



# Database Management Systems

## UNIT I

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.1

---

---

---

---

---

---

---

---



## Learning Objectives

### Basic Concepts:

1. Database and database users
2. Characteristics of the database
3. Database Systems
4. Concepts and Architecture
5. Data Models
6. Schemas and Instances
7. DBMS Architecture and Data Independence
8. Overview of Hierarchical
9. Network & Relational Data Base Management Systems

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.2

---

---

---

---

---

---

---

---



## Learning Objectives

### Data Modelling using Entity Relationship Model::

1. ER Model Concepts
2. Notation for ER diagram
3. Mapping constraints
4. Concepts of Keys
5. Extended ER Model
6. Generalization
7. Specialization
8. Aggregation
9. ER diagram to tables Mapping

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.3

---

---

---

---

---

---

---

---

## Basic Concepts

- A database is a **record keeping** concept/system.
- It contains information in the table for a particular enterprise.
- DBMS is **database management system**. Databases are the collection of data in order to store and retrieve data. The database consists of data which can be a numeric, alphabetic and also alphanumeric form. Analyzing data is a key feature of database management system that is DBMS.
- Purpose of the database:
  - To have the information handy. One can **retrieve** the data from the database whenever required.
  - It is possible to search the database whenever required.
  - Above all there is database management system to provide **centralized control** of all the data.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.4

---

---

---

---

---

---

---

---

---

---

## Database Abstraction

- Data abstraction is the process of **hiding unwanted and irrelevant details** from the end user.
- It helps to store information in such a way that the end user can access data which is necessary, the user will not be able to see what data is stored or how it is stored in a database.
- Data abstraction helps to keep **data secure from unauthorized access** and it hides all the implementation details.

```

graph TD
    subgraph View_Level [View Level]
        V1[View 1]
        V2[View 2]
        V3[View 3]
    end
    LL[Logical Level]
    PL[Physical Level]
    LL --> V1
    LL --> V2
    LL --> V3
    PL --> LL
  
```

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.5

---

---

---

---

---

---

---

---

---

---

## Key terms of database

- Database Schema:** **Design/Skelton** of the DB used to represent the structure, type of data, how they will be stored in the rows and columns, relationship between them etc.
- Data Constraint:** The constraint put on the table (column/rows) about what particular type of data should be stored in a particular column. e.g. DOB (this must not be text, it should be in DD-MM-YYYY format)
- Data Dictionary/Metadata:** **Data about the data** is metadata. Database schema along with different constraint is being stored in dictionary, which is a metadata only.
- Database Instance:** Used to define the complete database environment and its components.
- Query:** Query is used to **access data from the database**.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.6

---

---

---

---

---


---

---

---

---

---



## Key terms of database

- **Data Manipulation:** One can easily manipulate the data in database using three operations:
  - Insertion
  - Deletion
  - Updation
- **Data Engine:** It is an underlying component used to create and manage various database queries.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.7

---

---

---

---

---


---

---

---

---

---



## Types of database users

- Database users are categorized based upon their interaction to the database:
  - **Database Administrator (DBA):** The person/team **who defines the schema** and also control 3 levels of DB. They also provides security to DB.
  - **Naïve/parametric end users:** Users **who don't have any DBMS knowledge** but they frequently used DB applications. (clerks in bank)
  - **A system analyst:** A user who **analyse the requirement** of naïve/parametric users. They check whether their requirements are satisfied.
  - **Sophisticated Users:** Can be an engineer, scientists who are familiar with the DB and concepts. They can develop their own DB according to the requirements. They don't write the program code but **interact the database by writing SQL queries.**
  - **Database Engineers/designer:** users who **design the DB** which includes tables, indexes, views. How the data is stored and how they are related to each other.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.8

---

---

---

---

---


---

---

---

---

---



## Types of database users (Contd.)

- Database users are categorized based upon their interaction to the database:
  - **Application Programmer:** **Back-end programmer** who writes the code for te application.
  - **Casual/temporary users:** Who **occasionally access the DB** but each time they did, they want new information.
  - **Specialized Users:** Who **writes specialized application.**

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.9

---

---

---

---

---

---

---

---

---

---

## Database Architecture

- Database architecture refers to the **structured design** and methodology of a database system which forms the core of DBMS.
- One of the type of database is chosen, an architecture is planned accordingly.
- Majorly the architecture is of **three (3) Types**:
  - One-tier architecture
  - Two- tier architecture
  - Three-tier architecture

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.10

---

---

---

---

---

---

---

---

## One-tier Architecture

- The DB, user interface and application logic resides on one machine/server.
- Used for small scale applications.
- Most of the developers choose such systems as it's easy and simple. But, it doesn't provide many tools for the end-users.
- Thus, it is not suitable for data delivery platforms like websites globally but, may be useful for storing large data.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.11

---

---

---

---

---

---

---

---

## Two-tier Architecture

- Multiple Clients connecting directly to the DB.
- In such systems, the communication is between the client and server.
- The application tier is there to provide communication between the two and is independent in terms of the code level logic happening at server or client.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.12

---

---

---

---

---

---

---

---

## Three-tier Architecture

- Most modern web application uses three-tier architecture.
- Clients connect to backend and then in turn connect to DB.

**Three Tier Architecture**

```

graph TD
    subgraph Presentation_Tier [Presentation Tier]
        UV[User View]
    end
    subgraph Application_Tier [Application Tier]
        LV[Logical View]
    end
    subgraph Database_Tier [Database Tier]
        DB[(Database)]
    end
    UV <--> LV
    LV <--> DB
    
```

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.13

---

---

---

---

---

---

---

---

---

---

## Data Model Schema and Instances

- The data is being stored in a database at a particular moment of time is **Instance of the DB**.
- The overall design of a database is called **Schema**.
- The schema represents the **logical view** of the entire DB.
- A schema contains objects like tables, columns, foreign key, primary key, view etc.
- A schema can be represented with the help of a diagram. The diagram shows **database objects** and **relationships** with each other.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.14

---

---

---

---

---

---

---

---

---

---

## Data Model Schema and Instances (Contd.)

Student			
Name	Roll No.	Ph. No.	Address

Course			
Name	ID	Credit	Hours

Section			
Name	Semester	Year	Teacher

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.15

---

---

---

---

---

---

---

---

---

---

## Three-schema architecture

- It is used to describe three structure of a specific database system.
- The goal of three-schema architecture is to separate the user application and the physical database.

The diagram illustrates the three-schema architecture. At the top, 'End Users' are shown interacting with 'External Views' at the 'External Level'. Below this is the 'Conceptual Level' with 'Conceptual Schemas'. At the bottom is the 'Internal Level' with 'Internal Schemas' and a 'Stored Database'. Arrows indicate 'External/Conceptual Mapping' between the External and Conceptual levels, and 'Conceptual/Internal Mapping' between the Conceptual and Internal levels.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.16

---

---

---

---

---

---

---

---

---

---

## Three-level schema

- The three level schema is as follows:
  - Internal/Physical schema:**  
It describes the **physical storage** structure of DB. It describes low level data in detail.
  - Conceptual Schema:**  
It describes the structure of the whole database for a community of users.  
It **hides the structure of physical storage**, it describes the relationships, data types and constraints.
  - External/view level:**  
It includes **user view**.  
It describes the part of the database that a particular user group is interested in and hide the rest of the DB from that user group.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.17

---

---

---

---

---

---

---

---

---

---

## Data Independence

It is the capacity to change the schema at one level without changing the schema of another level.

Broadly this is of 2 types:

- Physical Data Independence**
  - Ability to change the structure of lowest level of DBMS without affecting the highest level of schema.  
e.g. If user wants to upgrade the storage system from hard-disk to SSD without disturbing the logical data/schema.
- Logical Data Independence**
  - It changes the conceptual schema without changing the external schema/application programs.  
e.g. Add/modify/delete without rewriting the application program.  
Or merging two records into one.  
**It is difficult to achieve logical data independence**

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.18

---

---

---

---

---

---

---

---

---

---

**Hierarchical Database Management System**

- It organizes the data in **tree like structure**, the topmost node is called the root node.
- Each node has exactly one parent and a parent can have **multiple child nodes**.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.19

---

---

---

---

---

---

---

---

**Network Database Management System**

- It is **advanced hierarchical model**.
- This **uses graphs** instead of trees.
- A child have more than one parent node.
- It uses the concepts of records and sets.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.20

---

---

---

---

---

---

---

---

**Relational Database Management System**

- There are no physical links
- Data is presented in table format only.
- Relationship between tables are set.

Table also called Relation

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Primary Key: CustomerID  
Domain: CustomerName, Status  
© guru99.com

Column OR Attributes  
Total # of column is Degree

Tuple OR Row  
Total # of rows is Cardinality

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.21

---

---

---

---

---

---

---

---

**Question**

State five (05) significant difference amongst hierarchical, network and relational data models.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.22

---



---



---



---



---



---

**Data modelling using the entity relationship model**

ER Model Concepts

- The entity relational model is all about **representation** of the **entities** and **relationship amongst these entities**.
- It is **structural format** of the Db.
- ER diagram makes easy to **convert the relations**.
- It is used to form the **logical view** of the system.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.23

---



---



---



---



---



---

**Symbols used in E-R model**

	Entity		Attribute
	Weak Entity		Key Attribute
	Associative Entity		Key Attribute
	Relationship		Key Attribute
	Identifying Relationship		Derived Attribute
	Mandatory Relationship		Optional Relationship
	Partial Participation		Total Participation

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.24

---



---



---



---



---



---

Entity

- In a database management system (DBMS), an entity is a **unique object** that is represented in a database using attributes. Entities can be tangible objects, like a car, or non-real-world objects, like a user account. They are **independent of other things in the database** and represent real-world concepts or objects with an independent existence.

Entity

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.25

---

---

---

---

---

---

---

---

Weak Entity

- In a relational database, a weak entity is an entity that **cannot be uniquely identified** by its attributes alone.
- It must use a **foreign key** in conjunction with its attributes to create a primary key.
- The foreign key is typically a primary key of an entity it is related to.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.26

---

---

---

---

---

---

---

---

Attributes

- In a database management system (DBMS), attributes describe the **characteristics of a component or entity**, such as a database table or field. Attributes are used to distinguish one entity from another, and are important for making databases searchable. For example, a customer's name or an order date are attributes that describe the entities of customers and orders

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.27

---

---

---

---

---

---

---

---

**Attributes (Contd.)**

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.28

---

---

---

---

---

---

---

---

**Attributes (Contd.)**

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.29

---

---

---

---

---

---

---

---

**Symbols used in E-R model**

**one-to-one (1:1)**  
 Employee (1) — Manage — (1) Department

**one-to-many (1:N)**  
 Publisher (1) — supplies — (N) Book

**many-to-one (N:1)**  
 Book (N) — has — (1) Section

**many-to-many (M:N)**  
 Course (M) — enrolled by — (N) Student

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.30

---

---

---

---

---

---

---

---

**Relationship Type**

- A relationship types is the association between entities.
- One-to-one relationship

```

    erDiagram
        Head ||--|| Branch : Handles
    
```

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.31

---

---

---

---

---

---

---

---

**Relationship Type**

- One-to-many relationship

```

    erDiagram
        Customer ||--}| Orders : Places
    
```

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.32

---

---

---

---

---

---

---

---

**Relationship Type**

- Many-to-one relationship

```

    erDiagram
        Students }|--|| DBMS Course : Enorolled
    
```

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.33

---

---

---

---

---

---

---

---

**Relationship Type**

- Many-to-many relationship

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.34

---

---

---

---

---

---

---

---

**Degree of a relationship set**

- The number of entity sets that participate in a relationship set is termed as the degree of that relationship set.

Degree of a relationship set = Number of entity sets participating in a relationship set

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.35

---

---

---

---

---

---

---

---

**Types of relationship sets**

- On the basis of degree of a relationship set, a relationship set can be classified into the following types-
- Unary relationship set
- Binary relationship set
- Ternary relationship set

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.36

---

---

---

---

---

---

---

---

**Unary relationship set**

- Unary relationship set is a relationship set where only one entity set participates in a relationship set.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.37

---

---

---

---

---

---

---

---

**Binary relationship set**

- Binary relationship set is a relationship set where two entity sets participate in a relationship set.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.38

---

---

---

---

---

---

---

---

**Ternary Relationship Set**

- Ternary relationship set is a relationship set where three entity sets participate in a relationship set.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.39

---

---

---

---

---

---

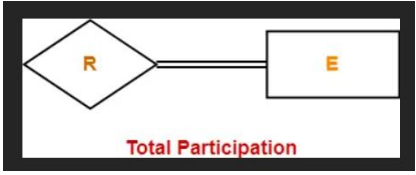
---

---

**Participation Constraints**

- Total Participation of an Entity set

Total participation in an entity set in an ER diagram means that every entity in the set must participate in at least one relationship instance in that relationship set.



© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.40

---

---

---

---

---

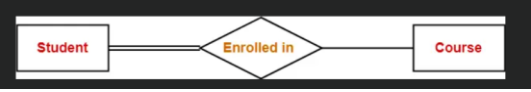
---

---

---

**Participation Constraints**

- Total Participation of an Entity set



© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.41

---

---

---

---

---

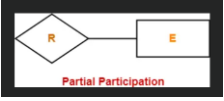
---

---

---

**Participation Constraints**

- Partial Participation of an Entity set
- It specifies that each entity in the entity set may or may not participate in the relationship instance in that relationship set.
- That is why, it is also called as optional participation.
- Partial participation is represented using a single line between the entity set and relationship set.



© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.42

---

---

---

---

---

---

---

---

**Participation Constraints**

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.43

---

---

---

---

---

---

---

---

**Relationship between Cardinality and Participation Constraints**

- Minimum cardinality tells whether the participation is partial or total.
- If minimum cardinality = 0, then it signifies partial participation.
- If minimum cardinality = 1, then it signifies total participation.
- Maximum cardinality tells the maximum number of entities that participates in a relationship set.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.44

---

---

---

---

---

---

---

---

**Extended ER Diagram**

- Extended ER is an enhanced data model. It represents the enhanced and complex databases.
- Sub classes and super classes:
  - Superclass is an entity that can be divided into various sub-classes.
  - Sub-classes inherits various properties of the superclass

Fig. Super class/Sub class Relationship

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.45

---

---

---

---

---

---

---

---

**Specialization**

- Specialization is a process of identifying the subsets of an entity.
- It is Top-down approach.

The diagram shows an entity 'EMPLOYEE' with attributes 'E.SALARY' and 'E.NAME'. It is specialized into two entities: 'TESTER' (with attribute 'TES.TYPE') and 'DEVELOPER'.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.46

---

---

---

---

---

---

---

---

**Generalization**

- Generalization is generalizing the entities having generalized attributes.
- It is Bottom-up approach.

The diagram shows two entities: 'FACULTY' (with attribute 'S.FEE') and 'STUDENT'. They are generalized into a 'PERSON' entity with attributes 'P.ADD' and 'P.NAME'.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.47

---

---

---

---

---

---

---

---

**Aggregation**

- Sometimes the relationships are aggregated into a higher-level entity.
- Aggregation is an abstraction through which we can represent relationships as higher level entity sets.

The diagram shows a relationship 'WORKS FOR' between 'PROJECT' (m) and 'EMPLOYEE' (n). This relationship is aggregated into a higher-level entity 'MACHINERY' through a 'REQUIRE' relationship (m to n).

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.48

---

---

---

---

---

---

---

---

ER diagram to tables mapping

- **Entity type becomes a table.**
  - In the given ER diagram, LECTURE, STUDENT, SUBJECT and COURSE forms individual tables.
- **All single-valued attribute becomes a column for the table.**
  - In the STUDENT entity, STUDENT\_NAME and STUDENT\_ID form the column of STUDENT table. Similarly, COURSE\_NAME and COURSE\_ID form the column of COURSE table and so on.
- **A key attribute of the entity type represented by the primary key.**
  - In the given ER diagram, COURSE\_ID, STUDENT\_ID, SUBJECT\_ID, and LECTURE\_ID are the key attribute of the entity.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.49

---

---

---

---

---

---

---

---

---

---

ER diagram to tables mapping

- **The multivalued attribute is represented by a separate table.**
  - In the student table, a hobby is a multivalued attribute. So it is not possible to represent multiple values in a single column of STUDENT table. Hence we create a table STUD\_HOBBY with column name STUDENT\_ID and HOBBY. Using both the column, we create a composite key.
- **Composite attribute represented by components.**
  - In the given ER diagram, student address is a composite attribute. It contains CITY, PIN, DOOR#, STREET, and STATE. In the STUDENT table, these attributes can merge as an individual column.
- **Derived attributes are not considered in the table.**
  - In the STUDENT table, Age is the derived attribute. It can be calculated at any point of time by calculating the difference between current date and Date of Birth.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.50

---

---

---

---

---

---

---

---

---

---

Practice

- Strong entity set with simple attributes
- Strong entity set with composite attributes
- Strong entities with multivalued attributes
- Translating relationship set into a table
- For binary relationship with cardinality ratios
  - m:n
  - 1:m
  - m:1
  - 1:1

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.51

---

---

---

---

---

---

---

---

---

---

**Practice ER diagram**

- 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 using the Chen notation as in your textbook. List your assumptions and clearly indicate the cardinality mappings as well as any role indicators in your ER diagram.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.52

---

---

---

---

---

---

---

---

---

---

**Solution ER diagram**

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.53

---

---

---

---

---

---

---

---

---

---

**ER diagram**

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.54

---

---

---

---

---

---

---

---

---

---

**Relational Schemas (Strong entity)**

**EMPLOYEE**  

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary
-------	-------	-------	-----	-------	---------	-----	--------

**DEPARTMENT**  

Dname	Dnumber
-------	---------

**PROJECT**  

Pname	Pnumber	Plocation
-------	---------	-----------

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.55

---

---

---

---

---

---

---

---

**Relational Schemas (Weak entity)**

**EMPLOYEE**  

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary
-------	-------	-------	-----	-------	---------	-----	--------

**DEPARTMENT**  

Dname	Dnumber
-------	---------

**PROJECT**  

Pname	Pnumber	Plocation
-------	---------	-----------

**DEPENDENT**  

Essn	Dependent_name	Sex	Bdate	Relationship
------	----------------	-----	-------	--------------

Diagram showing relationships: An arrow points from the 'Ssn' attribute of the EMPLOYEE entity to the 'Essn' attribute of the DEPENDENT entity. Another arrow points from the 'Dnumber' attribute of the DEPARTMENT entity to the 'Essn' attribute of the DEPENDENT entity. A third arrow points from the 'Pnumber' attribute of the PROJECT entity to the 'Essn' attribute of the DEPENDENT entity.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.56

---

---

---

---

---

---

---

---

**Relational Schemas (1:1 mapping)**

**EMPLOYEE**  

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary
-------	-------	-------	-----	-------	---------	-----	--------

**DEPARTMENT**  

Dname	Dnumber	Mgr_ssn	Mgr_start_date
-------	---------	---------	----------------

Diagram showing a 1:1 mapping relationship between the 'Ssn' attribute of the EMPLOYEE entity and the 'Mgr\_ssn' attribute of the DEPARTMENT entity, indicated by an arrow pointing from 'Mgr\_ssn' to 'Ssn'.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.57

---

---

---

---

---

---

---

---

**Relational Schemas (1:n mapping)**

The diagram illustrates the following tables and their attributes:

- EMPLOYEE**: Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Super\_ssn, Dno
- DEPARTMENT**: Dname, Dnumber, Mgr\_ssn, Mgr\_start\_date
- PROJECT**: Pname, Pnumber, Plocation, Dnum
- DEPENDENT**: Essn, Dependent\_name, Sex, Bdate, Relationship

Relationships shown:

- EMPLOYEE (1) to DEPARTMENT (n): Ssn to Dnumber
- EMPLOYEE (1) to DEPARTMENT (n): Super\_ssn to Mgr\_ssn
- EMPLOYEE (1) to DEPARTMENT (n): Dno to Dnumber
- EMPLOYEE (1) to PROJECT (n): Ssn to Dnum
- EMPLOYEE (1) to DEPENDENT (n): Ssn to Essn

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.58

---

---

---

---

---

---

---

---

**Relational Schemas (m : n mapping)**

The diagram illustrates the following tables and their attributes:

- EMPLOYEE**: Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Super\_ssn, Dno
- DEPARTMENT**: Dname, Dnumber, Mgr\_ssn, Mgr\_start\_date
- PROJECT**: Pname, Pnumber, Plocation, Dnum
- WORKS ON**: Essn, Pno, Hours
- DEPENDENT**: Essn, Dependent\_name, Sex, Bdate, Relationship

Relationships shown:

- EMPLOYEE (m) to DEPARTMENT (n): Ssn to Dnumber
- EMPLOYEE (m) to DEPARTMENT (n): Super\_ssn to Mgr\_ssn
- EMPLOYEE (m) to DEPARTMENT (n): Dno to Dnumber
- EMPLOYEE (m) to PROJECT (n): Ssn to Dnum
- EMPLOYEE (m) to DEPENDENT (n): Ssn to Essn
- EMPLOYEE (m) to WORKS ON (n): Ssn to Essn
- EMPLOYEE (m) to WORKS ON (n): Dno to Pno

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.59

---

---

---

---

---

---

---

---

**Relational Schemas (multivalued attributes)**

The diagram illustrates the following tables and their attributes:

- EMPLOYEE**: Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Super\_ssn, Dno
- DEPARTMENT**: Dname, Dnumber, Mgr\_ssn, Mgr\_start\_date
- DEPT\_LOCATIONS**: Dnumber, Dlocation
- PROJECT**: Pname, Pnumber, Plocation, Dnum
- WORKS\_ON**: Essn, Pno, Hours
- DEPENDENT**: Essn, Dependent\_name, Sex, Bdate, Relationship

Relationships shown:

- EMPLOYEE (1) to DEPARTMENT (n): Ssn to Dnumber
- EMPLOYEE (1) to DEPARTMENT (n): Super\_ssn to Mgr\_ssn
- EMPLOYEE (1) to DEPARTMENT (n): Dno to Dnumber
- EMPLOYEE (1) to DEPARTMENT (n): Dno to Dlocation (via DEPT\_LOCATIONS)
- EMPLOYEE (1) to PROJECT (n): Ssn to Dnum
- EMPLOYEE (1) to DEPENDENT (n): Ssn to Essn
- EMPLOYEE (1) to WORKS\_ON (n): Ssn to Essn
- EMPLOYEE (1) to WORKS\_ON (n): Dno to Pno

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63, by Dr. Rakhee Sharma, Asst. Prof BVICAM U1.60

---

---

---

---

---

---

---

---