JGI JAIN DEEMED-TO-BE UNIVERSITY | SCHOOL OF COMPUTER SCIENCE AND IT

Master of Computer Applications

23MCAC201 –
Advanced Database Management Systems

Credits: 3

L: T: P: E – 3-0-0-3

Prepared by

Dr. A. Rengarajan, Professor

Course Outcome (CO)

CO1	Demonstrate the Relational Database model with appropriate concepts.
CO2	Analyze normalization concepts to design accurate databases.
CO3	Apply the Transactions, concurrency control and recovery in DBMS.
CO4	Evaluate the requirement of parallel, Distributed Database Concepts
CO5	Implement the Applications using Advanced Database Concepts.

Text Books:

1. Abraham Silberschatz, Henry F.Korth and S.Sudarshan, (2013), "Database System Concepts", McGraw Hill 6th Edition. ISBN: 9780077418007 (Module: 1 & 2)
2. Elmasri and Navathe, (2015), "Fundamentals of Database Systems", Addison- Wesley 7th Edition. ISBN: 9780133970777 (Module: 3)
3. Tamer M, Patrick., (2011), "Principles of Distributed Database System", 3rd edition Springer. ISBN: 978-1441988331 (Module: 4 & 5)

Module – 1

RELATIONAL MODEL

- Overview
- Database languages
- Queries
- Client/Server Architecture
- Design with ER & EER Model
- Conceptual Design for Large Enterprises
- The Relational Model Integrity Constraints
- Key Constraints
- Relational Algebra
- Relational Calculus

Introduction

- ✓ **Database:** It is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. It is also used to organize the data in the form of a table, schema, views, and reports, etc.
  - ❖ For example: The college Database organizes the data about the admin, staff, students and faculty etc.
  - ❖ Using the database, you can easily retrieve, insert, and delete the information.
- ✓ **DBMS:** Database management system is a software which is used to manage the database. For example: MySQL, Oracle, etc., are a very popular commercial database which is used in different applications.

Introduction

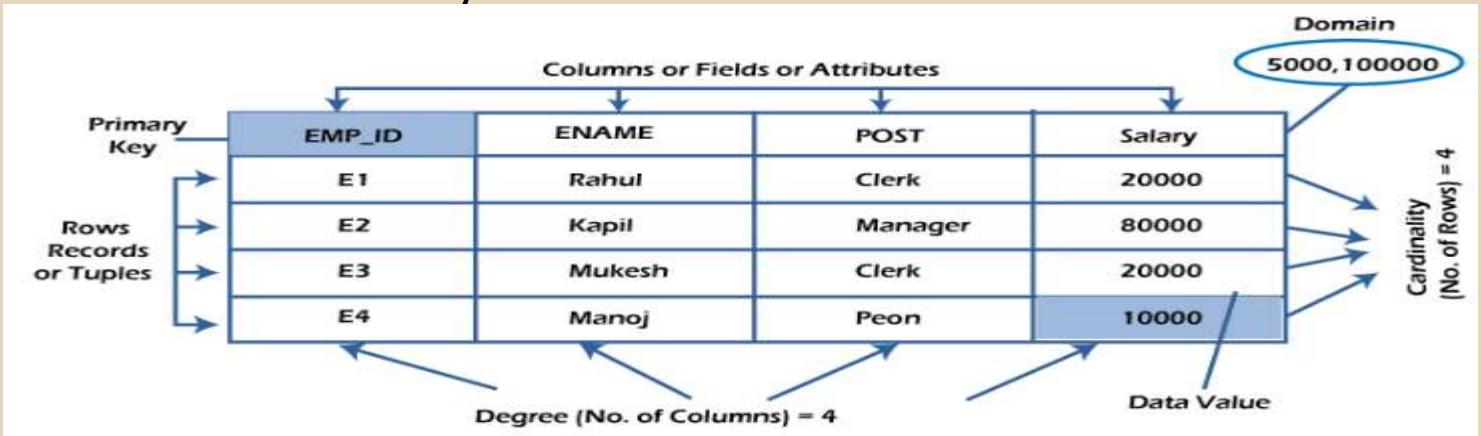
- ✓ DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.
- ✓ It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.
- ✓ DBMS allows users the following tasks:
  - ❖ Data Definition
  - ❖ Data Updation
  - ❖ Data Retrieval
  - ❖ User Administration

Introduction

- ✓ Characteristics of DBMS:
  - ❖DBMS contains automatic backup and recovery procedures.
  - ❖It can reduce the complex relationship between data.
  - ❖It is used to support manipulation and processing of data.
  - ❖It is used to provide security of data.
- ✓ Advantages of DBMS:
  - ❖Controls database redundancy
  - ❖Data Sharing
  - ❖Easily Maintenance
  - ❖Reduce time
  - ❖Backup & Multiple user interface
- ✓ Disadvantages of DBMS:
  - ❖Cost of hardware and software
  - ❖Size, High impact of failure & Complexity

Introduction

- ✓ **RDBMS:** A relational database is the most commonly used database. It contains several tables, and each table has its primary key.
- ✓ Properties of a Relation:
  - ❖ Each relation has a unique name by which it is identified in the DB.
  - ❖ Relation does not contain duplicate tuples.
  - ❖ The tuples of a relation have no specific order.
  - ❖ All attributes in a relation are atomic, i.e., each cell of a relation contains exactly one value.



DBMS vs RDBMS

JGI JAIN DEEMED-TO-BE UNIVERSITY | SCHOOL OF COMPUTER SCIENCE AND IT

S. No	DBMS	RDBMS
1	DBMS applications store **data as file**.	RDBMS applications store **data in a tabular form**.
2	In DBMS, data is generally stored in either a hierarchical form or a navigational form.	In RDBMS, the tables have an identifier called primary key and the data values are stored in the form of tables.
3	**Normalization is not** present in DBMS.	**Normalization is** present in RDBMS.
4	DBMS does **not apply any security** with regards to data manipulation.	RDBMS **defines the integrity constraint** for the purpose of ACID (Atomicity, Consistency, Isolation and Durability) property.
5	DBMS **does not support distributed database**.	RDBMS **supports distributed database**.
6	DBMS is meant to be for small organization and **deal with small data**. it supports **single user**.	RDBMS is designed to **handle large amount of data**. it supports **multiple users**.
7	Examples of DBMS are file systems, **xml** etc.	Example of RDBMS are **mysql**, **postgre**, **sql server**, **oracle** etc.

Relational Model – Overview

- ✓ Relational model can represent as a table with columns and rows. Each row is known as a tuple. Each table of the column has a name or attribute.
- ✓ **Domain:** It contains a set of atomic values that an attribute can take.
- ✓ **Attribute:** It contains the name of a column in a particular table. Each attribute Ai must have a domain, dom(Ai)
- ✓ **Relational instance:** In the relational database system, the relational instance is represented by a finite set of tuples. Relation instances do not have duplicate tuples.

Relational Model – Overview

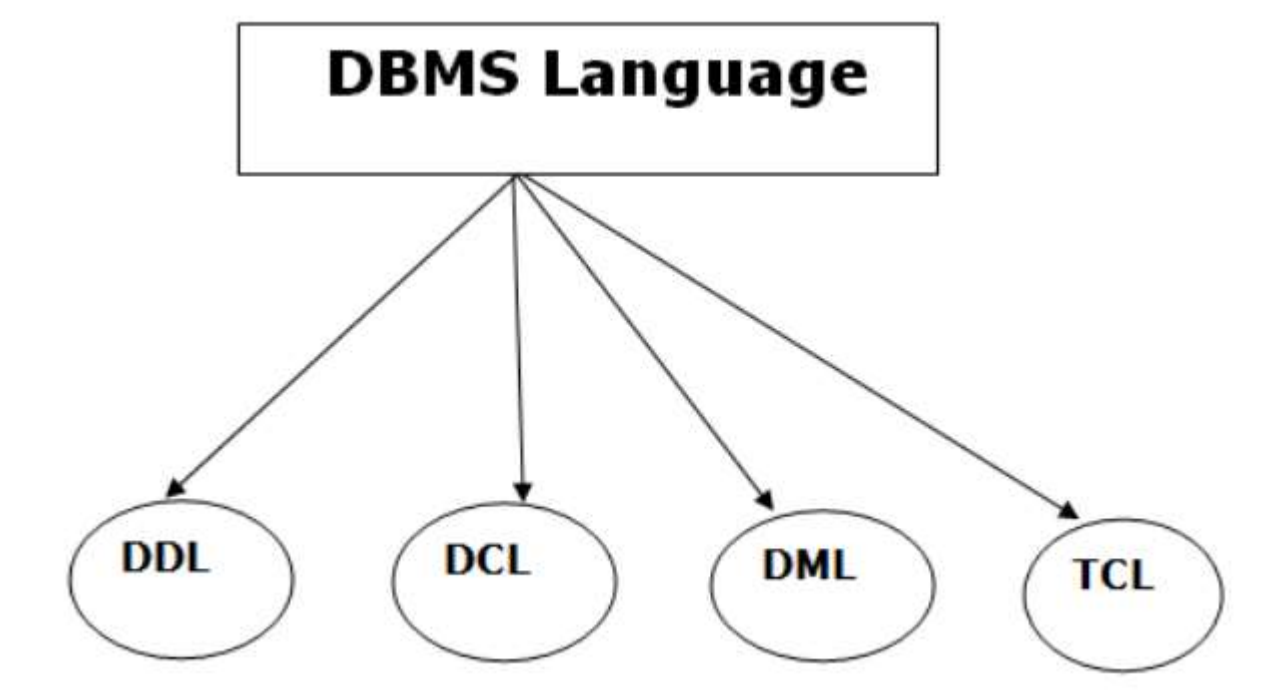
- ✓ **Relational schema:** A relational schema contains the name of the relation and name of all columns or attributes.
- ✓ **Relational key:** In the relational key, each row has one or more attributes. It can identify the row in the relation uniquely.

Example: STUDENT Relation

NAME	ROLL_NO	PHONE_NO	ADDRESS	AGE
Ram	14795	7305758992	Noida	24
Shyam	12839	9026288936	Delhi	35
Laxman	33289	8583287182	Gurugram	20
Mahesh	27857	7086819134	Ghaziabad	27
Ganesh	17282	9028 9i3988	Delhi	40

Database Language

- ✓ A DBMS has appropriate languages and interfaces to express database queries and updates.
- ✓ Database languages can be used to read, store and update the data in the database.



Database Language

- ✓ **DDL (Data Definition Language):** It is used to define database structure or pattern.
- ✓ It is used to create schema, tables, indexes, constraints, etc. in the database.
- ✓ Using the DDL statements, you can create the skeleton of the database.
- ✓ Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

Database Language

- ✓ **DDL Commands:** These commands are used to update the database schema that's why they come under Data definition language.
    1. **Create:** It is used to create objects in the database.
    2. **Alter:** It is used to alter the structure of the database.
    3. **Drop:** It is used to delete objects from the database.
    4. **Truncate:** It is used to remove all records from a table.
    5. **Rename:** It is used to rename an object.
    6. **Comment:** It is used to comment on the data dictionary.

Database Language

- ✓ **DML (Data Manipulation Language):** It is used for accessing and manipulating data in a database. It handles user requests. The commands are,
    1. **Select:** It is used to retrieve data from a database.
    2. **Insert:** It is used to insert data into a table.
    3. **Update:** It is used to update existing data within a table.
    4. **Delete:** It is used to delete all records from a table.
    5. **Merge:** It performs UPSERT operation, i.e., insert or update operations.
    6. **Call:** It is used to call a structured query language or a Java subprogram.
    7. **Explain Plan:** It has the parameter of explaining data.
    8. **Lock Table:** It controls concurrency.

Database Language

- ✓ **DCL (Data Control Language):** It is used to retrieve the stored or saved data.
- ✓ The DCL execution is transactional. It also has rollback parameters.
    1. **Grant:** It is used to give user access privileges to a database.
    2. **Revoke:** It is used to take back permissions from the user.
- ✓ There are the following operations which have the authorization of Revoke:

    CONNECT, INSERT, USAGE, EXECUTE, DELETE, UPDATE and SELECT.

Database Language

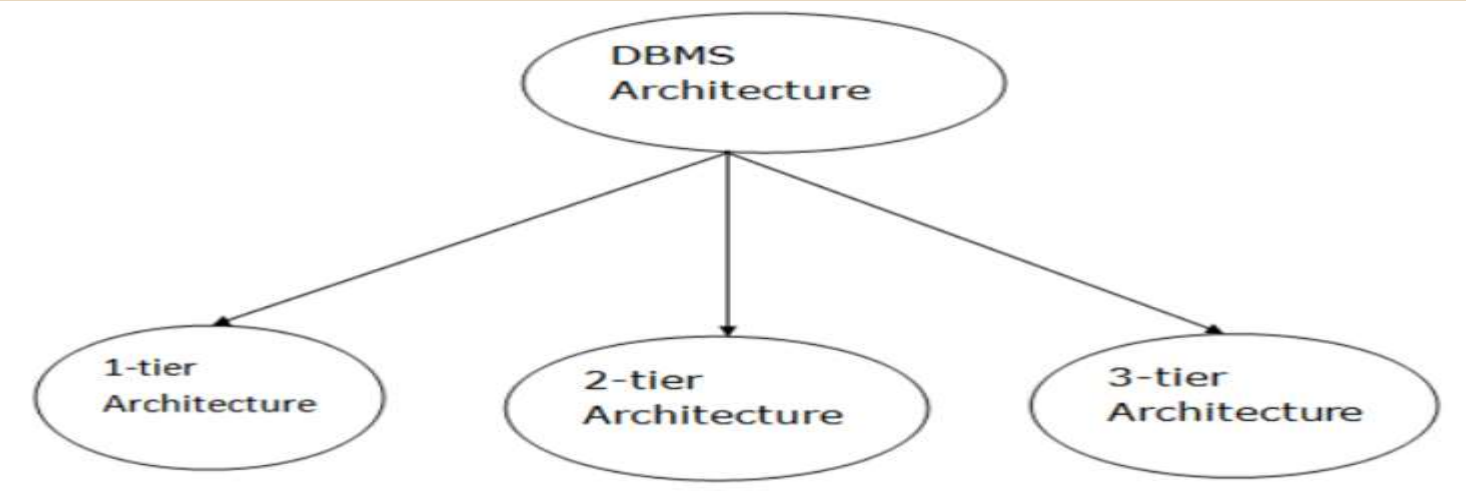
- **TCL (Transaction Control Language):** TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction.
    1. **Commit:** It is used to save the transaction on the database.
    2. **Rollback:** It is used to restore the database to original since the last Commit.

Queries

- ✓ In a relational database, which contains records or rows of information, the **SQL SELECT statement query** allows users to choose data and return it from a database to an application. The resulting query is stored in a result table, which is called a result-set.

Client / Server Architecture

- ✓ The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.
- ✓ The client/server architecture consists of many PCs and a workstation which are connected via the network.
- ✓ DBMS architecture depends upon how users are connected to the database to get their request done.



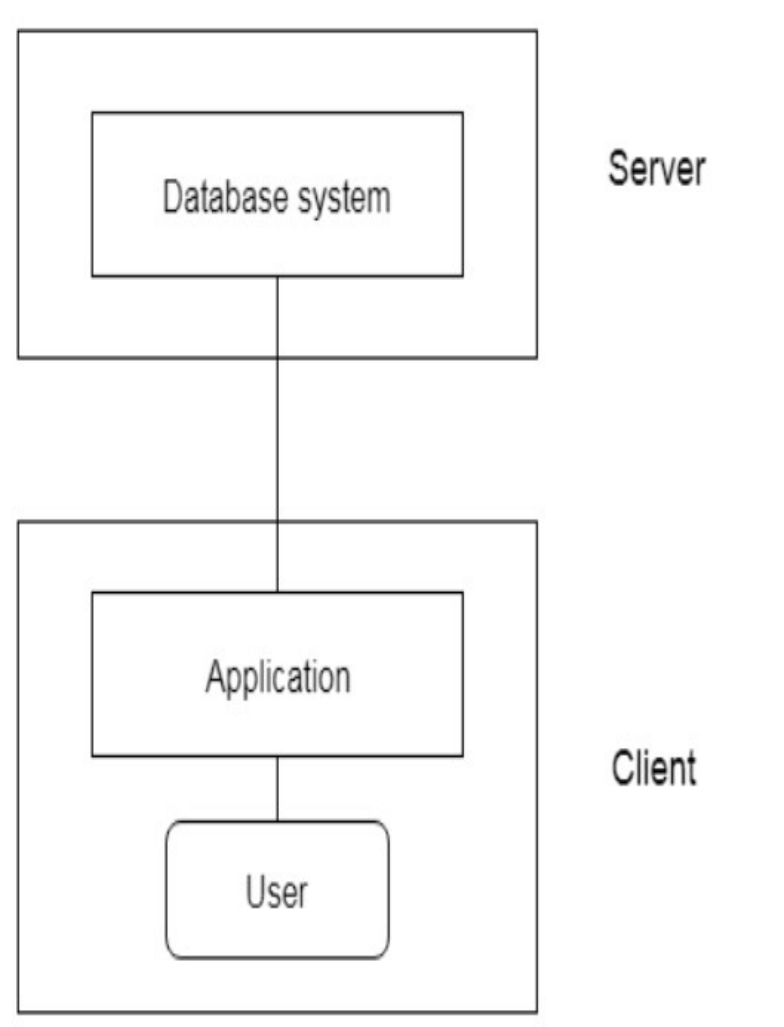
2 Tier Architecture

- ✓ The **2-Tier architecture** is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. For this interaction, API's like: ODBC, JDBC are used.
- ✓ The user interfaces and application programs are run on the client-side.
- ✓ The server side is responsible to provide the functionalities like: query processing and transaction management.
- ✓ To communicate with the DBMS, client-side application establishes a connection with the server side.

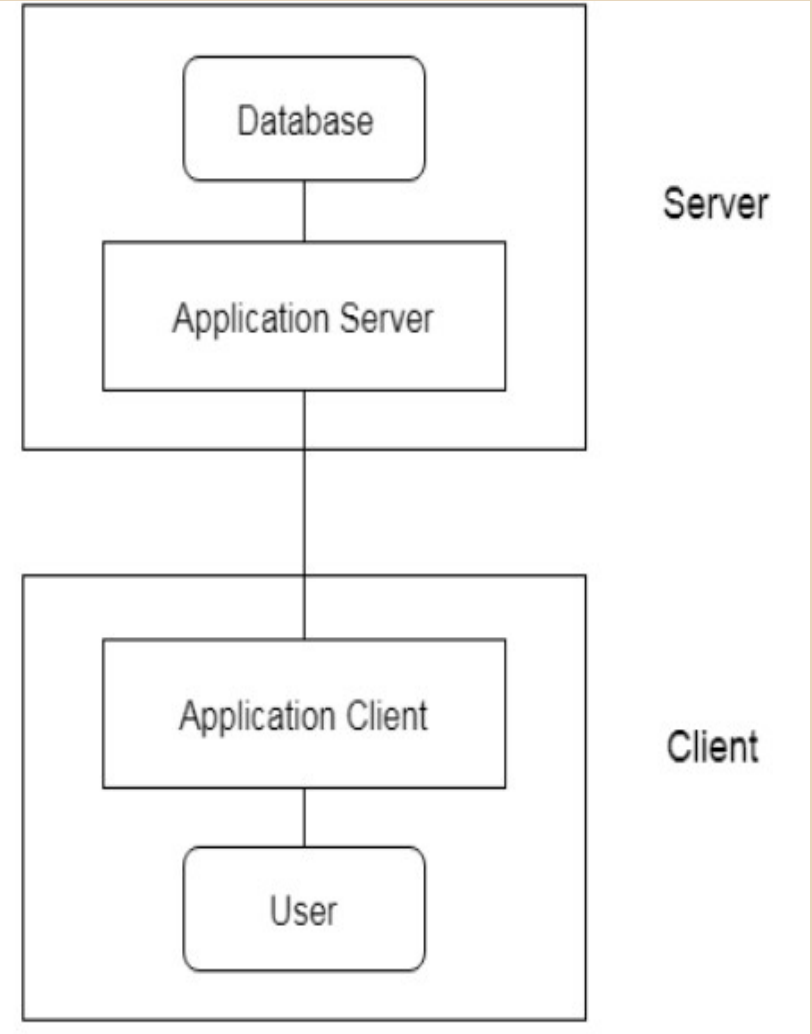
3 Tier Architecture

- ✓ The **3-Tier architecture** contains another layer between the client and server. In this architecture, client can't directly communicate with the server.
- ✓ The application on the client-end interacts with an application server which further communicates with the database system.
- ✓ End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.
- ✓ The 3-Tier architecture is used in case of large web application.

2 Tier Vs 3 Tier Architecture



2-tier Architecture



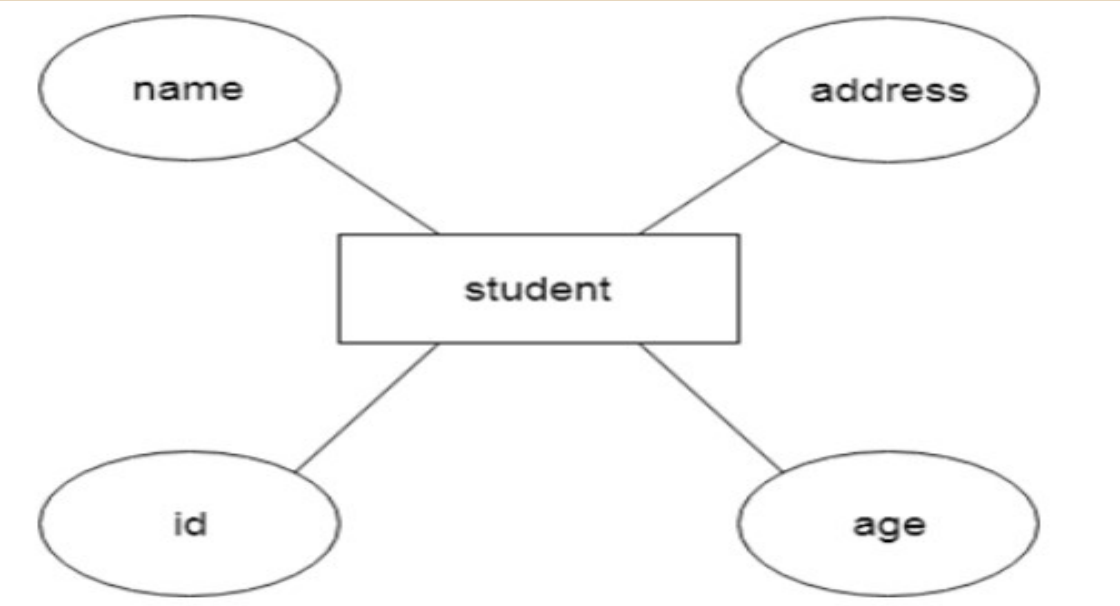
3-tier Architecture

Design with ER Model

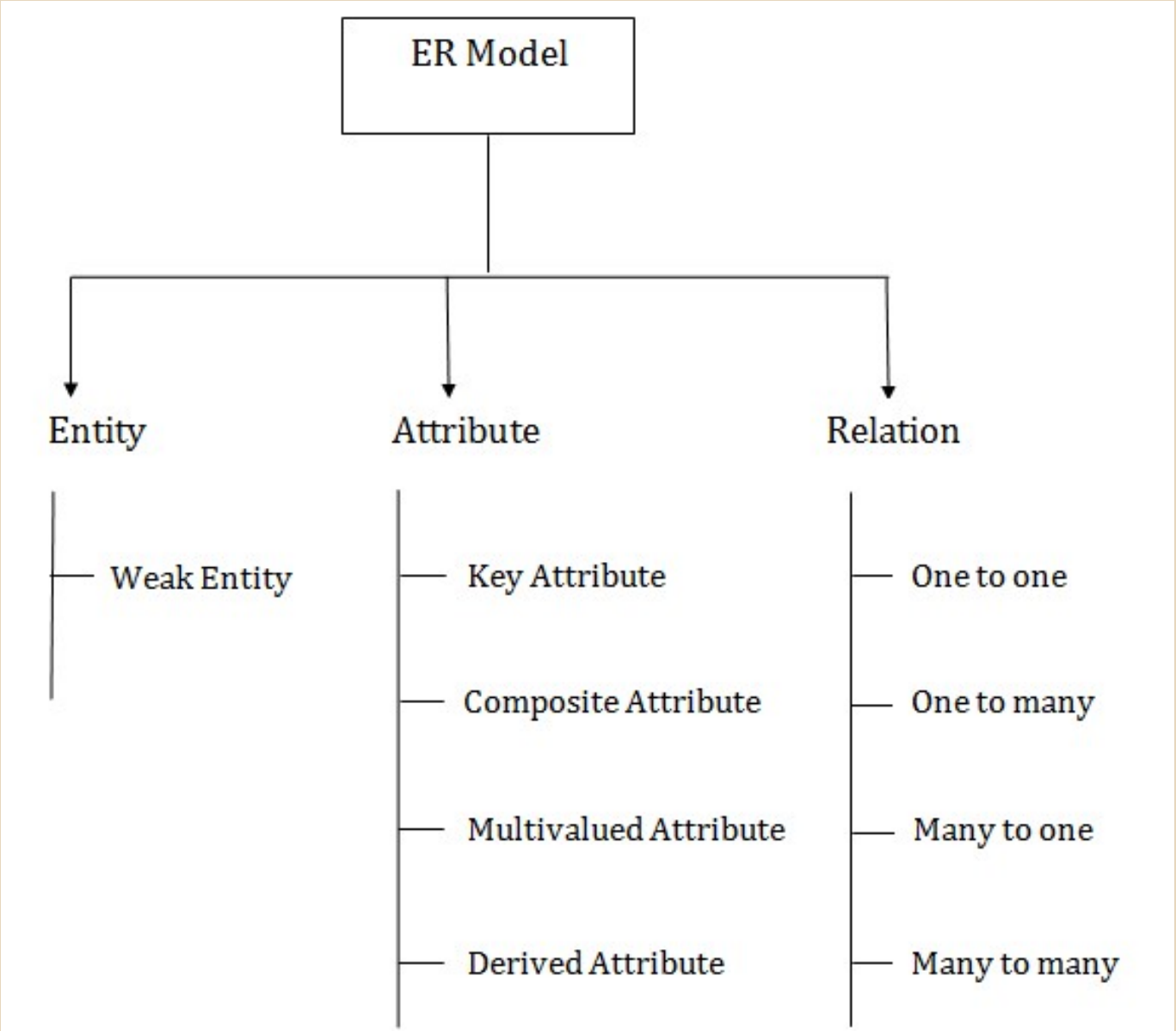
- ✓ **ER model** stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.
- ✓ It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.
- ✓ In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.

Design with ER Model

- ✓ **For example,** Suppose we design a school database. In this database, the student will be an entity with attributes like address, name, id, age, etc.
- ✓ The address can be another entity with attributes like city, street name, pin code, etc., and there will be a relationship between them.

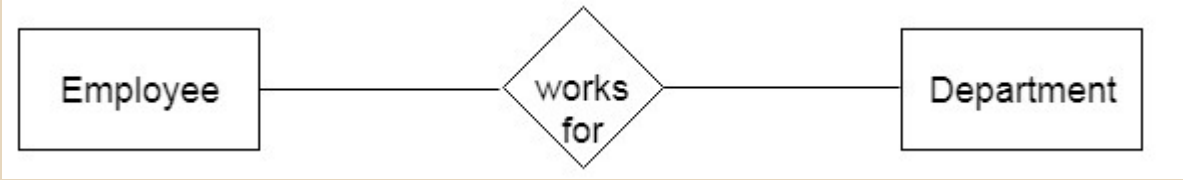


Components of ER Model



Components of ER Model

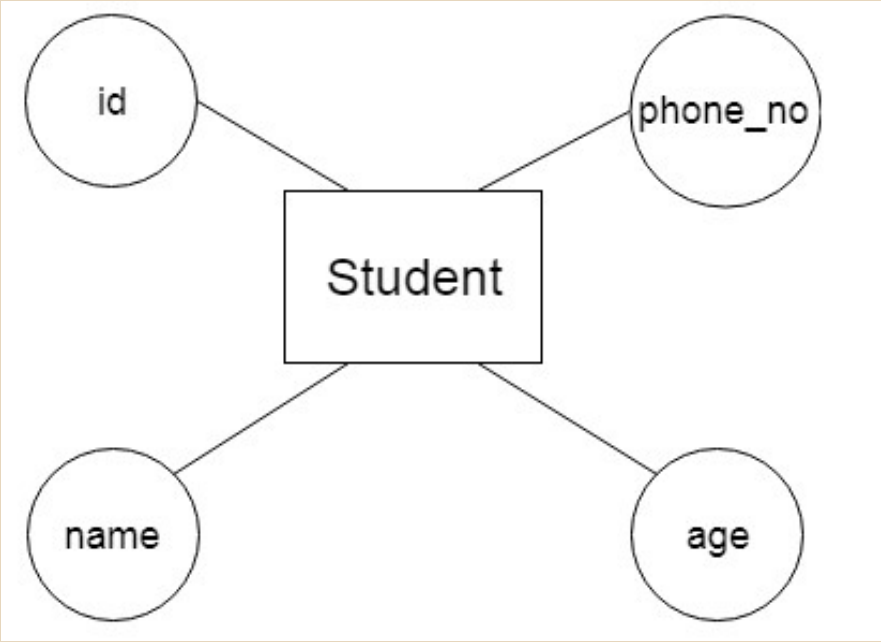
- ✓ **Entity:** An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.
- ✓ Consider an organization as an example- manager, product, employee, department etc. can be taken as an entity.



- ✓ **Weak Entity:** An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.

Components of ER Model

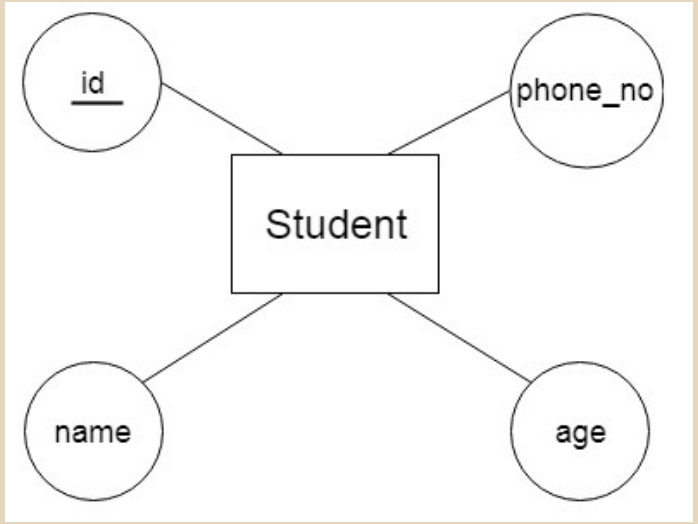
- ✓ **Attribute:** The attribute is used to describe the property of an entity. Ellipse is used to represent an attribute.
- ✓ For example, id, age, contact number, name, etc. can be attributes of a student.



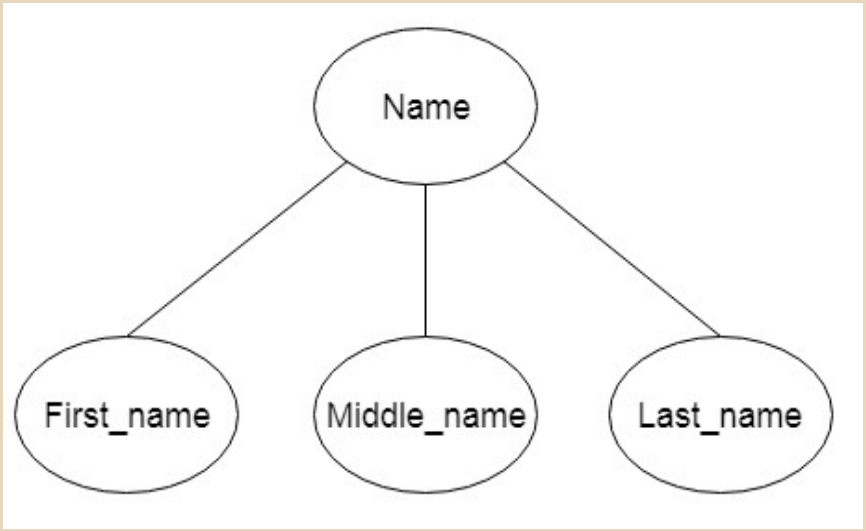
Components of ER Model

- ✓ **Key Attribute:** The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.
- ✓ **Composite Attribute:** An attribute that composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.
- ✓ **Multivalued Attribute:** An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent multivalued attribute.
- ✓ **Derived Attribute:** An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.

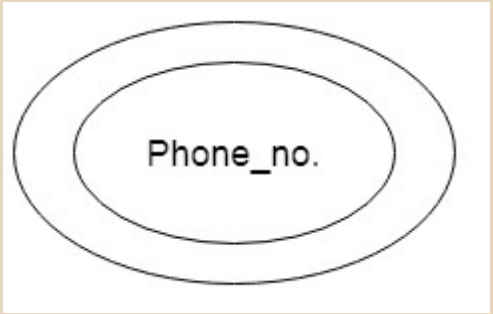
Components of ER Model



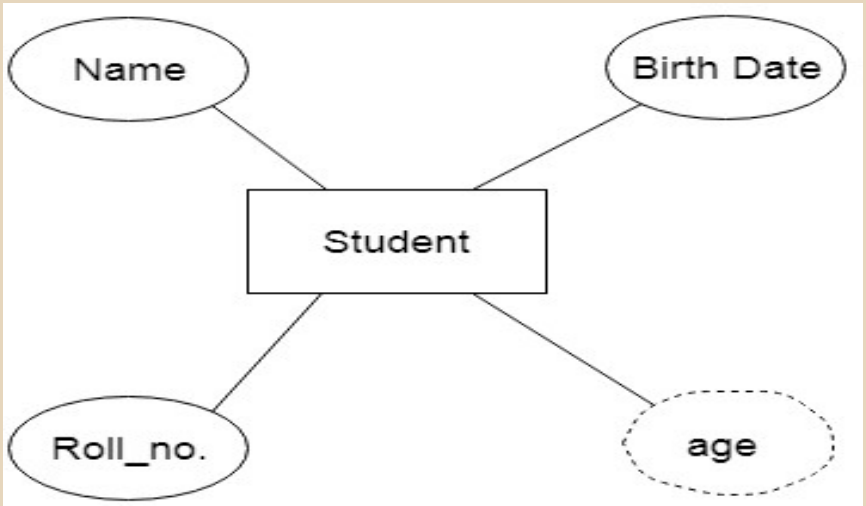
Key Attribute



Composite Attribute



Multivalued Attribute



Derived Attribute

Components of ER Model

- **Relationship:** A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

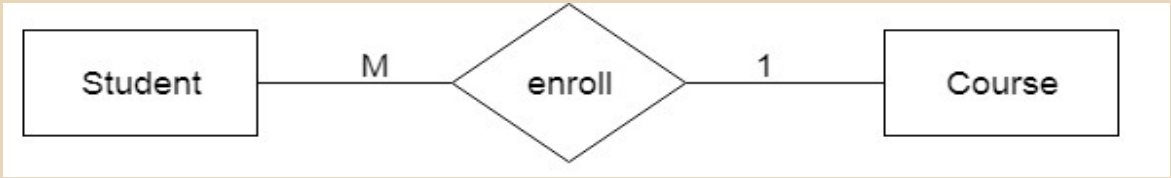
- **One-to-One Relationship:** When only one instance of an entity is associated with the relationship, then it is known as one to one relationship.
- **One-to-Many Relationship:** When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a one-to-many relationship.
- **Many-to-One Relationship:** When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.

Components of ER Model

- ✓ **Many-to-Many Relationship:** When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship.

One-to-one

One-to-many



Many-to-one

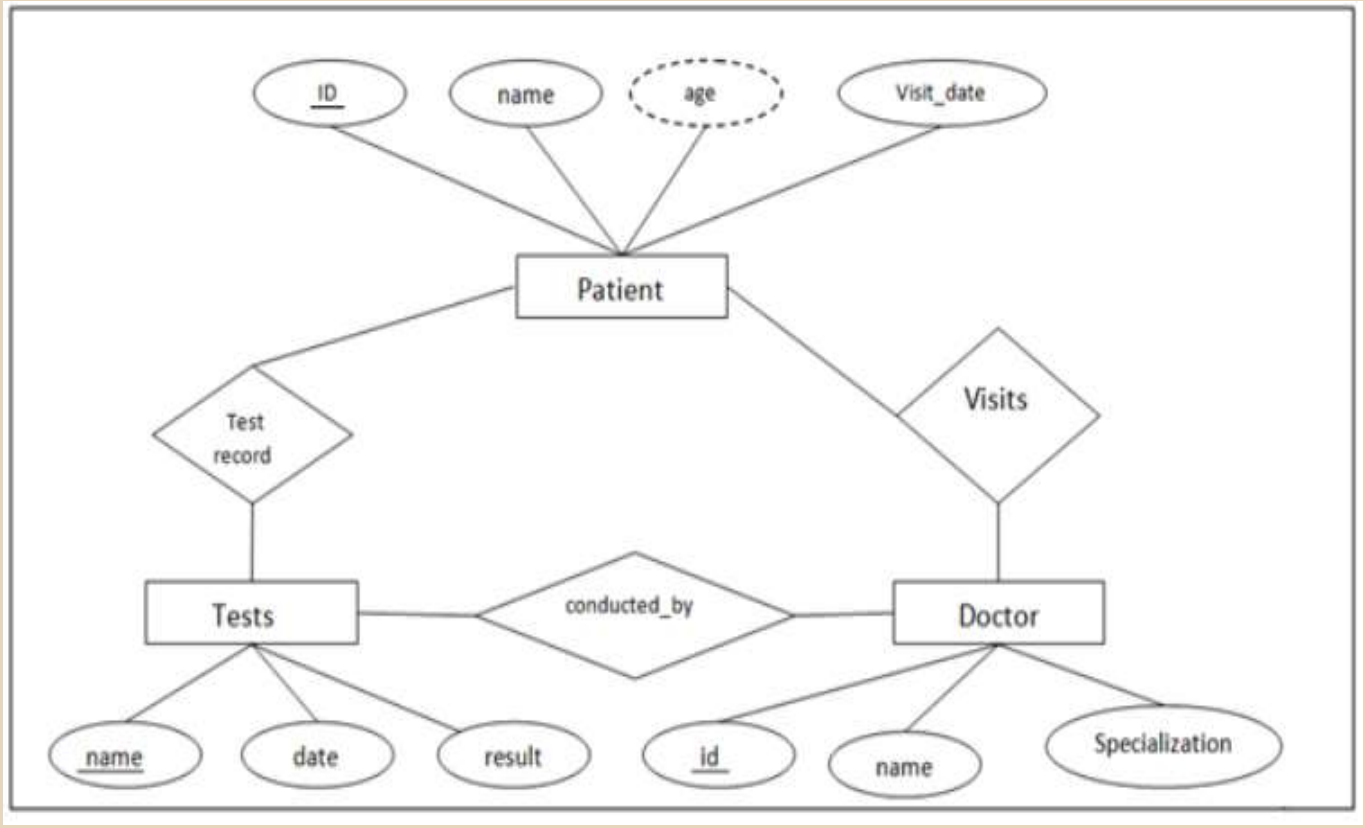
Many-to-many

Example

To create ER diagram for **Hospital Management System.**

Each of these entities have their respective attributes which are −

- Patients - ID(primary key), name, age, visit_date
- Tests- Name(primary key), date, result
- Doctor- ID(primary key), name, specialization

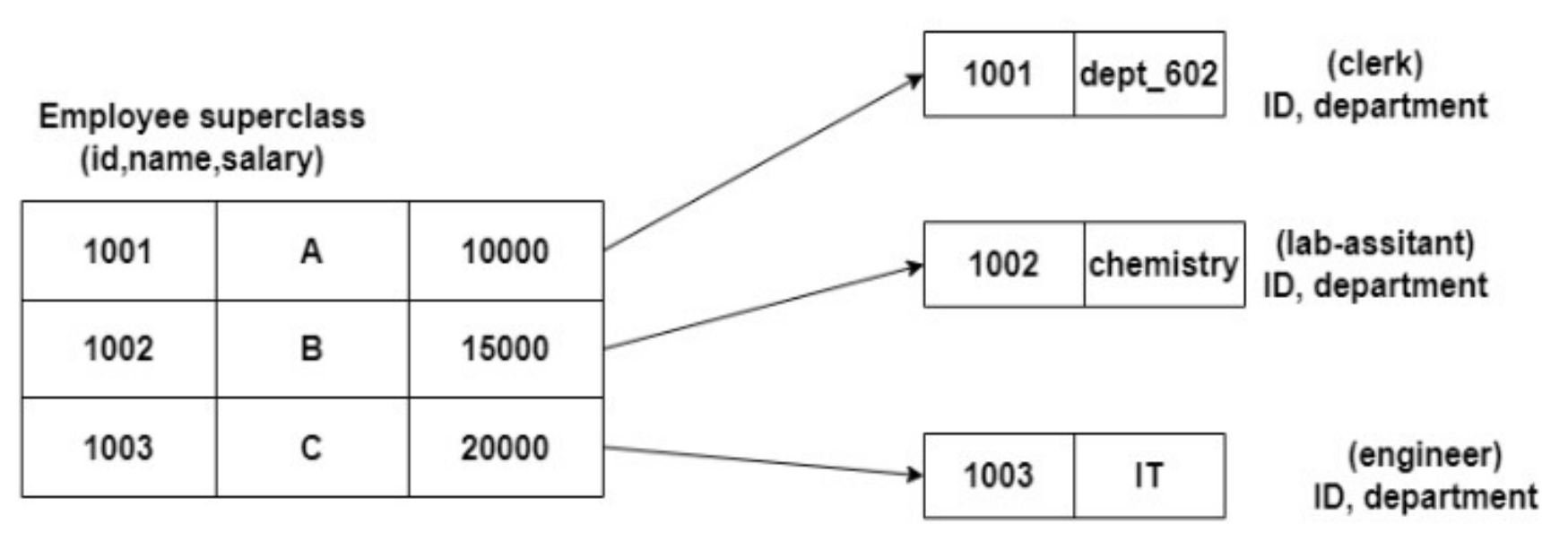


Design with EER Model

- ✓ It is getting harder and harder to apply the conventional ER paradigm for database modeling as data complexity rises today. The existing ER model needs to be enhanced or improved in order for it to better handle the complicated application in order to reduce the modeling complexity.
- ✓ The SubClass and SuperClass, Specialization and Generalization, Union or Category, Aggregation, etc., are displayed using this diagrammatic style.

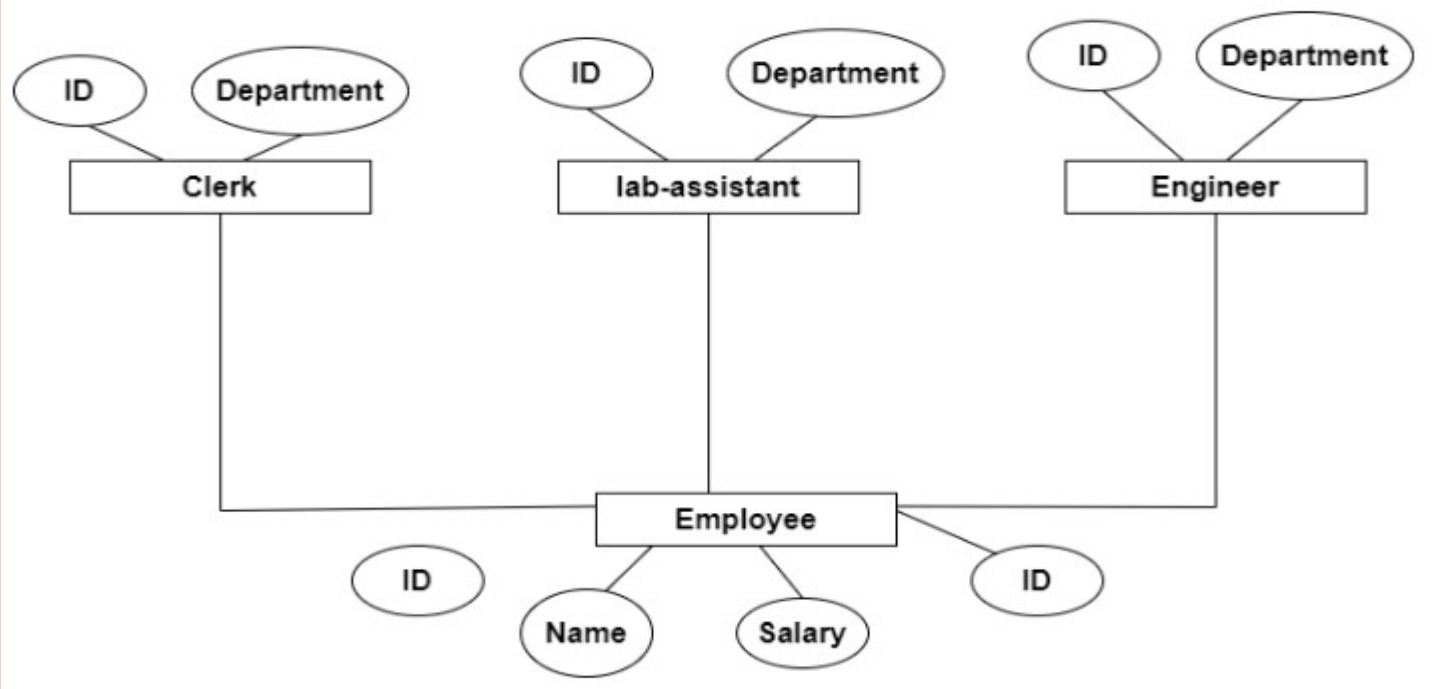
Design with EER Model

- ✓ **Example:** We can draw the ER diagram for these relationships. Let's suppose we have a superclass Employee and subclasses as a clerk, engineer, and lab assistant.



Design with EER Model

- ✓ The Enhanced ER diagram of the above example will look like this:



- ✓ In the above example, we have one superclass and three subclasses. Each subclass inherits all the attributes from its superclass so that a lab assistant will have all its attributes, like its name, salary, etc.

Conceptual Design for Large Enterprises

- ✓ To build the conceptual representation of the database, which has the identification of the important entities, relationships, and attributes.
- ✓ In this design methodology, the process of constructing a model of the data is used in an enterprise, independent of all physical considerations.
- ✓ The conceptual database design phase starts with the formation of a conceptual data model of the enterprise that is entirely independent of implementation details such as the target DBMS, use of application programs, programming languages used, hardware platform, performance issues, or any other physical deliberations.

Steps for Conceptual Database Design

1. Build a conceptual data model
2. Recognize entity types
3. Recognize the relationship types
4. Identify and connect attributes with entity or relationship types
5. Determine attribute domains
6. Determine candidate, primary, and alternate key attributes
7. Consider the use of improved modeling concepts (optional step)
8. Check model for redundancy
9. Validate the conceptual model against user transactions
10. Review the conceptual data model with user

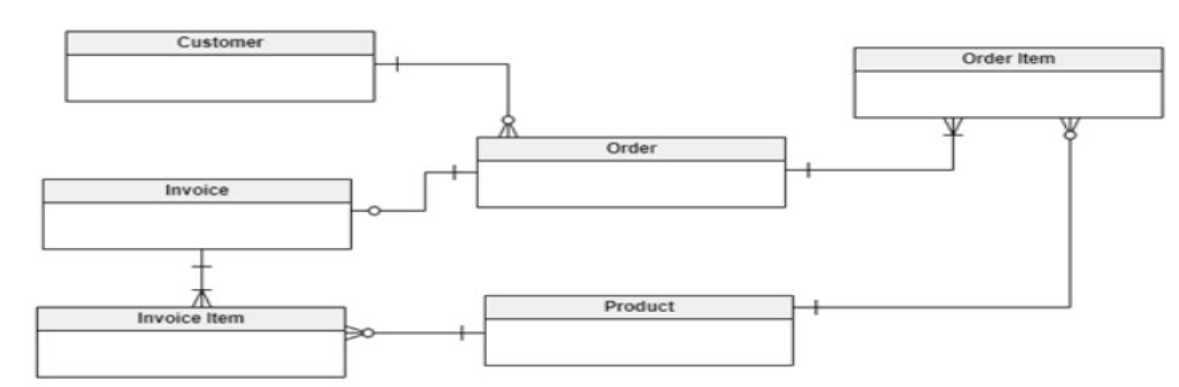
Example

To create a conceptual data model for **simple order management system**.

- The model has six entities (Customer, Order, Order Item, Product, Invoice, and Invoice Item) that represent the real-world information in our physical database.

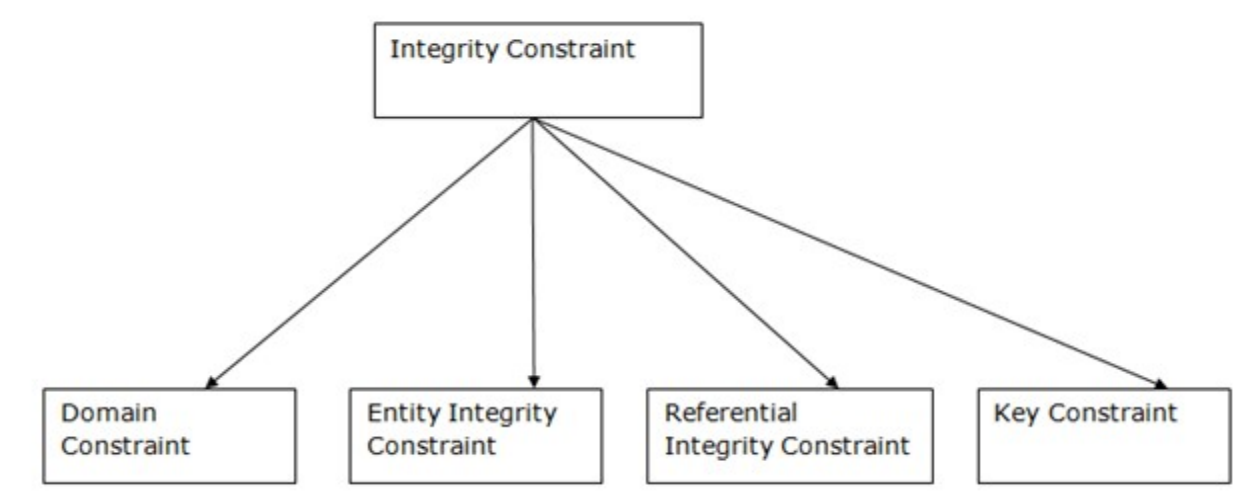
There are six relationships in this data model:

- Customer_Order: A one-mandatory-to-many-optional relationship.
- Order_Order Item: A one-mandatory-to-many-mandatory relationship.
- Product_Order Item: A one-mandatory-to-many-optional relationship.
- Order_Invoice: A one-mandatory-to-one-optional relationship.
- Invoice_Invoice Item: A one-mandatory-to-many-mandatory relationship.
- Product_Invoice Item: A one-mandatory-to-many-optional relationship.



The Relational Model Integrity Constraints

- ✓ Integrity constraints are a set of rules. It is used to maintain the quality of information.
- ✓ Integrity constraints ensure that the data insertion, updating, and other processes have to be performed in such a way that data integrity is not affected.
- ✓ Thus, integrity constraint is used to guard against accidental damage to the database.



Domain Constraints

- ✓ Domain constraints can be defined as the definition of a valid set of values for an attribute.
- ✓ The data type of domain includes string, character, integer, time, date, currency, etc. The value of the attribute must be available in the corresponding domain.
- ✓ For e.g.,

ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1004	Morgan	8th	A

Not allowed. Because AGE is an integer attribute

Entity Integrity Constraints

- ✓ The entity integrity constraint states that primary key value can't be null.
- ✓ This is because the primary key value is used to identify individual rows in relation and if the primary key has a null value, then we can't identify those rows.
- ✓ A table can contain a null value other than the primary key field.
- ✓ For e.g.,

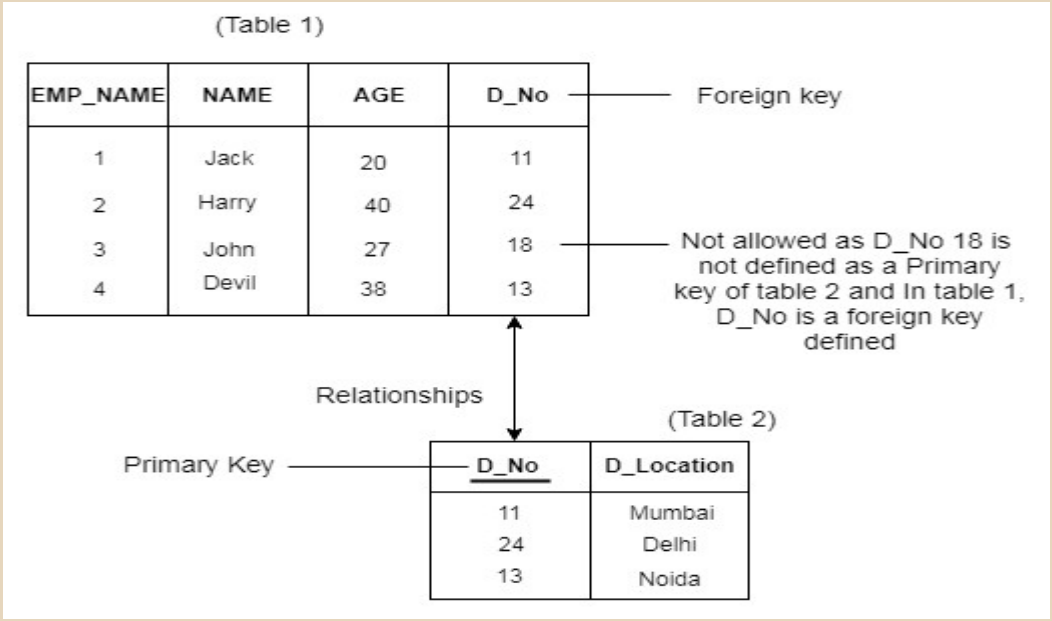
EMPLOYEE

EMP_ID	EMP_NAME	SALARY
123	Jack	30000
142	Harry	60000
164	John	20000
	Jackson	27000

Not allowed as primary key can't contain a NULL value

Referential Integrity Constraints

- ✓ A referential integrity constraint is specified between two tables.
- ✓ In the Referential integrity constraints, if a foreign key in Table 1 refers to the Primary Key of Table 2, then every value of the Foreign Key in Table 1 must be null or be available in Table 2.
- ✓ For e.g.,



Key Constraints

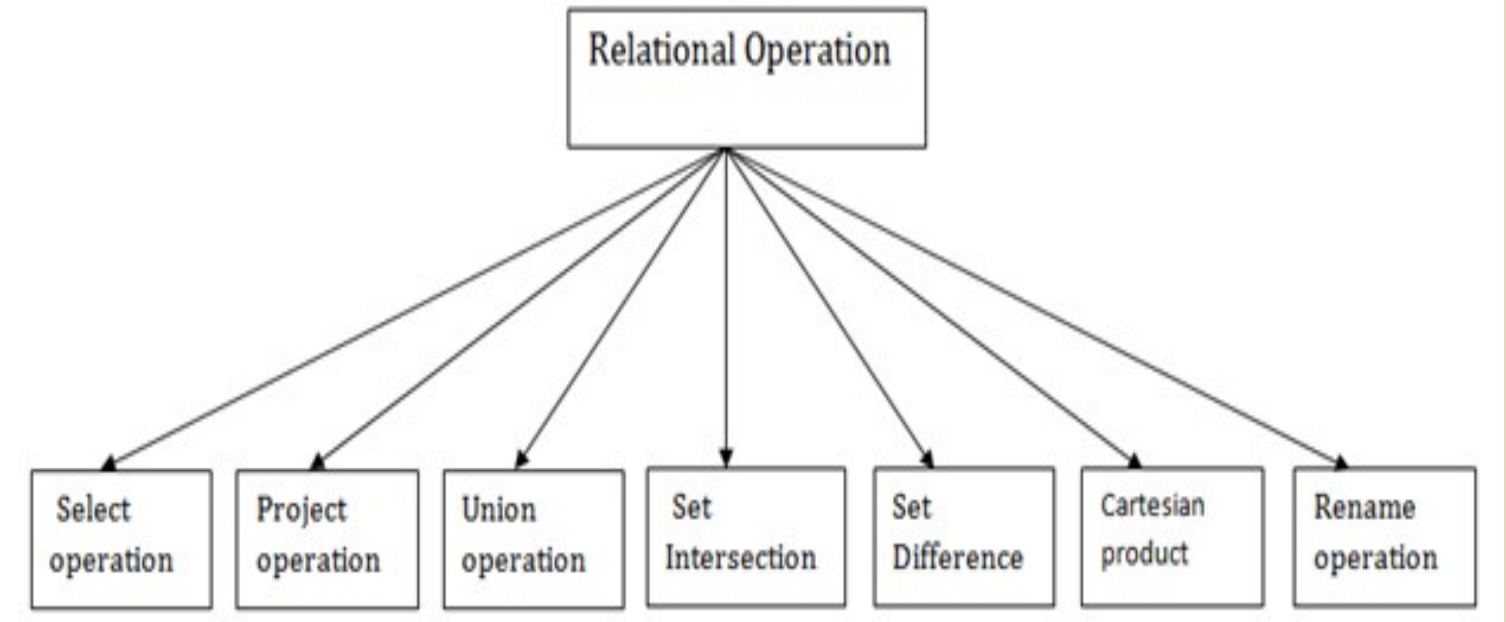
- ✓ Keys are the entity set that is used to identify an entity within its entity set uniquely.
- ✓ An entity set can have multiple keys, but out of which one key will be the primary key. A primary key can contain a unique and null value in the relational table.
- ✓ For e.g.,

ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1002	Morgan	8th	22

Not allowed. Because all row must be unique

Relational Algebra

- ✓ Relational algebra is a procedural query language. It gives a step by step process to obtain the result of the query. It uses operators to perform queries.
- ✓ Types of Relational operation as follows,



1. Select Operation

- ✓ The select operation selects tuples that satisfy a given predicate.
- ✓ It is denoted by sigma (σ).
- ✓ Notation: σ = p(r)
  - ✓ Where,
    - ❖ **σ** is used for selection prediction
    - ❖ **r** is used for relation
    - ❖ **p** is used as a propositional logic formula which may use connectors like: AND OR and NOT. These relational can use as relational operators like =, ≠, ≥, <, >, ≤.

1. Select Operation

JAIN DEEMED-TO-BE UNIVERSITY | SCHOOL OF COMPUTER SCIENCE AND IT

Example: LOAN Relation

BRANCH_NAME	LOAN_NO	AMOUNT
Downtown	L-17	1000
Redwood	L-23	2000
Perryride	L-15	1500
Downtown	L-14	1500
Mianus	L-13	500
Roundhill	L-11	900
Perryride	L-16	1300

Input: σ BRANCH_NAME="perryride" (LOAN)

Output:

BRANCH_NAME	LOAN_NO	AMOUNT
Perryride	L-15	1500
Perryride	L-16	1300

2. Project Operation

- ✓ This operation shows the list of those attributes that we wish to appear in the result. Rest of the attributes are eliminated from the table.
- ✓ It is denoted by ∏.
- ✓ Notation: ∏ A1, A2, An (r)
  - ✓ Where,
    - ❖ **A1**, **A2**, **A3** is used as an attribute name of relation **r**.

2. Project Operation

Example: CUSTOMER RELATION

NAME	STREET	CITY
Jones	Main	Harrison
Smith	North	Rye
Hays	Main	Harrison
Curry	North	Rye
Johnson	Alma	Brooklyn
Brooks	Senator	Brooklyn

Input: ∏ NAME, CITY (CUSTOMER)

Output:

NAME	CITY
Jones	Harrison
Smith	Rye
Hays	Harrison
Curry	Rye
Johnson	Brooklyn
Brooks	Brooklyn

3. Union Operation

- ✓ Suppose there are two tuples R and S. The union operation contains all the tuples that are either in R or S or both in R & S.
- ✓ It eliminates the duplicate tuples. It is denoted by ∪.
- ✓ Notation: R ∪ S

4. Set Intersection

- ✓ Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in both R & S.
- ✓ It is denoted by intersection ∩.
- ✓ Notation: R ∩ S

5. Set Difference

- ✓ Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in R but not in S.
- ✓ It is denoted by intersection minus (-).
- ✓ Notation: R - S

6. Cartesian Product

- ✓ The Cartesian product is used to combine each row in one table with each row in the other table. It is also known as a cross product.
- ✓ It is denoted by X.
- ✓ Notation: E X D

7. Rename Operation

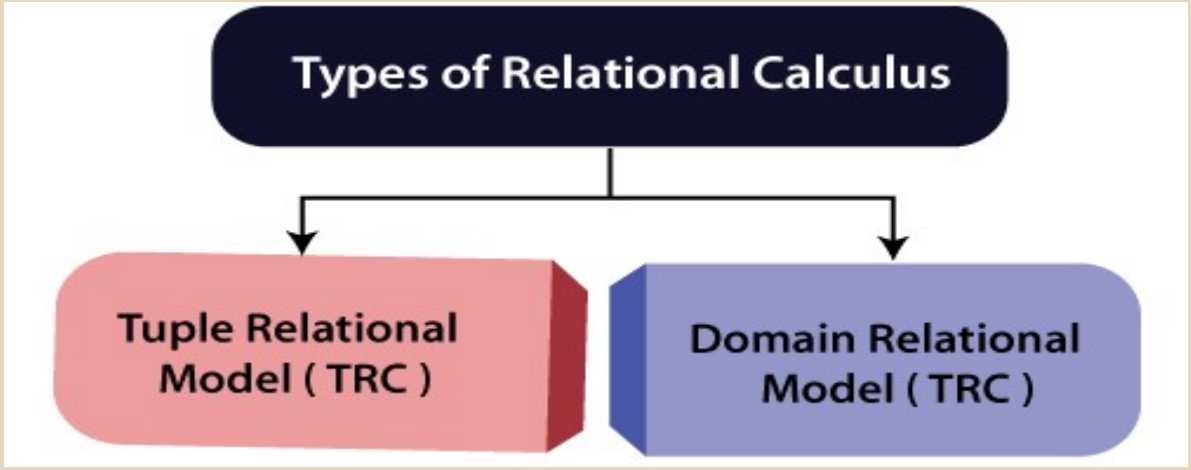
- ✓ The rename operation is used to rename the output relation. It is denoted by **rho** (ρ).
- ✓ **Example:** We can use the rename operator to rename STUDENT relation to STUDENT1.
- ✓ ρ(STUDENT1, STUDENT)

Relational Calculus

- ✓ There is an alternate way of formulating queries known as Relational Calculus. Relational calculus is a non-procedural query language.
- ✓ In the non-procedural query language, the user is concerned with the details of how to obtain the end results. The relational calculus tells **what to do** but never explains **how to do**.
- ✓ Most commercial relational languages are based on aspects of relational calculus including SQL-QBE (Query By Example) and QUEL (QUEry Language).
- ✓ QUEL is a data definition and data manipulation for INGRES (INteractive Graphics and REtrieval System).

Relational Calculus

- ✓ Many of the calculus expressions involves the use of Quantifiers. There are two types of quantifiers:
  1. **Universal Quantifiers:** The universal quantifier denoted by ∀ (for all) is read as for all which means that in a given set of tuples exactly all tuples satisfy a given condition.
  2. **Existential Quantifiers:** The existential quantifier denoted by ∃ (there exist/at least one) is read as for all which means that in a given set of tuples there is at least one occurrences whose value satisfy a given condition.



TRC

- ✓ It is a non-procedural query language which is based on finding a number of tuple variables also known as range variable for which predicate holds true.
- ✓ It describes the desired information without giving a specific procedure for obtaining that information.
- ✓ The tuple relational calculus is specified to select the tuples in a relation. In TRC, filtering variable uses the tuples of a relation. The result of the relation can have one or more tuples.
- ✓ Notation: {T | P (T)} or {T | Condition (T)}
  - ✓Where
    - ❖**T** is the resulting tuples
    - ❖**P(T)** is the condition used to fetch T.

DRC

- ✓ The second form of relation is known as Domain relational calculus. In domain relational calculus, filtering variable uses the domain of attributes.
- ✓ Domain relational calculus uses the same operators as tuple calculus. It uses logical connectives ∧ (and), ∨ (or) and ¬ (not).
- ✓ It uses Existential (∃) and Universal Quantifiers (∀) to bind the variable. The QBE or Query by example is a query language related to domain relational calculus.
- ✓ Notation: { a1, a2, a3, ..., an | P (a1, a2, a3, ... ,an)}
  - ✓ Where
    - ❖ **a1, a2** are attributes
    - ❖ **P** stands for formula built by inner attributes

Procedural Vs Non-Procedural

JAIN DEEMED-TO-BE UNIVERSITY | SCHOOL OF COMPUTER SCIENCE AND IT

S. No	Procedural	Non-Procedural
1	It is command-driven language.	It is a function-driven language
2	It works through the state of machine.	It works through the mathematical functions.
3	Its semantics are quite tough.	Its semantics are very simple.
4	It returns only restricted data types and allowed values.	It can return any datatype or value
5	Overall efficiency is very high.	Overall efficiency is low as compared to Procedural Language.
6	Size of the program written in Procedural language is large.	Size of the Non-Procedural language programs are small.
7	It is not suitable for time critical applications.	It is suitable for time critical applications.
8	Iterative loops and Recursive calls both are used in the Procedural languages.	Recursive calls are used in Non-Procedural languages.
9	**Examples:** FORTRAN, COBOL, ALGOL, BASIC, C and Pascal.	**Examples:** SQL, PROLOG, LISP.

Possible Interview Questions

JGI JAIN DEEMED-TO-BE UNIVERSITY SCHOOL OF COMPUTER SCIENCE AND IT

1. What is DBMS?
2. What is a database?
3. What is a database system?
4. What are the advantages of DBMS?
5. What is a checkpoint in DBMS?
6. When does checkpoint occur in DBMS?
7. What do you mean by transparent DBMS?
8. What are the unary operations in Relational Algebra?
9. What is RDBMS?
10. How many types of database languages are?
11. What do you understand by Data Model?
12. Define a Relation Schema and a Relation.

Possible Interview Questions

JGI JAIN DEEMED-TO-BE UNIVERSITY SCHOOL OF COMPUTER SCIENCE AND IT

13. What is a degree of Relation?
14. What is the Relationship?
15. What are the disadvantages of file processing systems?
16. What is data abstraction in DBMS?
17. What are the three levels of data abstraction?
18. What is Relational Algebra?
19. What is Relational Calculus?

https://www.edureka.co/blog/interview-questions/sql-interview-questions