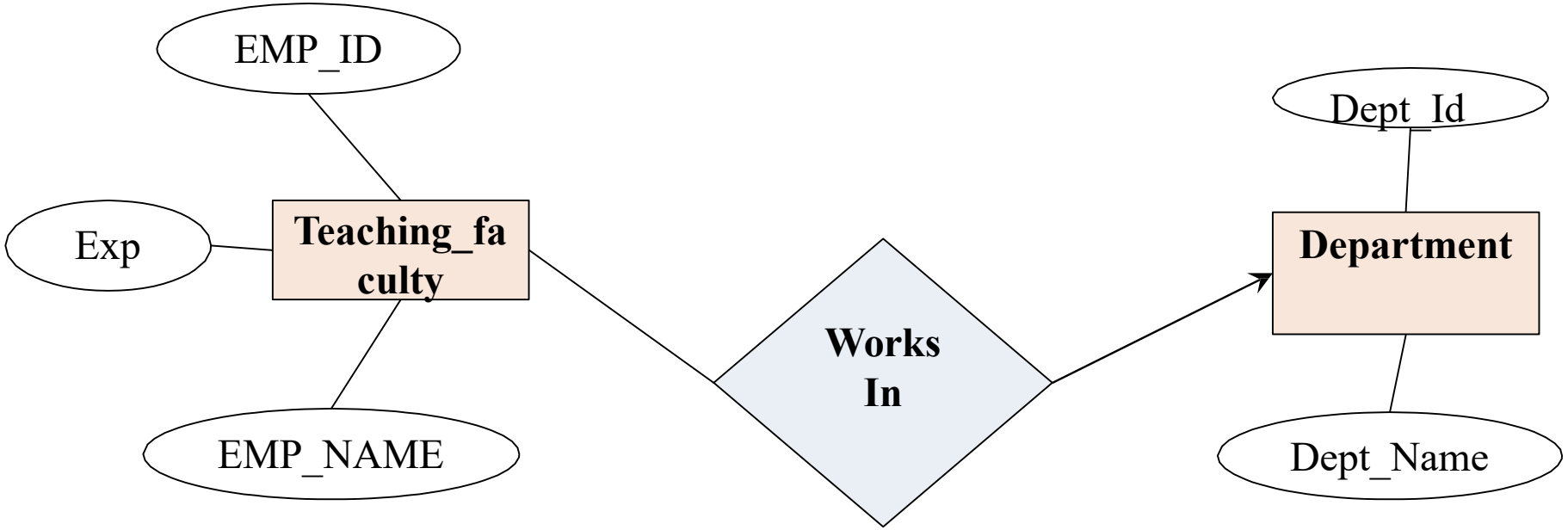


Figure -1.12 E-R diagram that represents many – to - one.

Modelling of Constraints

- ‘A’ & ‘B’ are two entity sets.

- ‘R’ is a binary relationship set between A & B.

Mapping cardinalities:

- **Many – to – many**

An entity in A is associated with ***any number*** of entities in B, and an entity in B is associated with ***any number of*** entities in A.

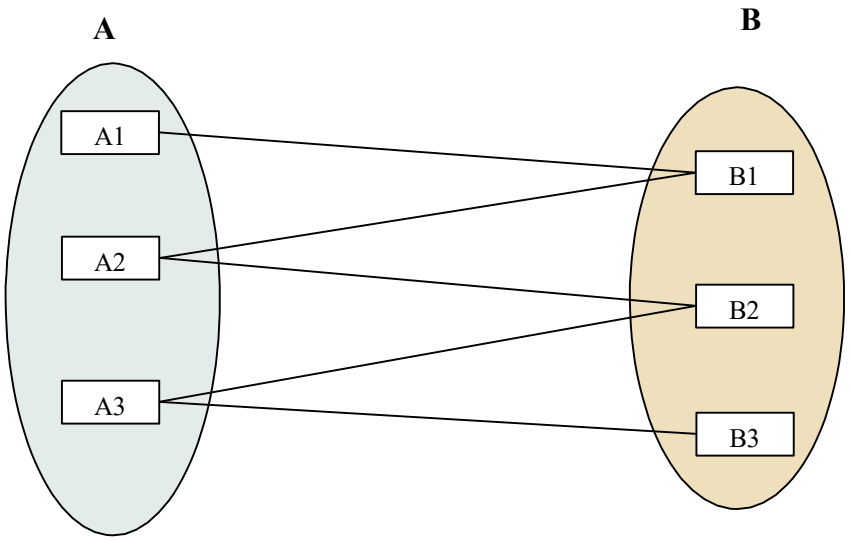
Modelling of Constraints

- ‘A’ & ‘B’ are two entity sets.

- ‘R’ is a binary relationship set between A & B.

Mapping cardinalities:

- **Many – to – many**



Modelling of Constraints

Mapping cardinalities:

- **Many – to – many**

  Example:

  Many *teaching faculty* can teach many subjects.

Modelling of Constraints

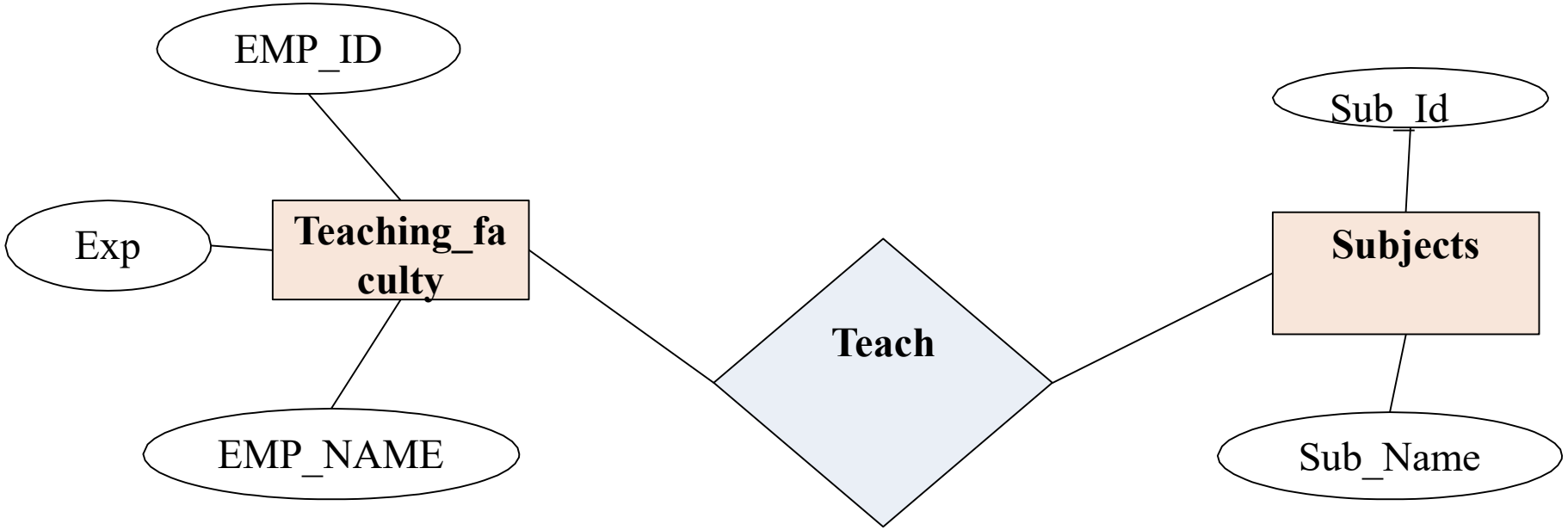


Figure -1.13 E-R diagram that represents many – to - many.

Modelling of Constraints

Participation constraints:

- It says about *number of entities* of entity set are participating in the relationship set.

Modelling of Constraints

Participation constraints:

- If *every* entity of the entity set is participating in the relationship set at least once then the entity set participation is ***total participation***.

Modelling of Constraints

Participation constraints:

- If *only some* entities of the entity set are participating in a relationship set then the entity set participation is ***partial participation***.

Modelling of Constraints

Participation constraints:

- Example:

  Every *teaching faculty* should teach at least one *subject*.

  - Teaching faculty entity set – total participation.
  - Subject entity set – partial participation.

Modelling of Constraints

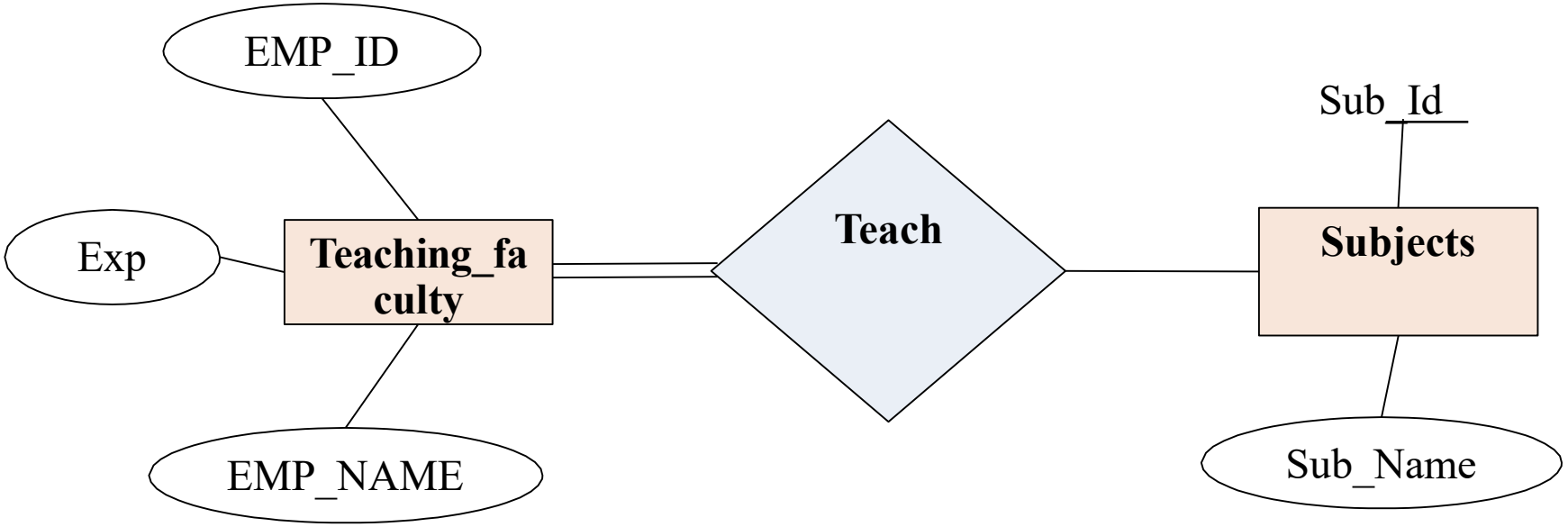


Figure -1.14 E-R diagram that represents participation constraints.

Modelling of Constraints

Keys:

- Set of attributes that is used to identify the entities uniquely in an entity set.
- Unique identification of the entities in an entity set.

Modelling of Constraints

Keys:

- Types of keys –
  - Super key.
  - Candidate key.
  - Primary key.

Modelling of Constraints

Keys:

- Super key – Set of one or more attributes that are collectively helps identify the uniquely in an entity set.
- Example:
  - Id
  - Id, name

Modelling of Constraints

Keys:

- Candidate key – Minimal super keys.
- Example:
  - Id
  - Name,dept,contact

Modelling of Constraints

Keys:

- Primary key – A key chosen among candidate keys by designers.
- Primary key attribute values are never, or very rarely, changed.
- Example:
  - Id.

Modelling of Constraints

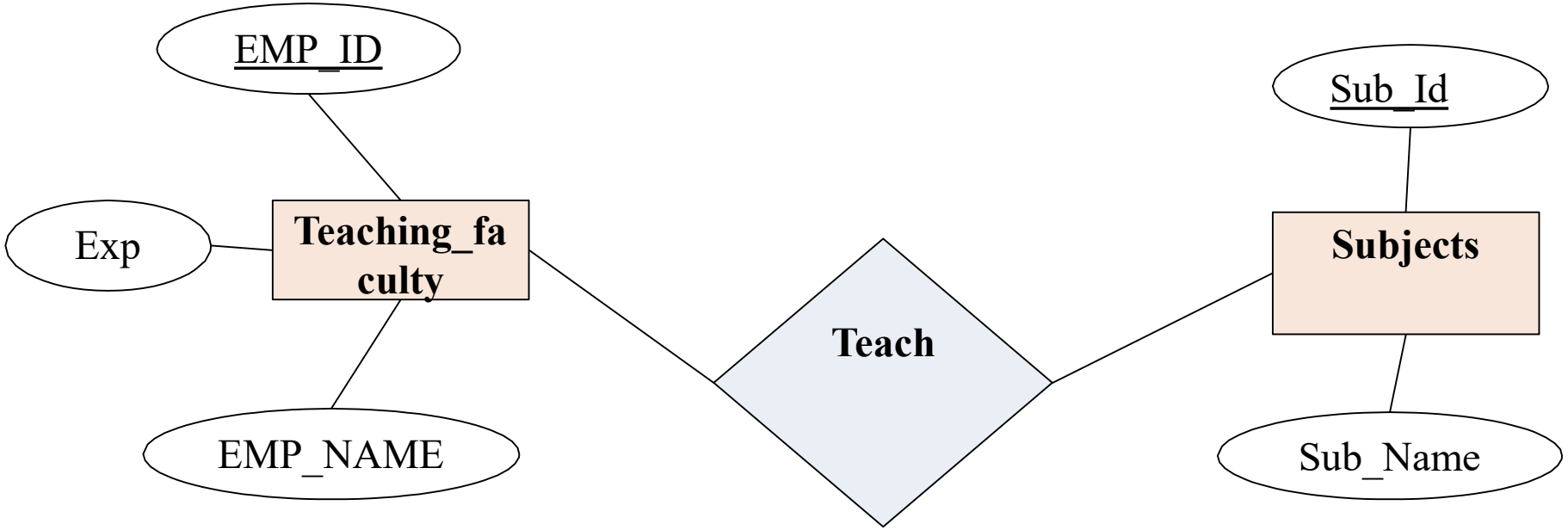


Figure -1.15 E-R diagram that represents key.

Modelling of Constraints

Keys (Relationship sets):

- Primary key –

  $Prim(E_1)$ U $Prim(E_2)$ …… $Prim(E_n)$

  (or)

  $Prim(E_1)$ U $Prim(E_2)$ …… $Prim(E_n)$ U {attributes of relationship set}

- Super key –

  $Prim(E_1)$ U $Prim(E_2)$ …… $Prim(E_n)$.

Modelling of Constraints

- If an entity set have no key / does not have sufficient keys to form primary then it is called ***weak entity set***.
- If an entity set have primary key then it can be called as ***strong entity set***.

Modelling of Constraints

- A weak entity set must be associated with a strong entity set which is called as ***owner / identifying entity set***.
- The associating relationship between owner and weak entity sets is called **identifying relationship**.

Modelling of Constraints

- A weak entity set has ***existence dependency*** on owner / identifying entity set.
- i.e Weak entity set has total participation in the relationship set.
- Identifying relationship forms **many – to – one** relationship from weak entity set to the owner entity set.

Modelling of Constraints

- The primary key of weak entity set is formed by combining partial key of weak entity set and primary key of owner entity set.
- A weak entity set is represented using ***double rectangle***.
- Identifying relationship set is represented using ***double rhombus***.

Modelling of Constraints

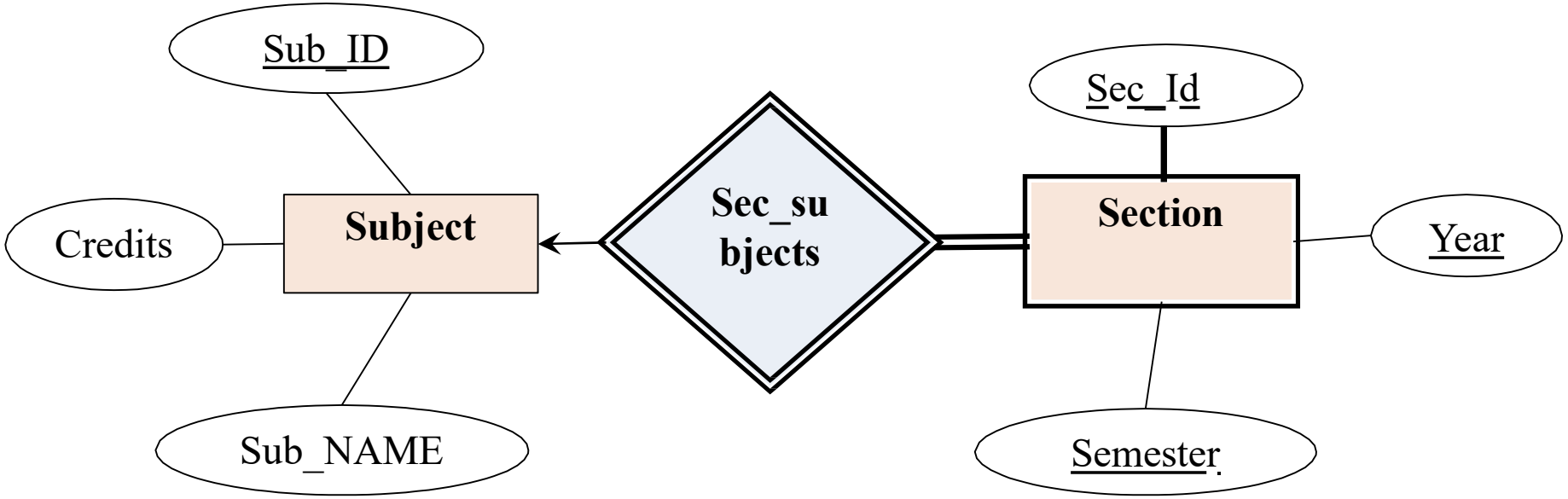


Figure -1.16 E-R diagram that represents weak entity sets.

Reduction of ER diagrams to Relational tables

- Relational table –
  - Table name.
  - Attributes name.
  - Primary key representations.

  Table_name (List_of_colums)

  - Primary key attributes are underlined.

✓ Each row/ tuple in the table represents an entity in entity set.

Reduction of ER diagrams to Relational tables

- For each entity set and for each relationship set in the database design, there is a unique relation schema to which we assign the name of the corresponding entity set or relationship set.

Reduction of ER diagrams to Relational tables

Strong entity sets with simple attributes:

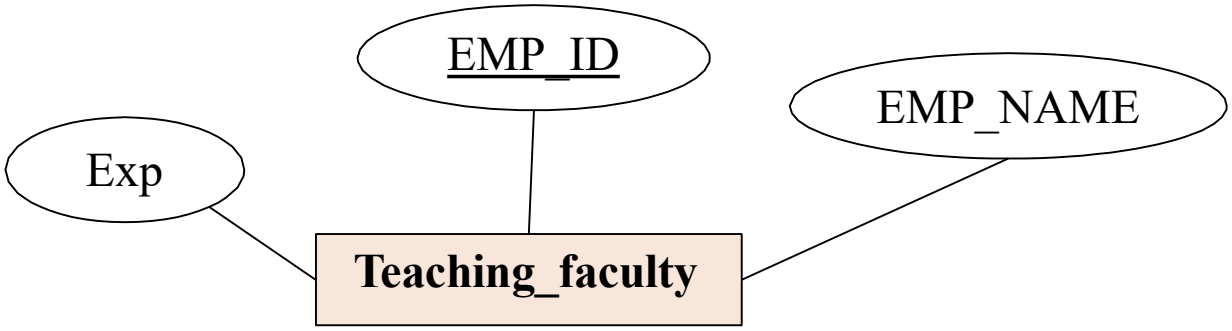
Let ***E*** be a strong entity set with only simple descriptive attributes $a_1, a_2, \ldots, a_n$.

- Entity set name – Table name.
- Attribute name – Column name.
- Primary key of entity set is primary key for table.

Reduction of ER diagrams to Relational tables

Strong entity sets with simple attributes:

Example:



Teaching_faculty (EMP_ID, EMP_NAME, Exp)

Reduction of ER diagrams to Relational tables

Strong entity sets with composite attributes:

Let ***E*** be a strong entity set with composite descriptive attributes

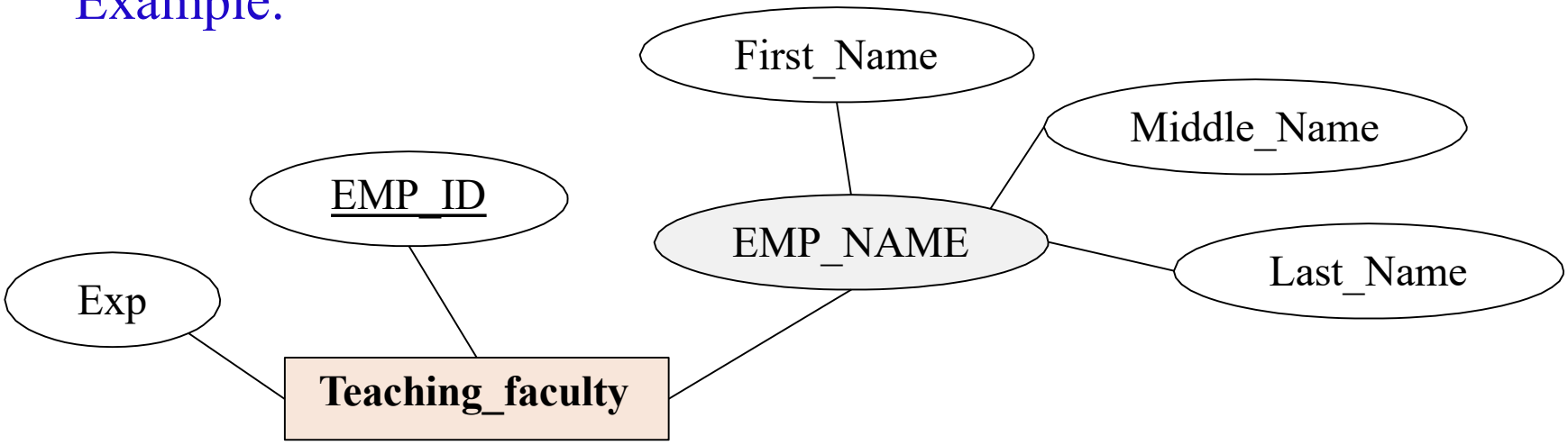
$a_1, a_2 (a_{21}, a_{22}, a_{23} \ldots a_{2n}) \ldots , a_n$.

- Entity set name – Table name.
- Simple attribute Names – Column names.
- Sub attributes of composite attribute name – Column names.
- Primary key of entity set is primary key for table.

Reduction of ER diagrams to Relational tables

Strong entity sets with composite attributes:

Example:



Teaching_faculty (EMP_ID, First_Name, Middle_Name, Last_Name, Exp)

Reduction of ER diagrams to Relational tables

Weak entity sets:

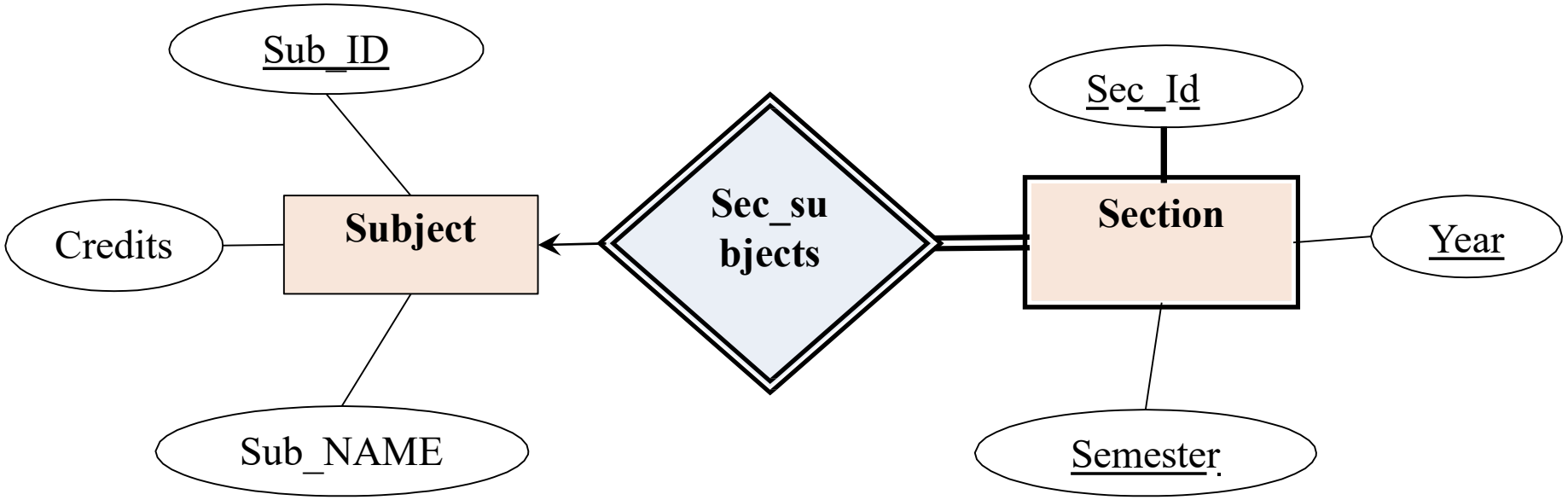
- Let, A is Weak entity set and $a_1, a_2, \ldots, a_n$ are attributes of A.
- B is a Owner entity set and $b_1, b_2, \ldots, b_n$ are prime attributes of B.

Then

- Table name – Weak entity set name.
- Columns – $\{a_1, a_2, \ldots, a_n\}$ U $\{b_1, b_2, \ldots, b_n\}$
- Primary key contains of B's primary key and partial key of A.
- Foreign key is $\{b_1, b_2, \ldots, b_n\}$

Reduction of ER diagrams to Relational tables

Weak entity sets:



Section(sub_ID, Sec_Id, Year, Semester)

Reduction of ER diagrams to Relational tables

Relationship sets:

- Let, A is Relationship set and $a_1, a_2, \ldots, a_n$ are attributes of A.
- $b_1, b_2, \ldots, b_n$ are prime attributes of entity sets which are participating in relationship set A.

Then,

- Table name – Relationship set name.
- Columns – $\{a_1, a_2, \ldots, a_n\} \cup \{b_1, b_2, \ldots, b_n\}$

Reduction of ER diagrams to Relational tables

Relationship sets:

- Primary key –

  Binary many – to – many: Union of primary key attributes of all participating entity sets.

  Binary many – to – one (or) one – to – many: Primary key of entity set on the many side relationship set.

  Binary one – to – one: primary key of any one entity set.

Reduction of ER diagrams to Relational tables

Relationship sets:

- Primary key –

  N-ary relationship set without arrow edges: Union of primary key attributes of entity sets that are participating in relationship sets.

  N-ary relationship set with an arrow on one of its edges: primary keys of the entity sets not on the “arrow” side of the relationship sets.

Reduction of ER diagrams to Relational tables

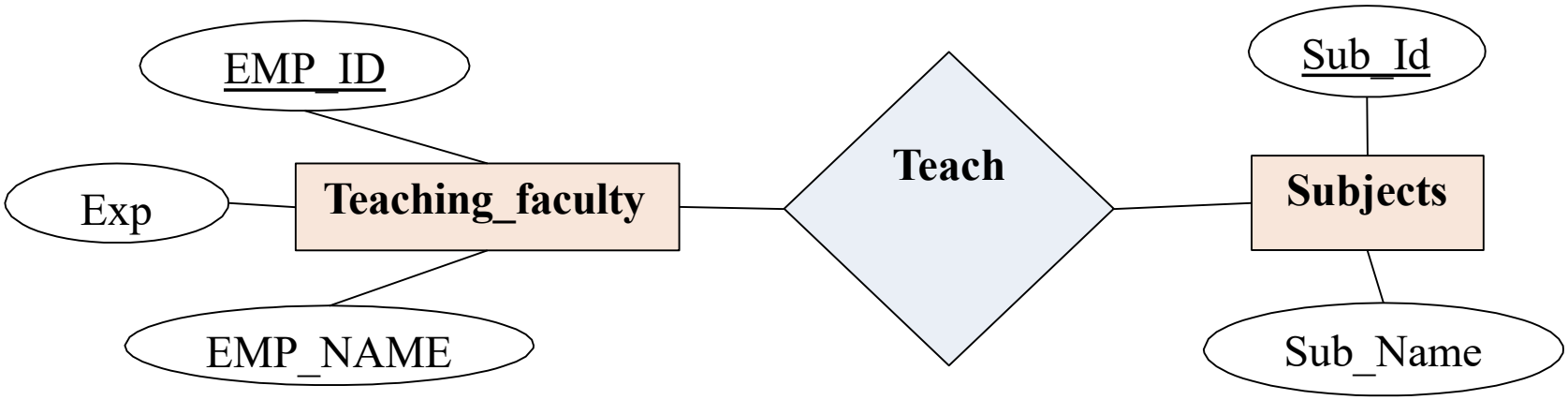
Relationship sets:

- Foreign key constraints –

  For each entity set E_i related to relationship set A, we create a foreign-key constraint from table, with the attributes of A that were derived from primary-key attributes of E_i referencing the primary key of the table representing E_i.

Reduction of ER diagrams to Relational tables

Relationship sets: Example



Teaching_faculty(EMP_ID,EMP_NAME,Exp)

Subjects(Sub_Id, Sub_Name)

Teach(EMP_ID, Sub_Id)

Thank You