

Database Design/ Data Modelling

By –

Rajeev Srivastava

DBMS's 3 Layered Architecture

Database System Architecture can be defined in three levels

- **External Level** – The way data will be represented to the end user. View/Top/High Level.
- **Conceptual Level & Logical Level** (DBMS Independent)– Defines Conceptual or Logical structure of the system
 - Conceptual: entities, relationships.
 - Logical: entities, relationships, relationship cardinality, attributes, constraints etc.
- **Physical Level (Internal/Low)** (DBMS Specific) – Defines the way data is stored on physical storage. Data types, DataFiles, TableSpaces, Partitioning, Indexing etc.

Database Design/Data Modelling

- Data Modelling is the process of defining –
 - What data to be stored
 - How data is associated
 - The rules for data organization

Database Design: Goals

- Accuracy-Integrity
- Consistency
- Speed/Performance
- Extensibility
- Scalability
- Error Correction
- Robustness
- Ease of Use
- Portability
- Security
- Sharing
- Ability to Perform Complex Calculations

Understand User/System's Requirements for Database Design

- Functional Requirements (+Reports Requirement)
- Data Flow
- Data Format & Validity
- Data Security Needs

Create Use Cases and Identify Data Needs for each Use Case




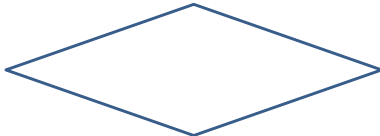
Identify **Entities** and **Relationship** among Entities -

Entity Relationship (ER) Model

- Proposed by Peter Chen in 1970s.
- An object model: graphically represents **Entities** and **Relationship** among entities.
- Describes data aspects of a system under implementation on **abstract level**.
- Represents Conceptual/Logical Design of a database.
- Major Notations
 - Chen Notation
 - Crow-Foot Notation
 - IDEF1X Notation

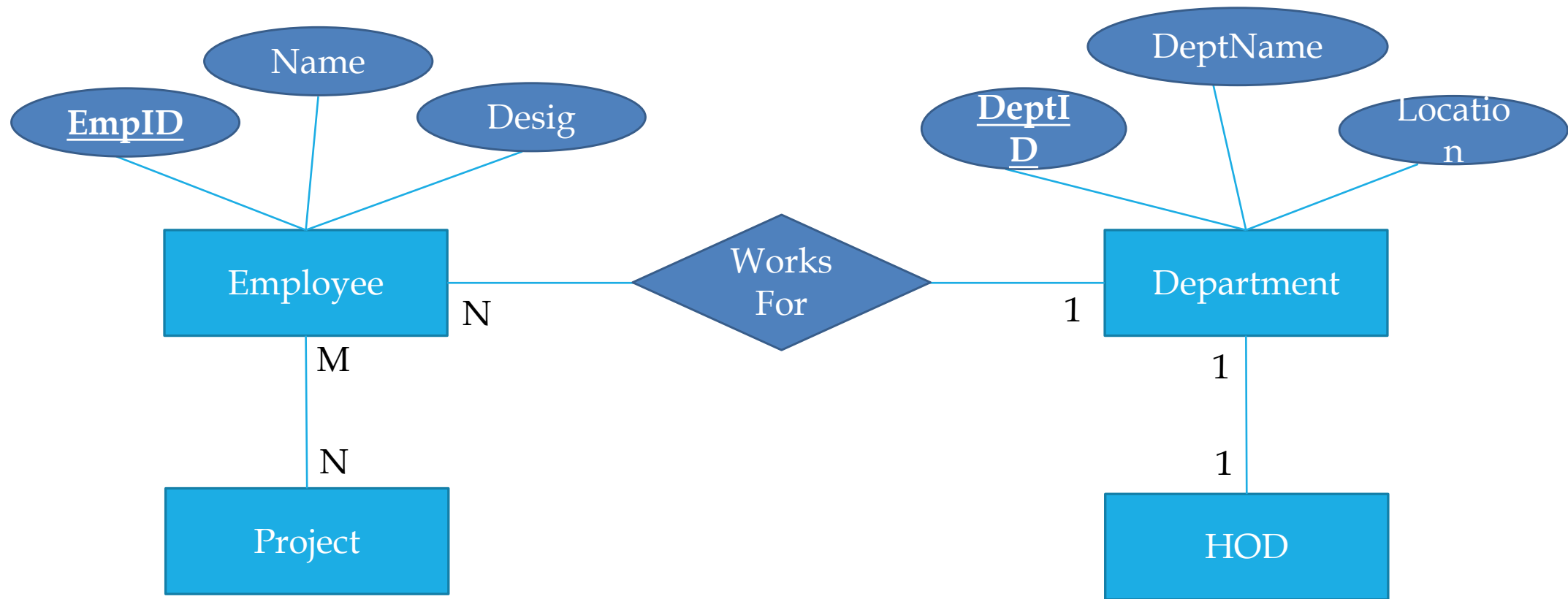
Learn about various symbols available in different ER notations.

Entity Relationship (ER) Model

Basic Symbols	Symbol	Example
Entity	Rectangle	
Attribute	Ellipse	
Link between Entity and Attribute	Line	
Relationship Among Entities	Diamond	

Simple ER Depiction

Identify Entities, Relation among entities and Attributes in a System/Business and create an ER Diagram.



Cardinality Ratio of Relationship

Relationship among entities can be represented in following three ways –

- One-to-one
- One-to-many or Many-to-one
- Many-to-many

Find out at least 3
examples of each
type of Relationship
in real world

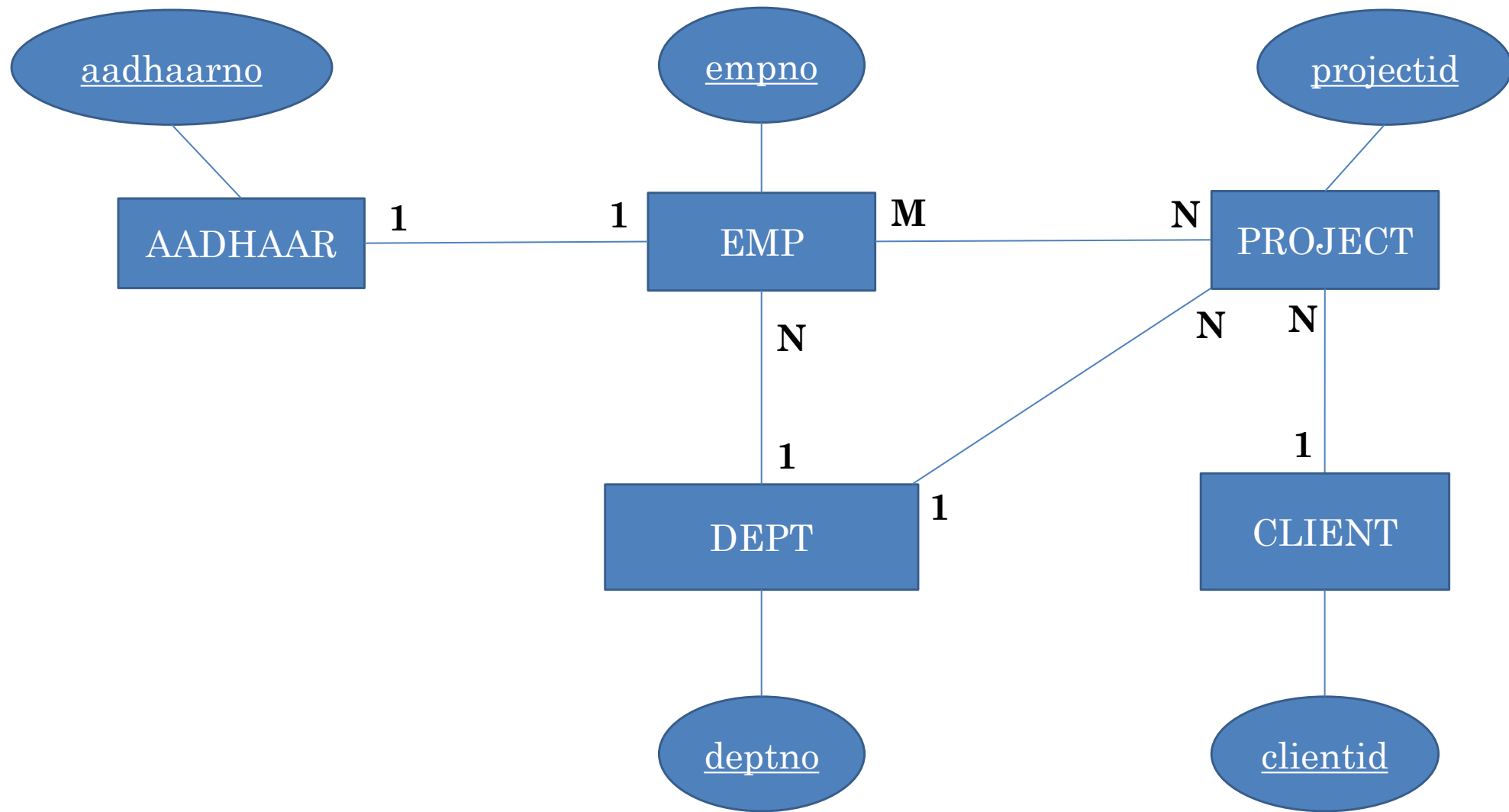
ER Diagram...more

- An entity that can not be identified by its attributes alone is called as **Weak Entity**. Its existence depends on another entity – it is represented by **double border rectangle**.
- **Multivalued attribute** (an attribute representing more than one value for a single entity) is represented by **double line ellipse**.
- **Inheritance – is A** – represented by a Triangle

ER Diagram...more

- Entities are generally Nouns. Relationships are Verbs
- Draw relationships from Left to Right and Top to Bottom.
- Ternary or n-nary Relationship can be depicted in the same way as Binary.
- Notes and comments may be added to ERD. But it should be done in a clean way.

The ultimate goal of creating an ERD is to help understanding the project. Hence easy information representation should be given priority over technical (notational) correctness.



ERD to Relational Mapping

• RELATION	PRIMARY KEY	FOREIGN KEY
• EMP	<u>empno</u>	deptno,
• AADHAAR	<u>aadhaarno</u>	empno
• DEPT	<u>deptno</u>	
• PROJECT	<u>projectid</u>	deptid, clientid
• CLIENT	<u>clientid</u>	
• EMP_PROJECT	<u>empno, projectid</u>	empno, projectid

Database Designing for RDBMS

- Gather Functional Requirements (Data Needs)
- Convert Data Needs into a Data Model (ER Model)
- Convert the ER Model to Relational Model
- **Normalize**
- Implement the Relational Model using RDBMS of your Choice.

→ SQL