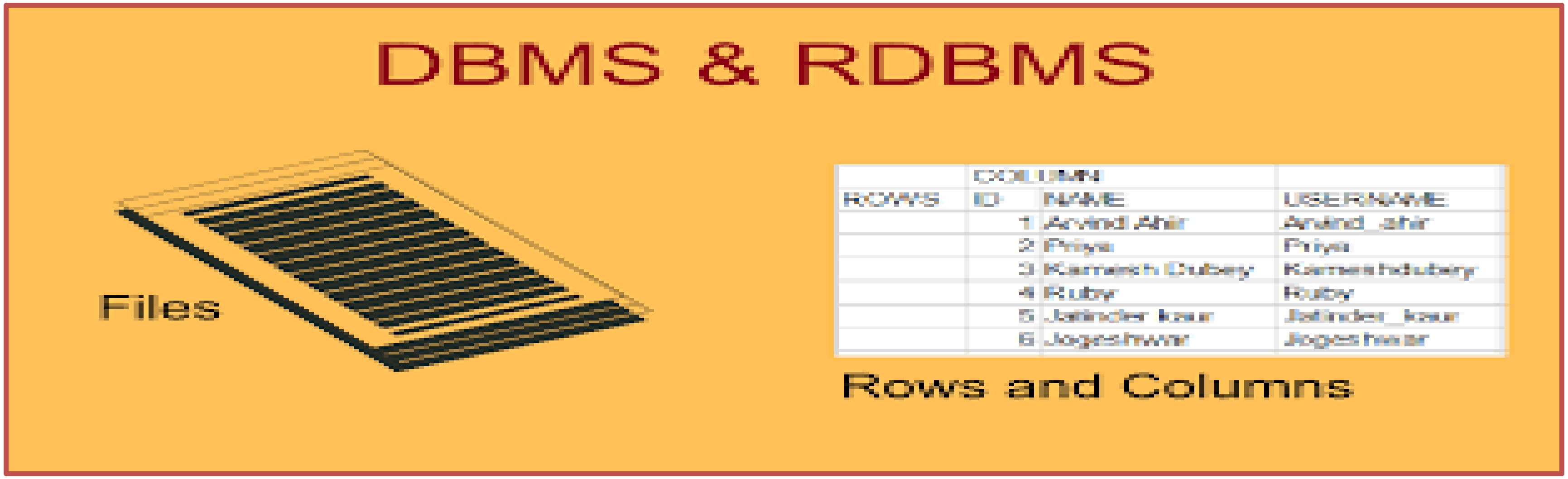
RDBMS

Relational Database Management System

Prof Jayashree M Kudari



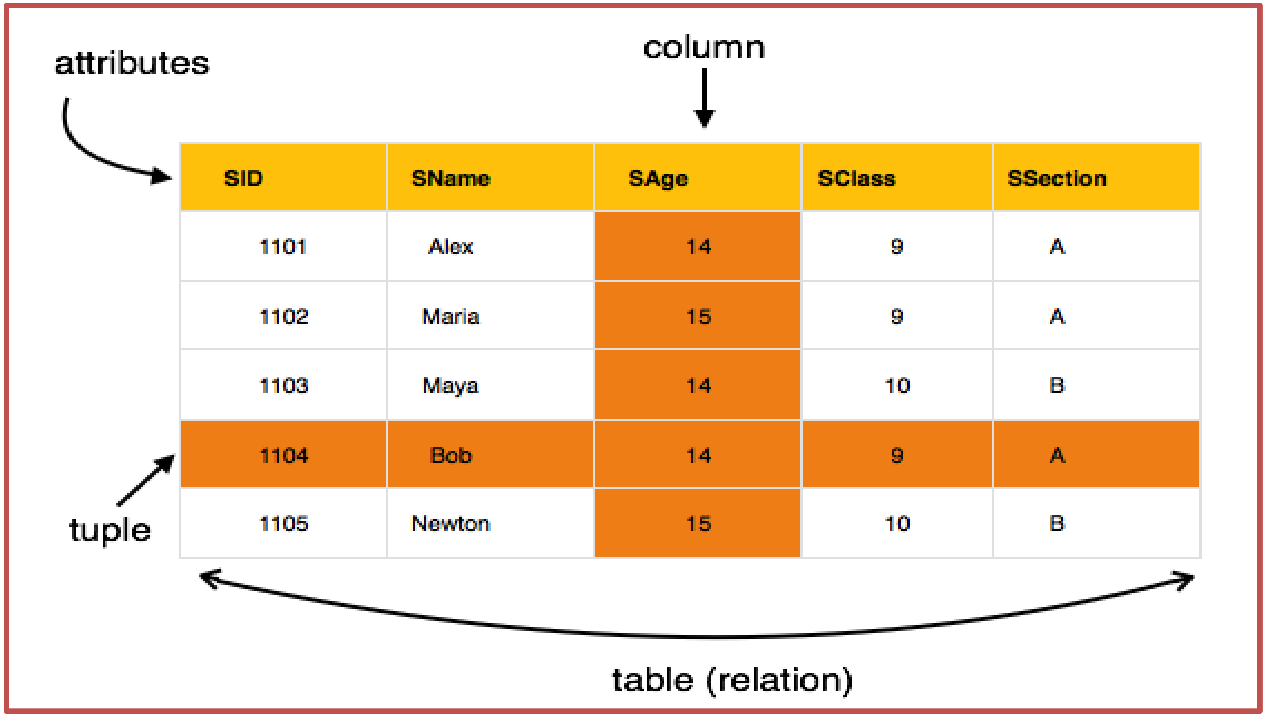
DBMS & RDBMS

	COLUMN		
ROWS	ID	NAME	USERNAME
	1	Arvind Ahir	Arvind_ahir
	2	Priya	Priya
	3	Kamesh Dubey	Kameshdubey
	4	Ruby	Ruby
	5	Jatinder kaur	Jatinder_kaur
	6	Jogeshwar	Jogeshwar

Rows and Columns

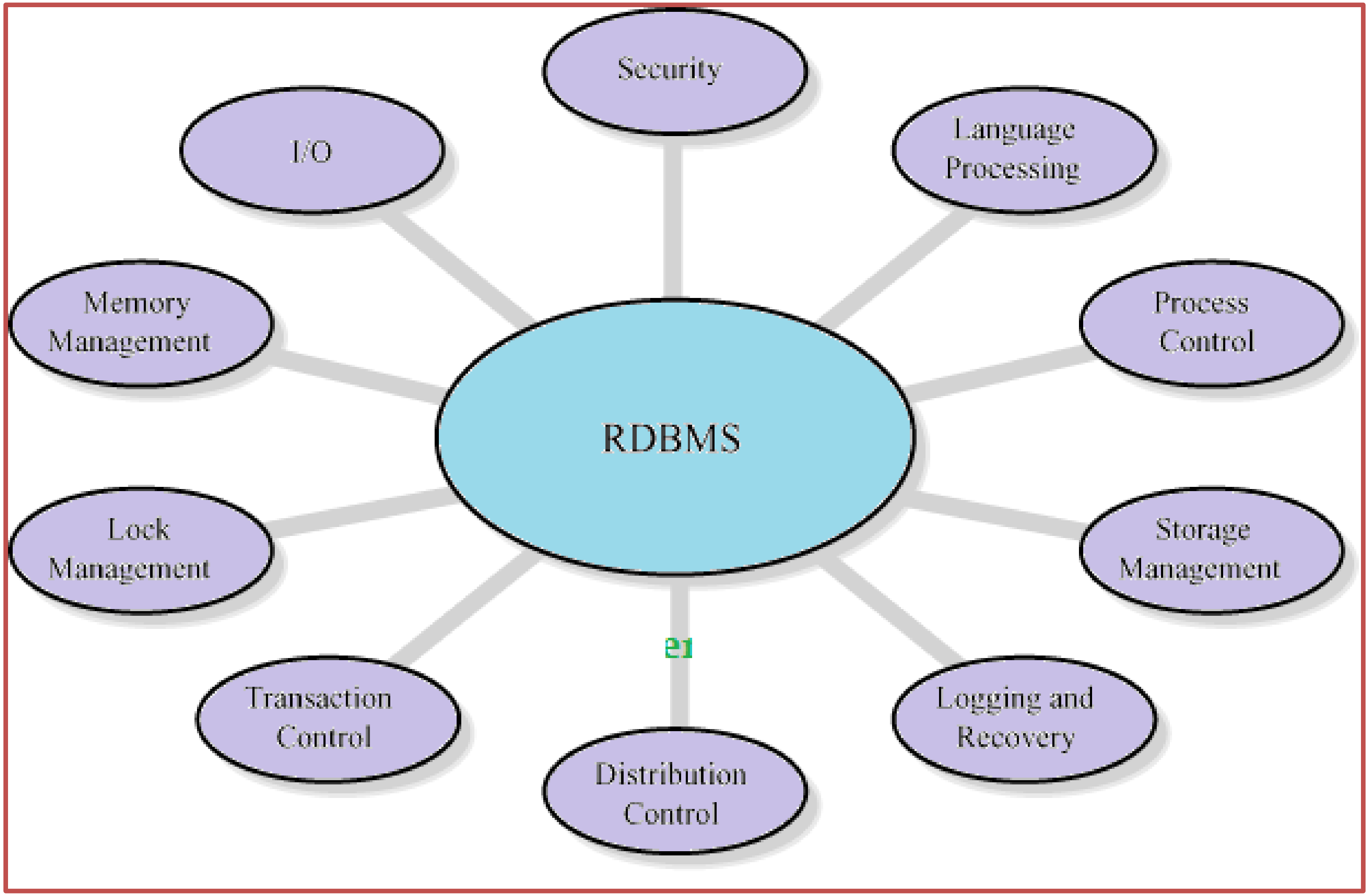
The RDBMS is a general-purpose software system that facilitates the processes of **defining, constructing, manipulating, and sharing** databases among various users and applications.

- A relational database refers to a database that stores data in a structured format, using rows and columns. This makes it easy to locate and access specific values within the database. It is "relational" because the values within each table are related to each other. Tables may also be related to other tables. The relational structure makes it possible to run queries across multiple tables at once.

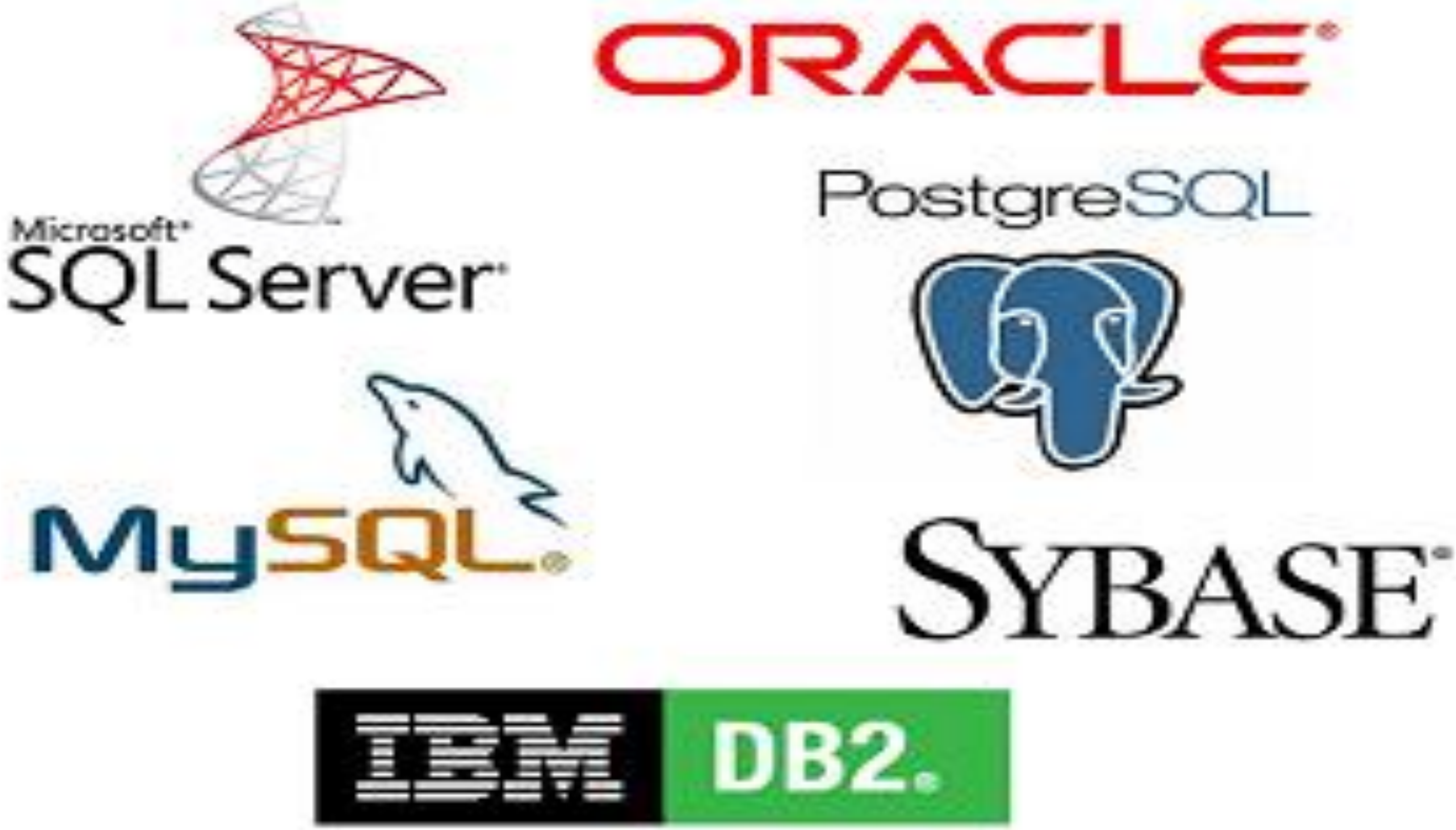


SID	SName	SAge	SClass	SSection
1101	Alex	14	9	A
1102	Maria	15	9	A
1103	Maya	14	10	B
1104	Bob	14	9	A
1105	Newton	15	10	B

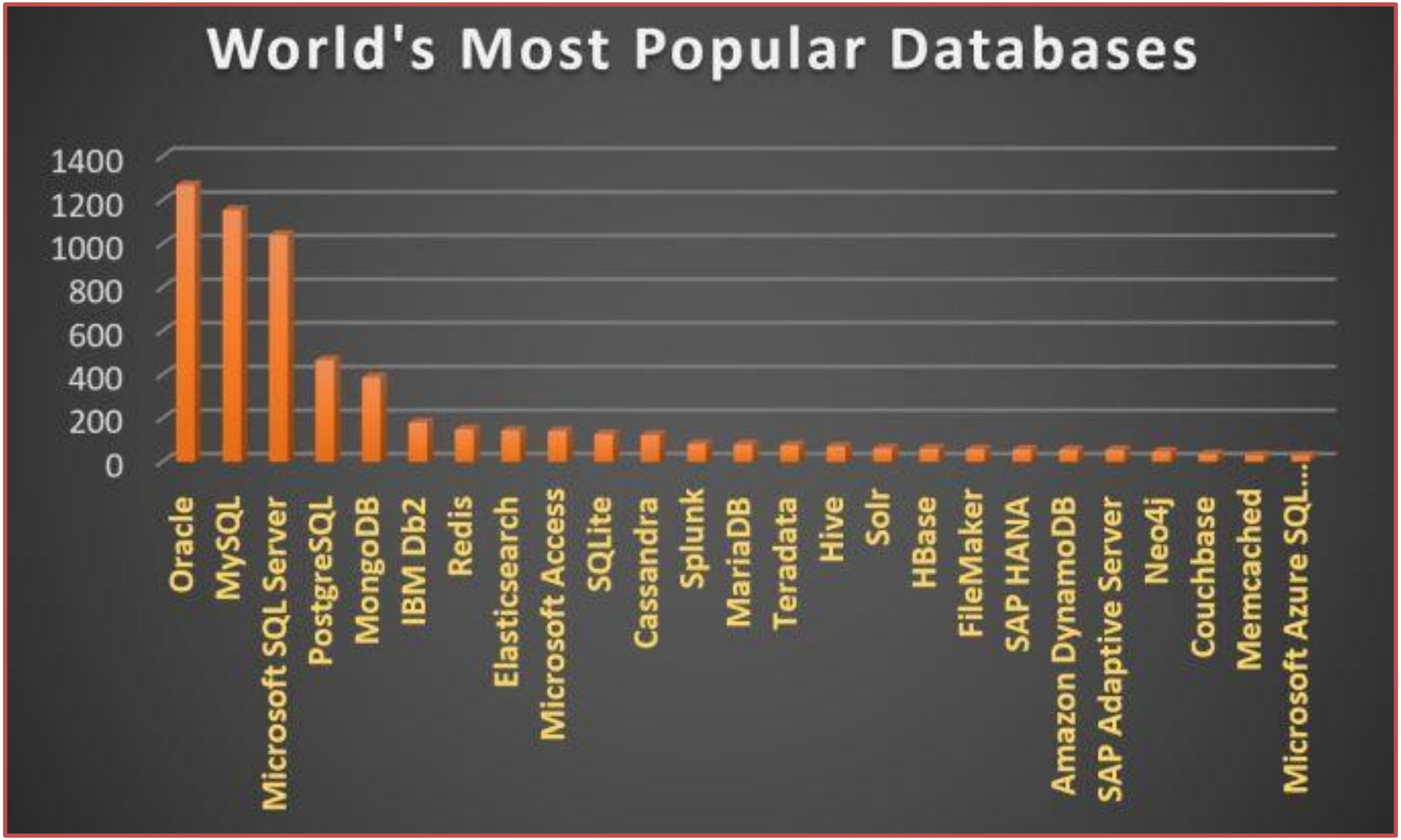
Features of RDBMS



RDBMS TOOLS


Microsoft SQL Server
ORACLE
PostgreSQL
MySQL
SYBASE
IBM DB2

MOST POUPULAR RDBMS TOOLS

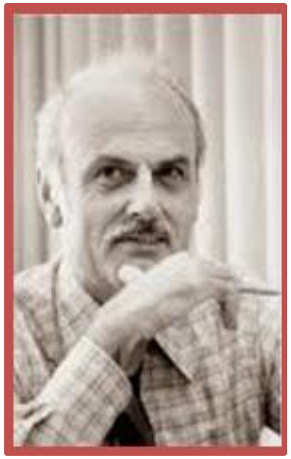


DBMS vs RDBMS

S.NO.	DBMS	RDBMS
1	DBMS stands for **Database Management System**.	RDBMS stands for **Relational Database Management System**.
2	DBMS offers an **organized way** of storing, managing and retrieving information.	RDBMS provides all features of DBMS with an **added referential integrity concept**.
3	In DBMS, data is stored as **a file**.	In RDBMS, data are stored in a **tabular form**.
4	In DBMS, there is **no relationship concept**.	RDBMS is used to build up the **relationship concept** between two database objects, i.e., tables
5	DBMS supports **single user** only.	RDBMS supports **multiple users**.
6	DBMS treats data **as files** internally.	RDBMS treats data **as tables** internally.
7	In DBMS, there is **no security** of data.	In RDBMS, there are **multiple levels of security**, at logging level, at the command level, at the object level.

DBMS vs RDBMS

8	DBMS does **not** support **client-server architecture**.	RDBMS support **client-server architecture**.
9	DBMS does **not** support **distributed architecture**.	RDBMS support **distributed architecture**.
10	DBMS supports **3 rules of E.F.CODD** out of 12 rules.	RDBMS supports minimum **6 rules of E.F.CODD** out of 12 rules.
11	DBMS does **not** impose **integrity constraints**	RDBMS impose the **integrity constraints**.
12	DBMS provides **some uniform methods** to access the stored information.	RDBMS system supports **a tabular structure** of the data and **a relationship** between them to access the stored information.
14	DBMS requires **low** software and hardware requirements.	RDBMS requires **high** software and hardware requirements.
15	Examples of DBMS: **file systems**, **XML,** etc.	Example of RDBMS: **MySQL**, **postgres**, **SQL Server**, **Oracle,** etc.

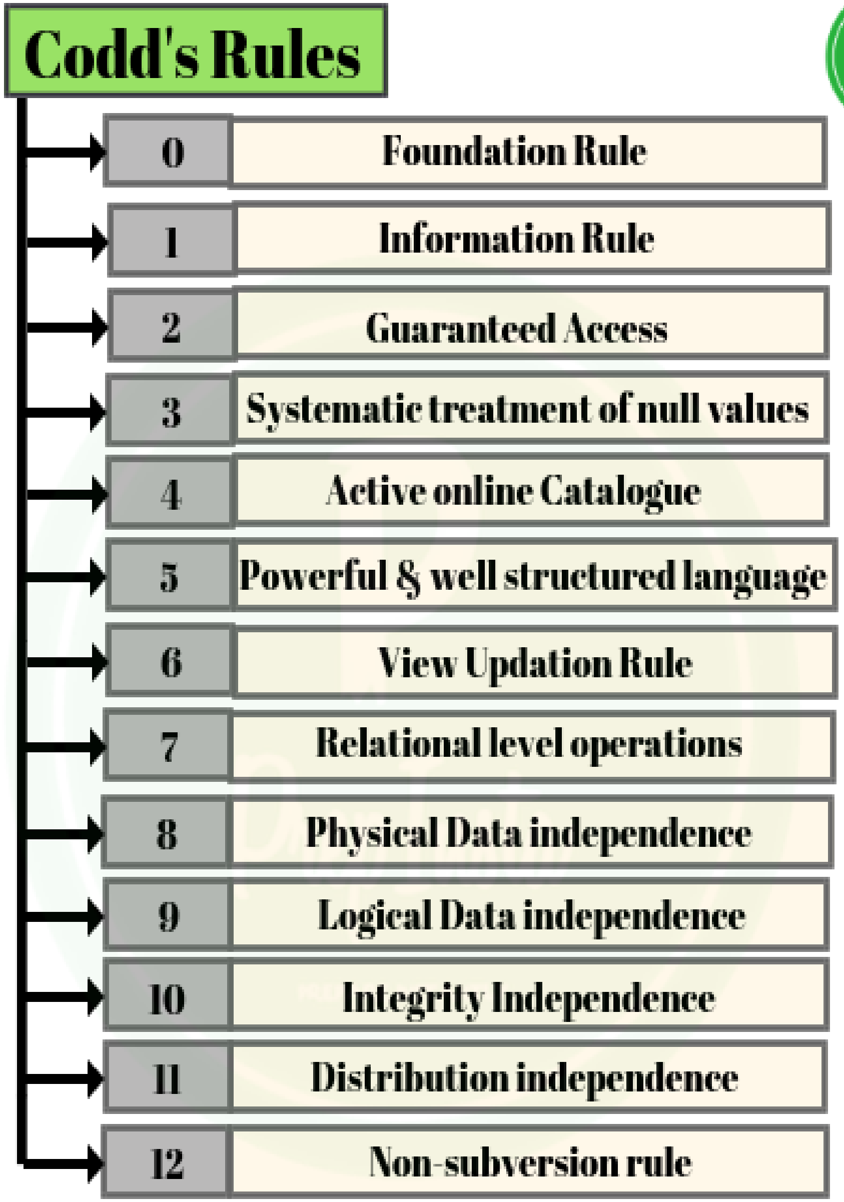


Relational Database Model

12 Rules by

Edgar Frank Ted Codd

[19th August 1923 to 18th April 2003]



Codd's Rules

0	Foundation Rule
1	Information Rule
2	Guaranteed Access
3	Systematic treatment of null values
4	Active online Catalogue
5	Powerful & well structured language
6	View Updation Rule
7	Relational level operations
8	Physical Data independence
9	Logical Data independence
10	Integrity Independence
11	Distribution independence
12	Non-subversion rule

- ✓ **Rule 1: Information rule**

All information(including metadata) is to be represented as stored data in cells of tables.

- ✓ **Rule 2: Guaranteed Access**

Each unique piece of data(atomic value) should be accessible by : Table Name + Primary Key(Row) + Attribute(column).

NOTE: Ability to directly access via POINTER is a violation of this rule.

- ✓ **Rule 3: Systematic treatment of NULL**

Null has several meanings, it can mean missing data, not applicable or no value. It should be handled consistently. Also, Primary key must not be null, ever. Expression on NULL must give null.

- ✓ **Rule 4: Active Online Catalog**

Database dictionary(catalog) is the structure description of the complete Database and it must be stored online. The Catalog must be governed by same rules as rest of the database. The same query language should be used on catalog as used to query database.

- ✓ **Rule 5: Powerful and Well-Structured Language**

Well structured language must be there to provide all manners of access to the data stored in the database. Example: SQL, etc. If the database allows access to the data without the use of this language, then that is a violation.

- ✓ **Rule 6: View Updation Rule**

All the view that are theoretically updatable should be updatable by the system as well.

- ✓ **Rule 7: Relational Level Operation**

There must be Insert, Delete, Update operations at each level of relations. Set operation like Union, Intersection and minus should also be supported.

- ✓ **Rule 8: Physical Data Independence**

The physical storage of data should not matter to the system. If say, some file supporting table is renamed or moved from one disk to another, it should not effect the application.

- ✓ **Rule 9: Logical Data Independence**

If there is change in the logical structure(table structures) of the database the user view of data should not change. Say, if a table is split into two tables, a new view should give result as the join of the two tables. This rule is most difficult to satisfy.

- ✓ **Rule 10: Integrity Independence**

The database should be able to enforce its own integrity rather than using other programs. Key and Check constraints, trigger etc, should be stored in Data Dictionary. This also make RDBMS independent of front-end.

- ✓ **Rule 11: Distribution Independence**

A database should work properly regardless of its distribution across a network. Even if a database is geographically distributed, with data stored in pieces, the end user should get an impression that it is stored at the same place. This lays the foundation of distributed database.

- ✓ **Rule 12: Non subversion Rule**

If a system has an interface that provides access to low-level records, then the interface must not be able to subvert the system and bypass security and integrity constraints.

Purpose of Database Systems

- ✓ Database management systems were developed to handle the following difficulties of typical file-processing systems supported by conventional operating systems.
- ✓ Reduce Data redundancy and inconsistency.
- ✓ Reduce Difficulty in accessing data.
- ✓ Reduce Data isolation – multiple files and formats.
- ✓ Reduce Integrity problems.
- ✓ Concurrent access by multiple users
- ✓ Reduces Security problem.

Advantages of RDBMS

- **Better data sharing**
  - The ability to share the same data resource with multiple applications or users. It implies that the data are stored in one or more servers in the network and that there is some software locking mechanism that prevents the same set of data from being changed by two people at the same time. Data sharing is a primary feature of a database management system (RDBMS).
- **Better data security**
  - A RDBMS provides a framework for better implementation of data policies and data privacy.
- **Improved data integration**
  - Data integration involves combining data from several disparate sources, which are stored using various technologies and provide a unified view of the data.

Advantages of RDBMS

- **Minimized data inconsistency**
  - Data inconsistency is a condition that occurs between files when similar data is kept in different formats in two different files, or when matching of data must be done between files. As a result of the data inconsistency.
  - E.g.: For example, if a hospital is using the traditional file processing, changing the address of a patient will cause changes in several other files because they are kept separately. Inevitably, a case would arise where the address of the patient will be correct in one record and different in another.
- **Improved decision making**
  - Improved and better-managed data makes it possible to generate better quality information which helps in better decision making.
- **Increased end-user productivity**
  - Easy accessibility and availability of data, along with the tools that convert data into usable information, permits end users to make informed decisions. This brings the difference success and failure in the global economy.

Disadvantages

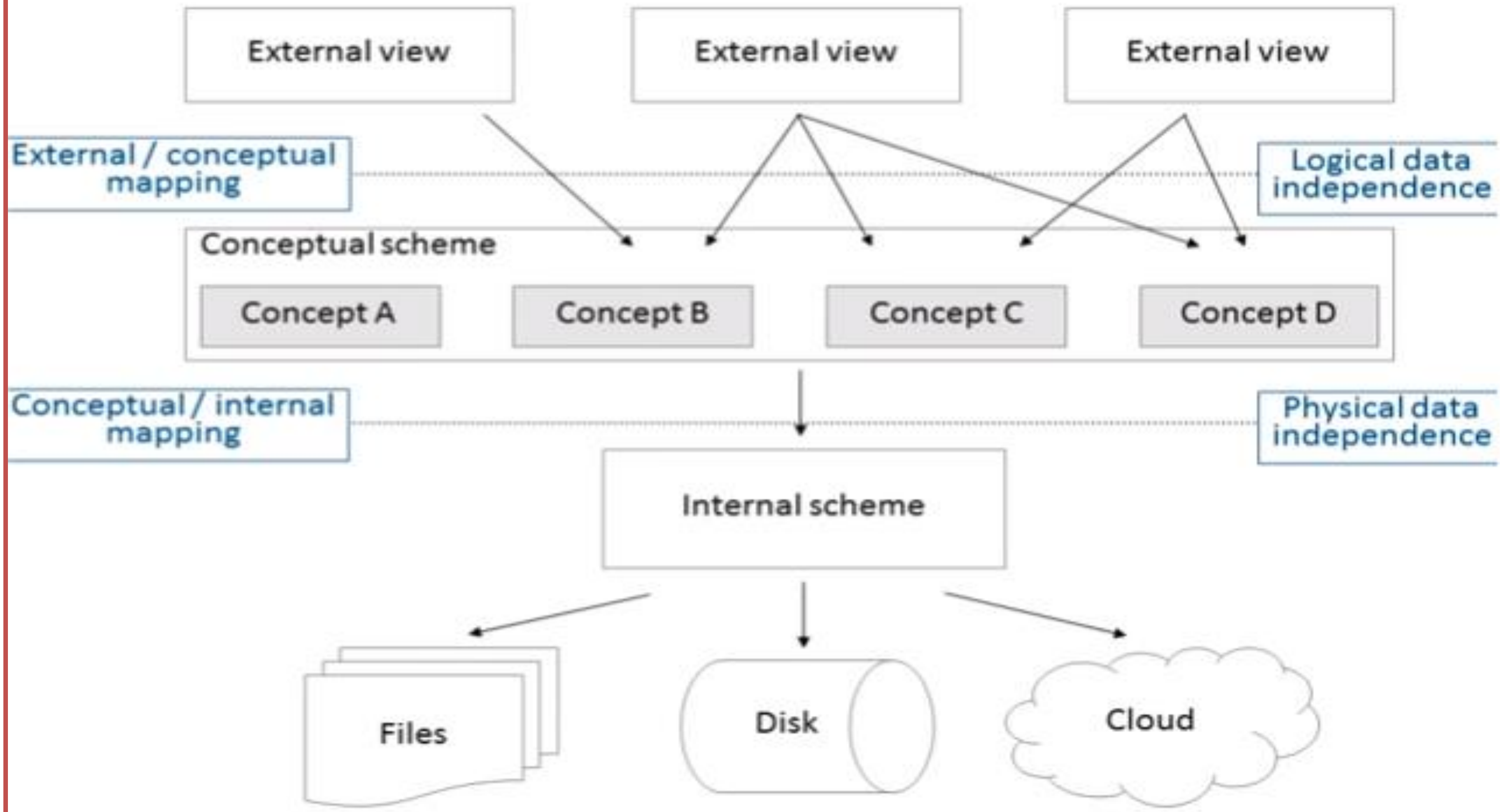
- Increased costs
  - One of the disadvantages of DBMS is database systems require sophisticated hardware, software, and highly skilled person.
  - Training, licensing, and regulation compliance costs are often unheeded when database systems are employed.

- Management intricacy
  - Database systems interface with many different technologies and have a significant impact on a company's resources and culture.
  - The changes introduced by the adoption of a database system must be properly managed to ensure that they help the company's objectives.

- Frequent upgrade/replacement cycles
  - DBMS vendors frequently upgrade their products by adding new functionality. Such new features often come bundled in new upgrade versions of the software. Some of these versions require hardware upgrades. Not only do the upgrades themselves cost money, but it also costs money to train database users and administrators to properly use and manage the new features.

View of Data

The three-schema architecture

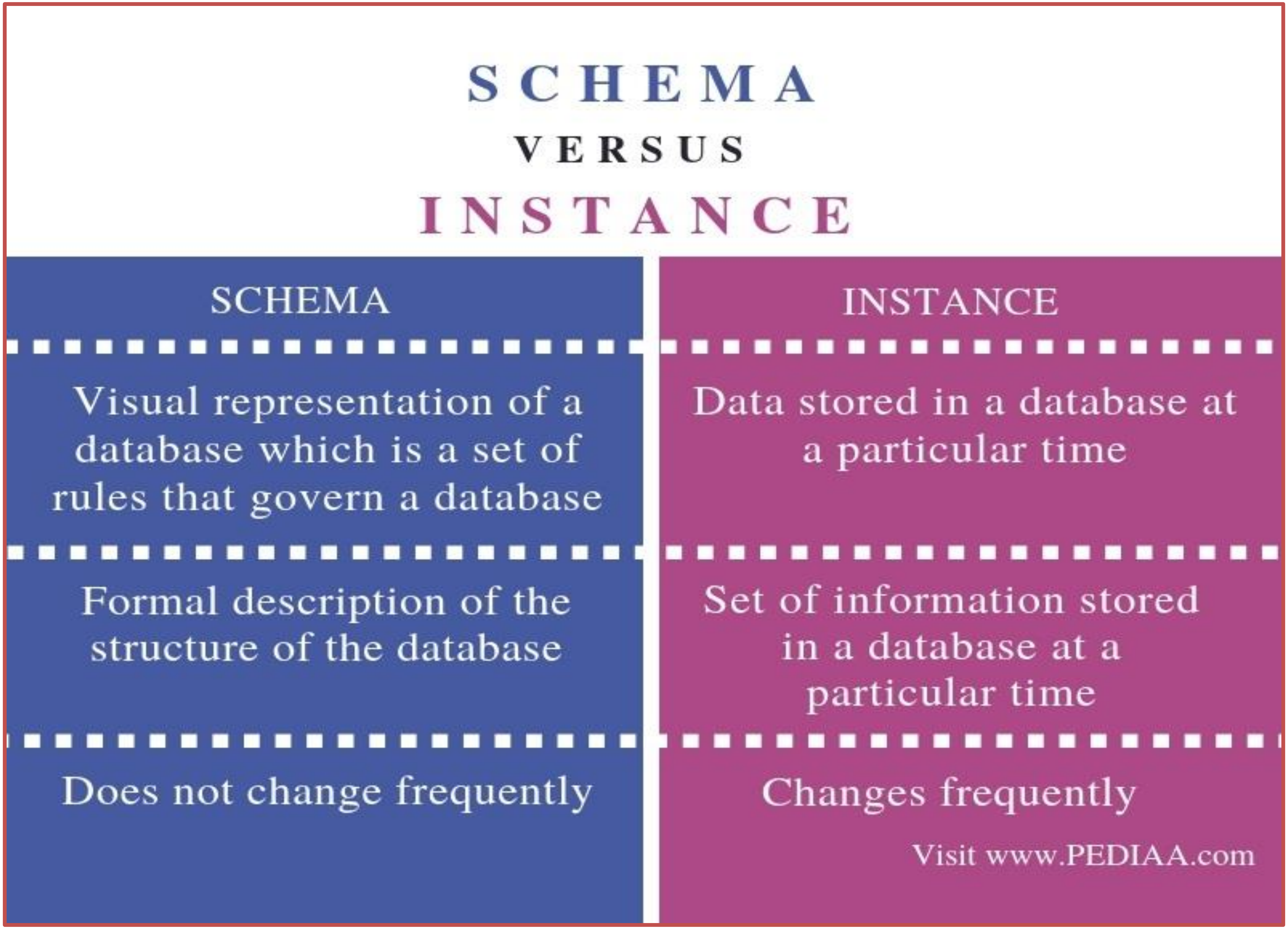


- **View**: This is the highest level of abstraction.
  - Only a part of the actual database is viewed by the users.
  - This level exists to ease the accessibility of the database by an individual user. Users view data in the form of rows and columns.
  - Tables and relations are used to store data.
  - Users can just view the data and interact with the database, storage and implementation details are hidden from them.
- **Logical**:
  - This level comprises of the information that is actually stored in the database in the form of tables.
  - It also stores the relationship among the data entities in relatively simple structures.
- **Physical**:
  - This is the lowest level of data abstraction. It tells us how the data is actually stored in memory.
  - The access methods like sequential or random access and file organization methods like B+ trees, hashing used for the same. Suppose we need to store the details of an employee. Blocks of storage and the amount of memory used for these purposes is kept hidden from the user.

- Similar to types and variables in programming languages
- **Schema –** the logical structure of the database
- e.g., the database consists of information about a set of customers and

accounts and the relationship between them)

- Analogous to type information of a variable in a program
  - **Physical schema:** database design at the physical level
  - **Logical schema:** database design at the logical level
- **Instance –** The data stored in database at a particular moment of time is called instance of database.

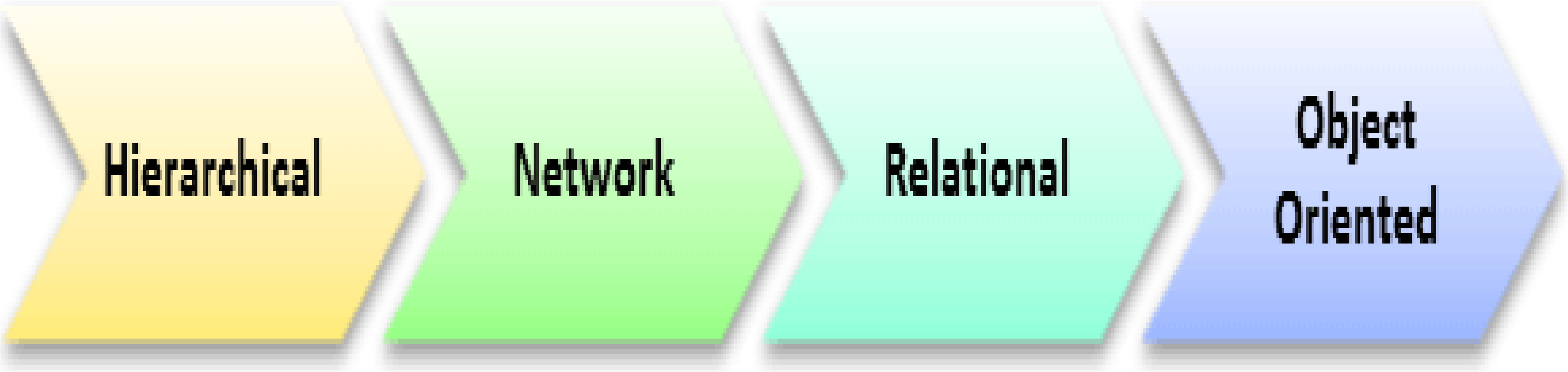


SCHEMA VERSUS INSTANCE

SCHEMA	INSTANCE
Visual representation of a database which is a set of rules that govern a database	Data stored in a database at a particular time
Formal description of the structure of the database	Set of information stored in a database at a particular time
Does not change frequently	Changes frequently

Visit www.PEDIAA.com

DATABASE MODELS

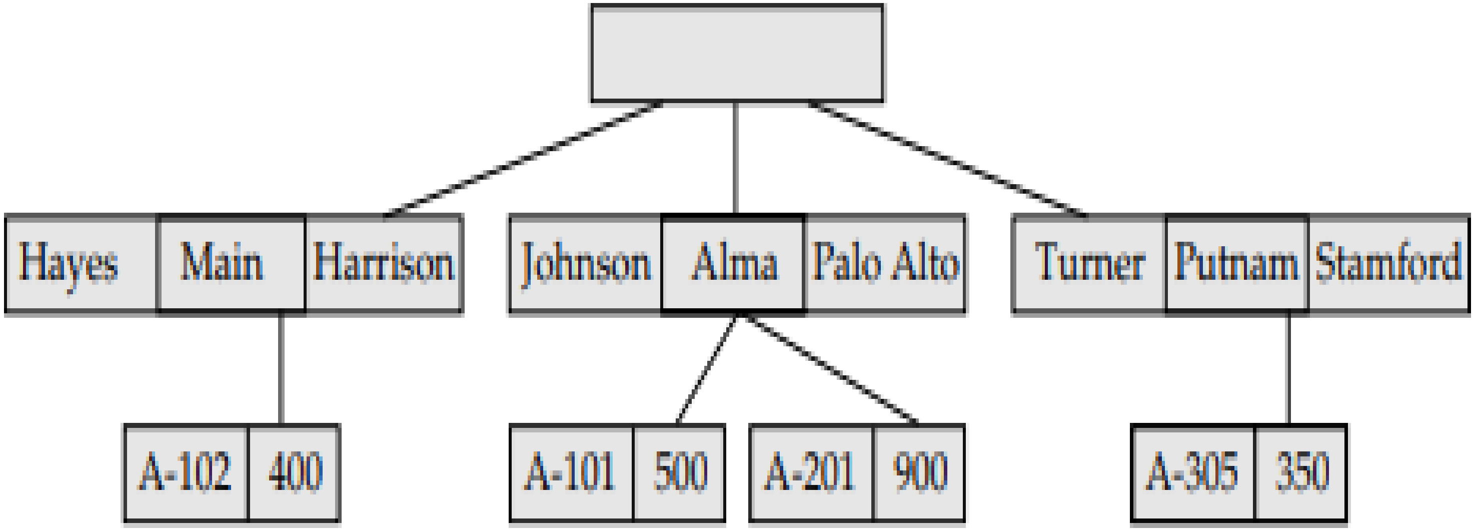


- Record-based logical models
  - Relational Model
  - Network Model
  - Hierarchical Model

- Object-Oriented Model
- Entity-Relationship Diagram

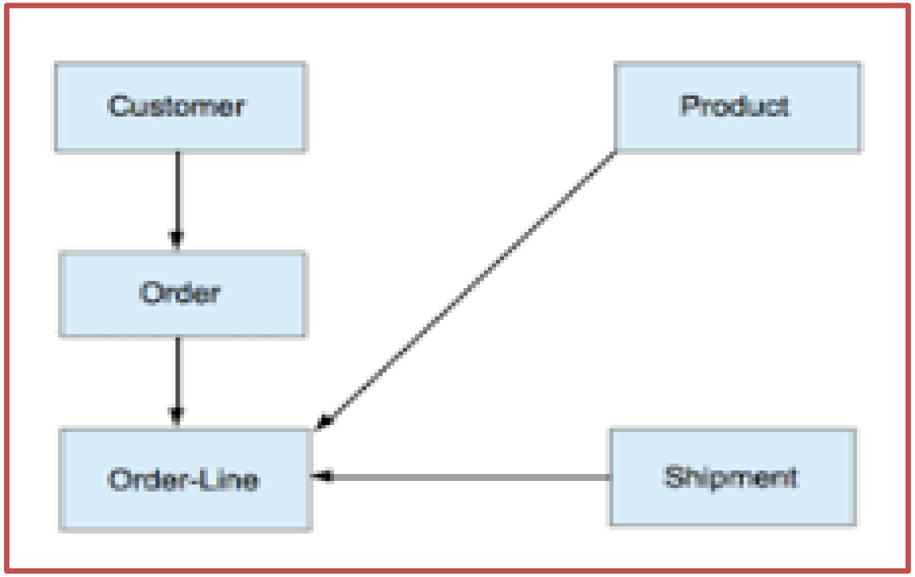
Hierarchical Database **Model**

- A hierarchical database consists of a collection of records that are connected to each other through links. A record is similar to a record in the network model. Each record is a collection of fields (attributes), each of which contains only one data value. A link is an association between precisely two records. Thus, a link here is similar to a link in the network model.



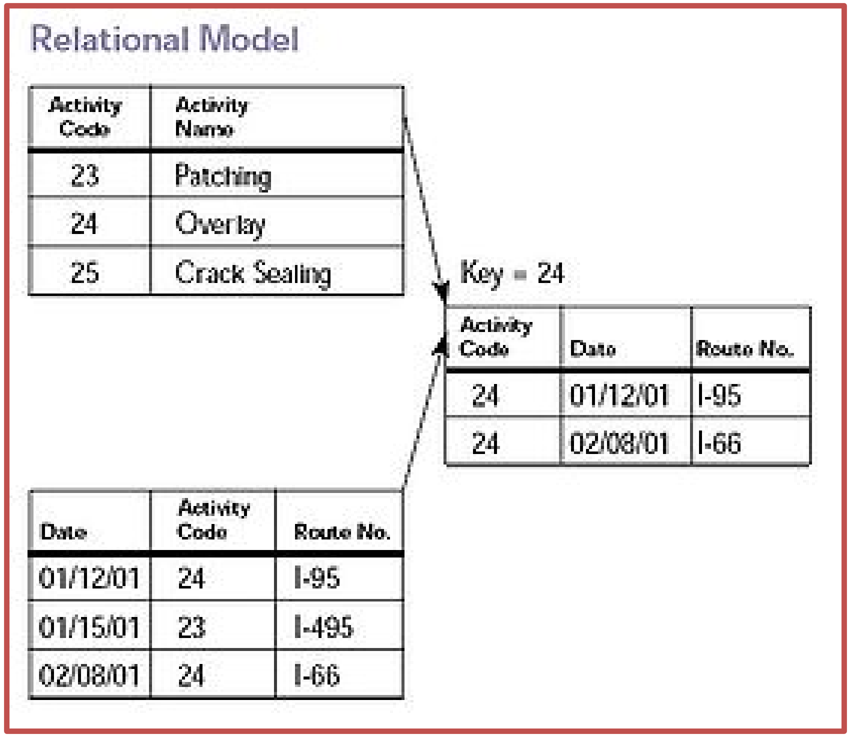
Network Model

- The **network model** is a database **model** considered as a flexible way of representing objects and their relationships. Its distinguishing feature is that the schema, viewed as a graph in which object types are nodes and relationship types are arcs, is not restricted to being a hierarchy or lattice.



Relational Model

- The most common model, the relational model sorts data into tables, also known as relations, each of which consists of columns and rows.
- Each column lists an attribute of the entity in question, such as price, zip code, or birth date. Together, the attributes in a relation are called a domain.
- Each row, also called a tuple, includes data about a specific instance of the entity. The model also accounts for the types of relationships between those tables, including one-to-one, one-to-many, and many-to-many relationships. Here's an example:



Relational Model

Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

Key = 24

Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495
02/08/01	24	I-66

Object-oriented database model

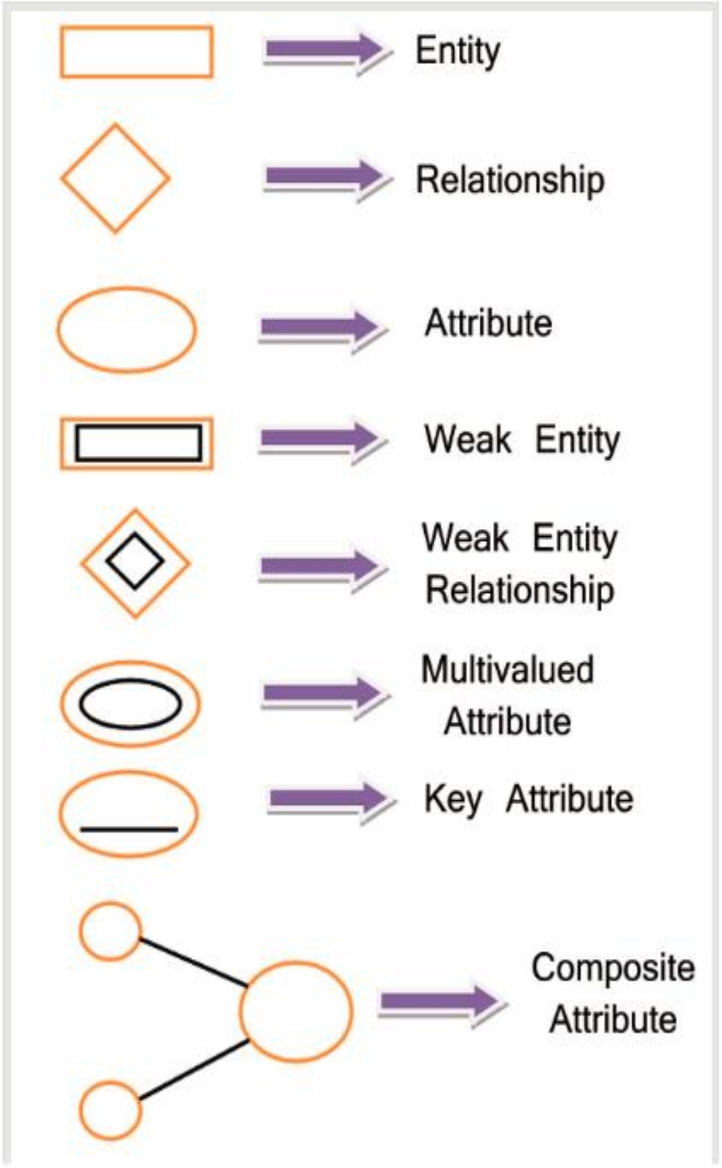
- This model defines a database as a collection of objects, or reusable software elements, with associated features and methods. There are several kinds of object-oriented databases:
- A **multimedia database** incorporates media, such as images, that could not be stored in a relational database.
- A **hypertext database** allows any object to link to any other object. It’s useful for organizing lots of disparate data, but it’s not ideal for numerical analysis.
- The object-oriented database model is the best known post-relational database model, since it incorporates tables, but isn’t limited to tables. Such models are also known as hybrid database models.

Entity –Relationship Diagrams in DBMS: Components, Symbols, And Notations

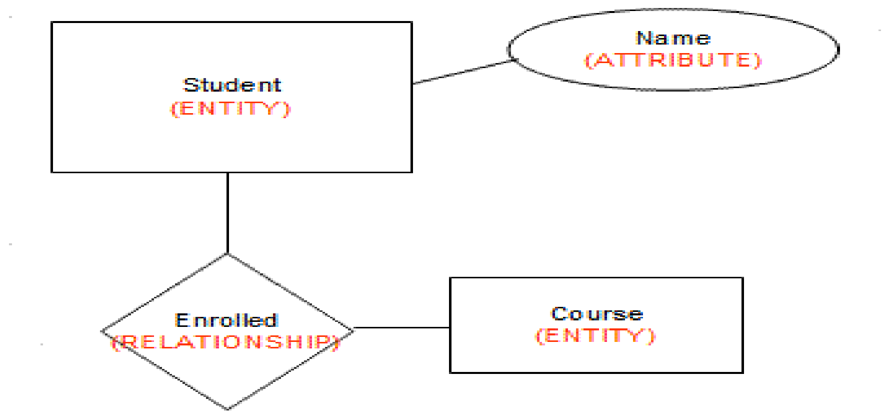
- E-R diagrams are used to represent E-R model in a database, which makes them easy to be converted into relations (tables).
- E-R diagrams require no technical knowledge & no hardware support.
- These diagrams are very easy to understand and easy to create .
- It gives a standard solution of visualizing the data logically.
- **Entity**
- **What are Attributes? And Types of Attributes.**
- **Keys**
- **Relationships**

Symbols:

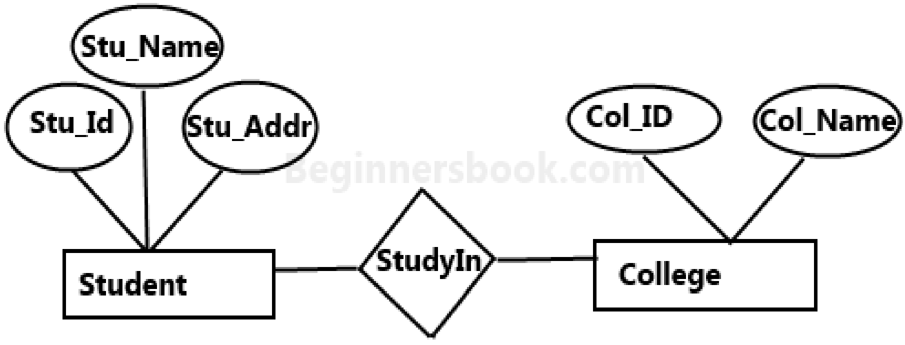
ER Diagrams Symbols, And Notations



EXAMPLE-1



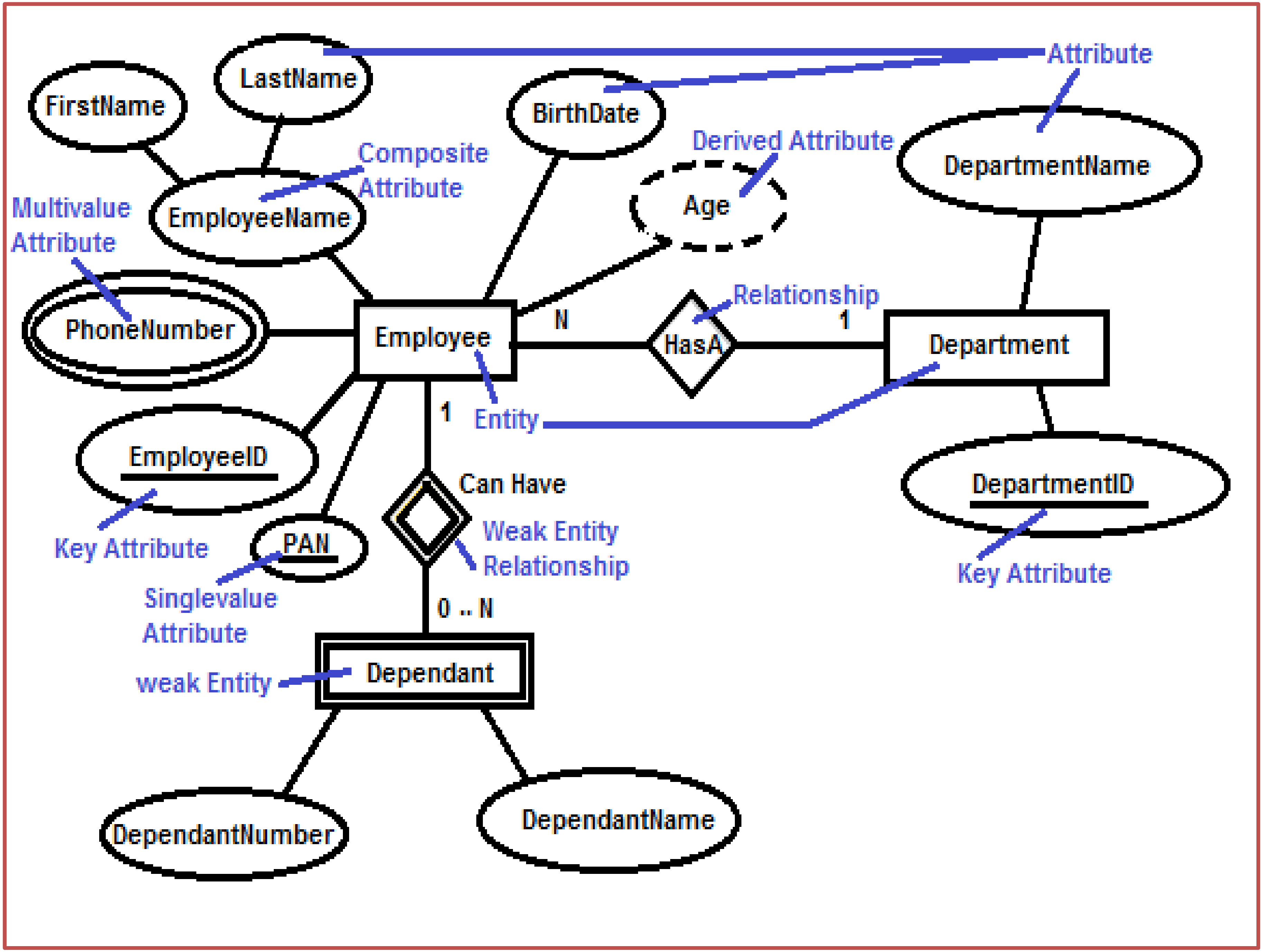
EXAMPLE-2



Sample E-R Diagram

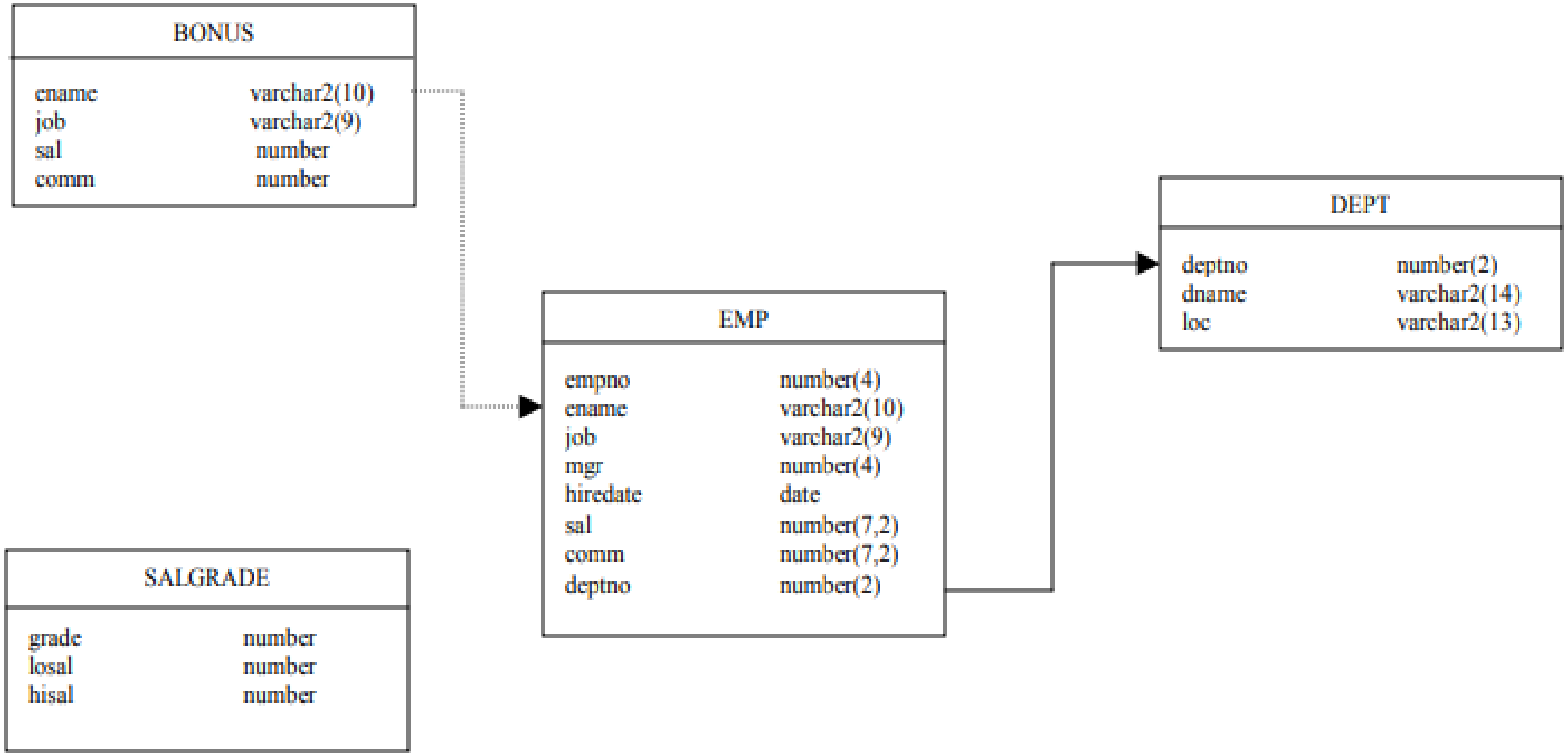
Types of Attributes

- **Simple attribute** − Simple attributes are atomic values, which cannot be divided further. For example, a student's phone number is an atomic value of 10 digits.
- **Composite attribute** − Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first_name and last_name.
- **Derived attribute** − Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average_salary in a department should not be saved directly in the database, instead it can be derived. For another example, age can be derived from data_of_birth.
- **Single-value attribute** − Single-value attributes contain single value. For example − Social_Security_Number.
- **Multi-value attribute** − Multi-value attributes may contain more than one values. For example, a person can have more than one phone number, email_address, etc.


Attribute
LastName
FirstName
BirthDate
Derived Attribute
DepartmentName
Composite
Attribute
EmployeeName
Age
Multivalue
Attribute
Relationship
PhoneNumber
Employee
N
HasA
1
Department
1
Entity
EmployeeID
DepartmentID
Can Have
Weak Entity
Relationship
Key Attribute
PAN
Key Attribute
Singlevalue
Attribute
0 .. N
weak Entity
Dependant
DependantNumber
DependantName

Data Masker for Oracle
Scott Sample Schema - Entity Relationship Diagram

An entity relationship diagram for the tables in the Scott sample schema which is supplied with Oracle database.



BONUS	
ename	varchar2(10)
job	varchar2(9)
sal	number
comm	number

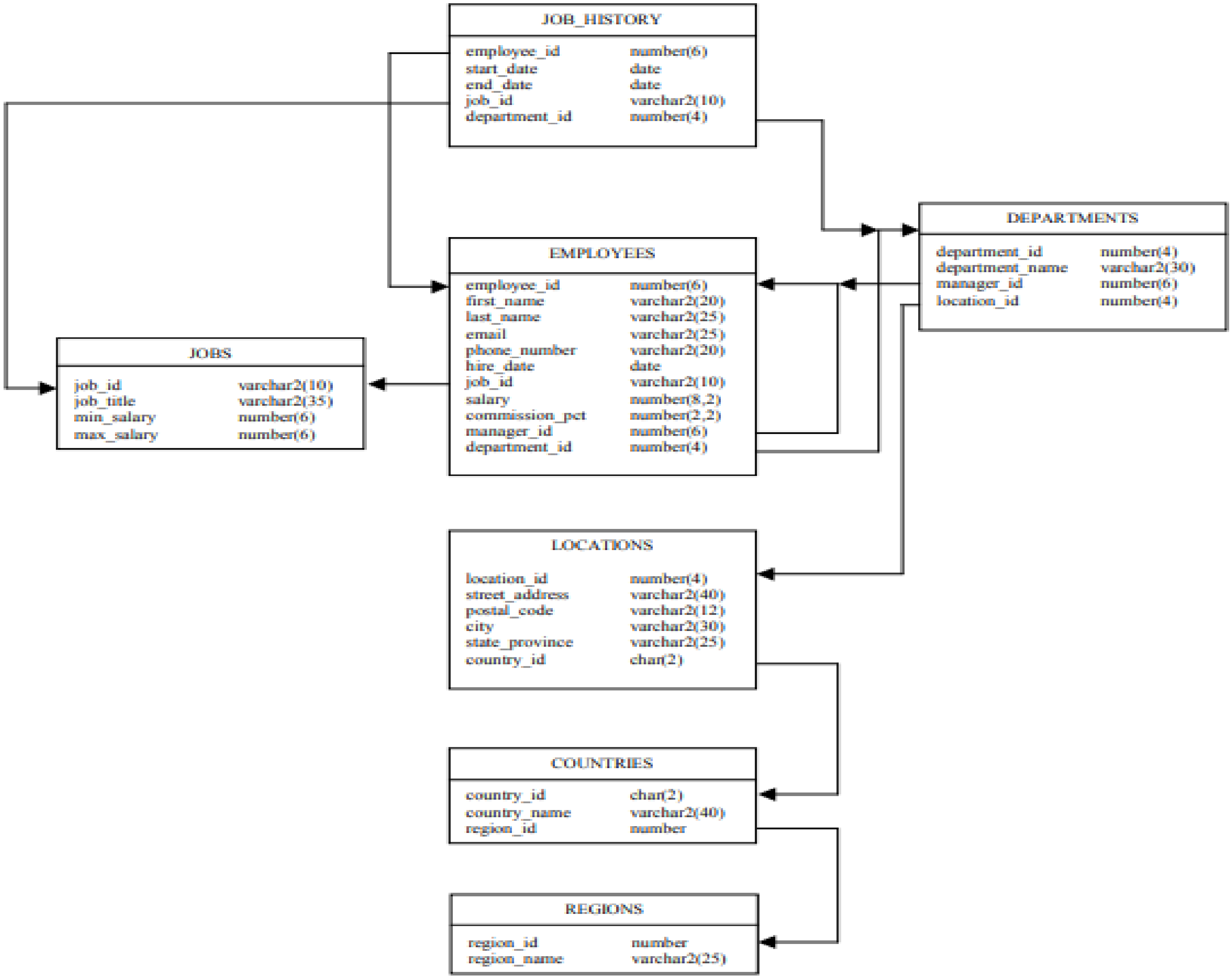
EMP	
empno	number(4)
ename	varchar2(10)
job	varchar2(9)
mgr	number(4)
hiredate	date
sal	number(7,2)
comm	number(7,2)
deptno	number(2)

DEPT	
deptno	number(2)
dname	varchar2(14)
loc	varchar2(13)

SALGRADE	
grade	number
losal	number
hisal	number

Identify key attribute and draw ER-Diagram for the following **Movie Database**

1. Movie{#MID, mname, typeofmovie, duration, budget, year, date}
2. DIRECTOR{#DID,MID,fname,lname,dob, place}
3. MUSICDIRECTOR{#MUID, MIDfname,lname,dob,place}
4. ACTOR{#AID,MID,fname,lname,dob,place}
5. PRODUCE{#PID,MID,fname,lname,place}



JOB_HISTORY

employee_id	number(6)
start_date	date
end_date	date
job_id	varchar2(10)
department_id	number(4)

DEPARTMENTS

department_id	number(4)
department_name	varchar2(30)
manager_id	number(6)
location_id	number(4)

EMPLOYEES

employee_id	number(6)
first_name	varchar2(20)
last_name	varchar2(25)
email	varchar2(25)
phone_number	varchar2(20)
hire_date	date
job_id	varchar2(10)
salary	number(8,2)
commission_pct	number(2,2)
manager_id	number(6)
department_id	number(4)

JOBS

job_id	varchar2(10)
job_title	varchar2(35)
min_salary	number(6)
max_salary	number(6)

LOCATIONS

location_id	number(4)
street_address	varchar2(40)
postal_code	varchar2(12)
city	varchar2(30)
state_province	varchar2(25)
country_id	char(2)

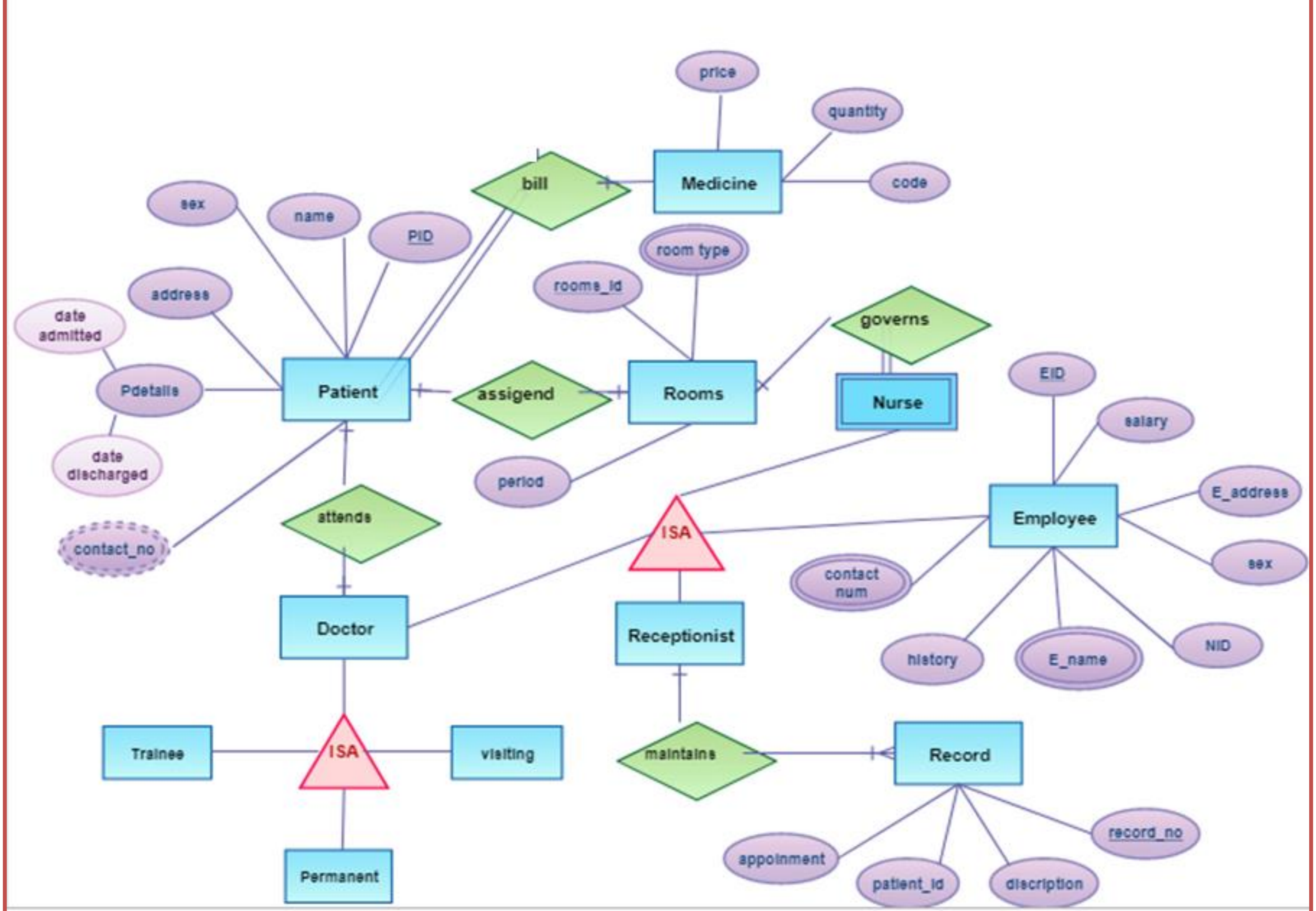
COUNTRIES

country_id	char(2)
country_name	varchar2(40)
region_id	number

REGIONS

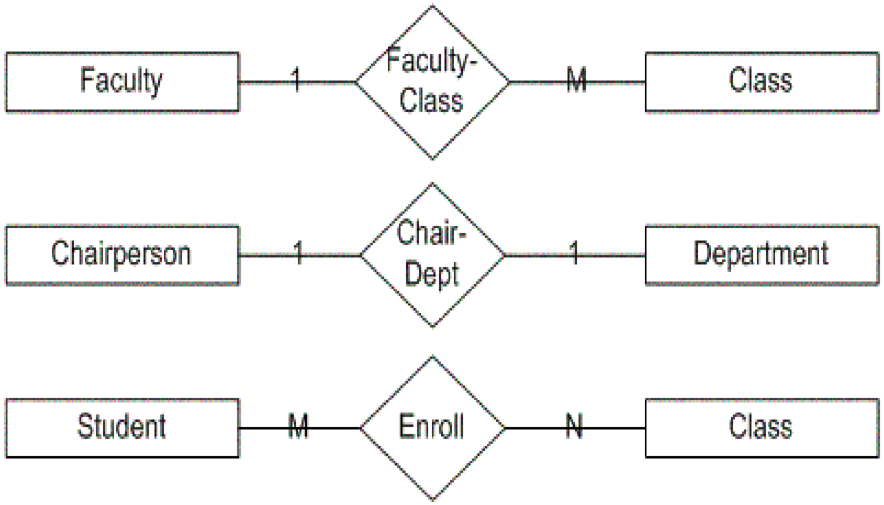
region_id	number
region_name	varchar2(25)

E-R Diagram for Hospital Management System

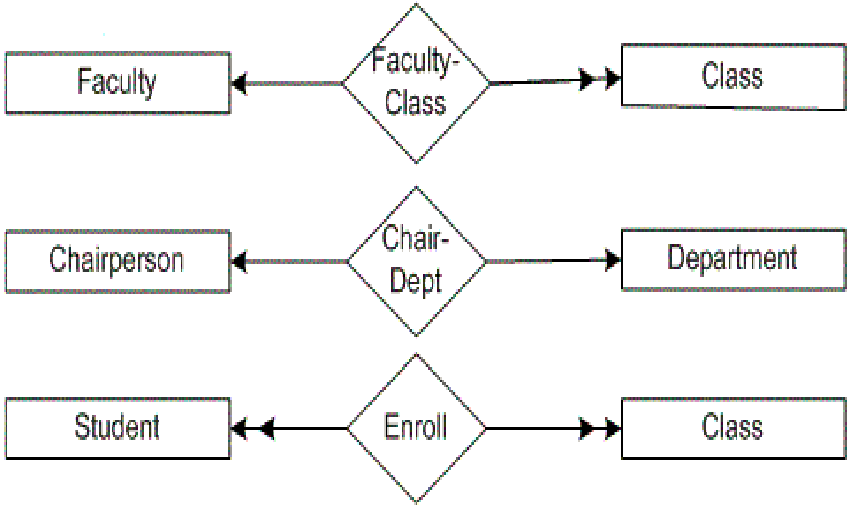


Cardinality of Relationships

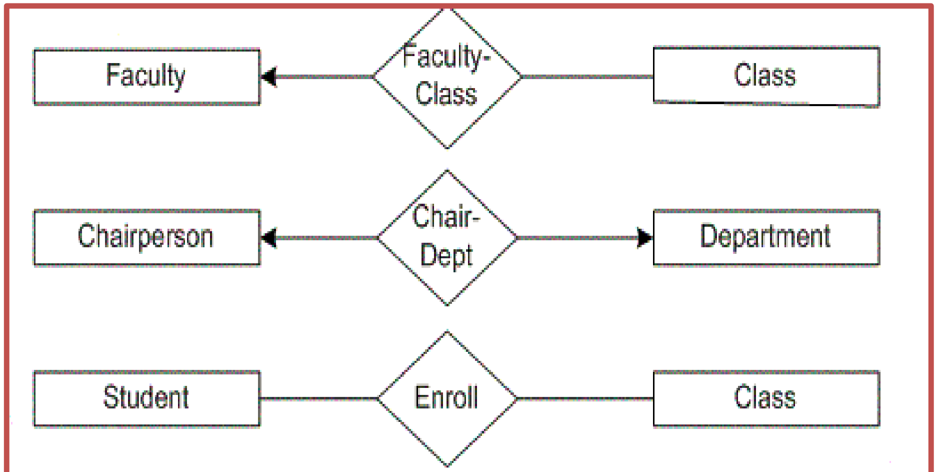
- Cardinality is the number of entity instances to which another entity set can map under the relationship. This does not reflect a requirement that an entity has to participate in a relationship. Participation is another concept.
- **One-to-one: X-Y is 1:1** when each entity in X is associated with at most one entity in Y, and each entity in Y is associated with at most one entity in X.
- **One-to-many: X-Y is 1:M** when each entity in X can be associated with many entities in Y, but each entity in Y is associated with at most one entity in X.
- **Many-to-many: X:Y is M:M** if each entity in X can be associated with many entities in Y, and each entity in Y is associated with many entities in X ("many" =>one or more and sometimes zero)
- Of these choices, please use the first method!



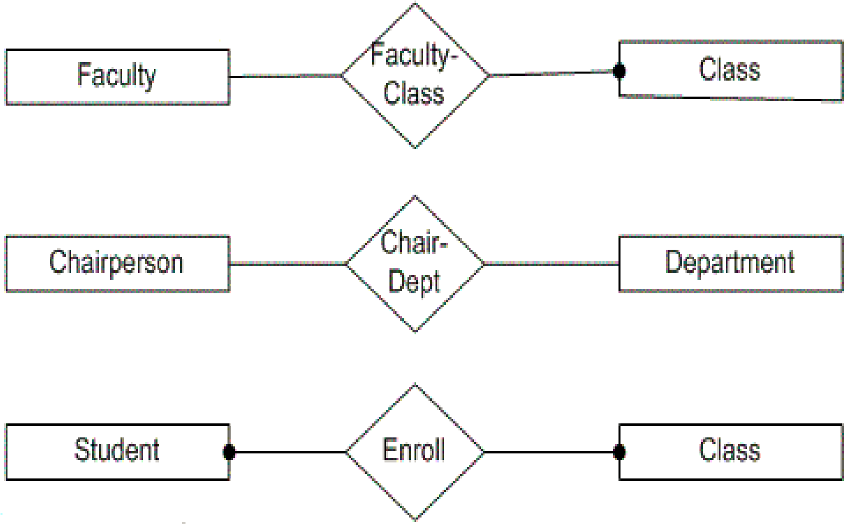
Method 1: One: 1
Many M,N



Method 2: One: single arrow
Many: double arrow



Method 3: One: single arrow
Many: no arrow



Method 4: One: no arrow
Many: big dot

Different Types of Database Users in DBMS

- **Application Programmers**
- As its name shows, application programmers are the one who writes application programs that uses the database. These application programs are written in programming languages like COBOL or PL (Programming Language 1), Java and fourth generation language. These programs meet the user requirement and made according to user requirements. Retrieving information, creating new information and changing existing information is done by these application programs.

- **End Users**
- End users are those who access the database from the terminal end. They use the developed applications and they don't have any knowledge about the design and working of database. These are the second class of users and their main motto is just to get their task done. There are basically two types of end users that are discussed below.
- **SQL Programmer**
- These users have great knowledge of query language. Casual users access data by entering different queries from the terminal end. They do not write programs but they can interact with the system by writing queries.
- DataBase Administrators

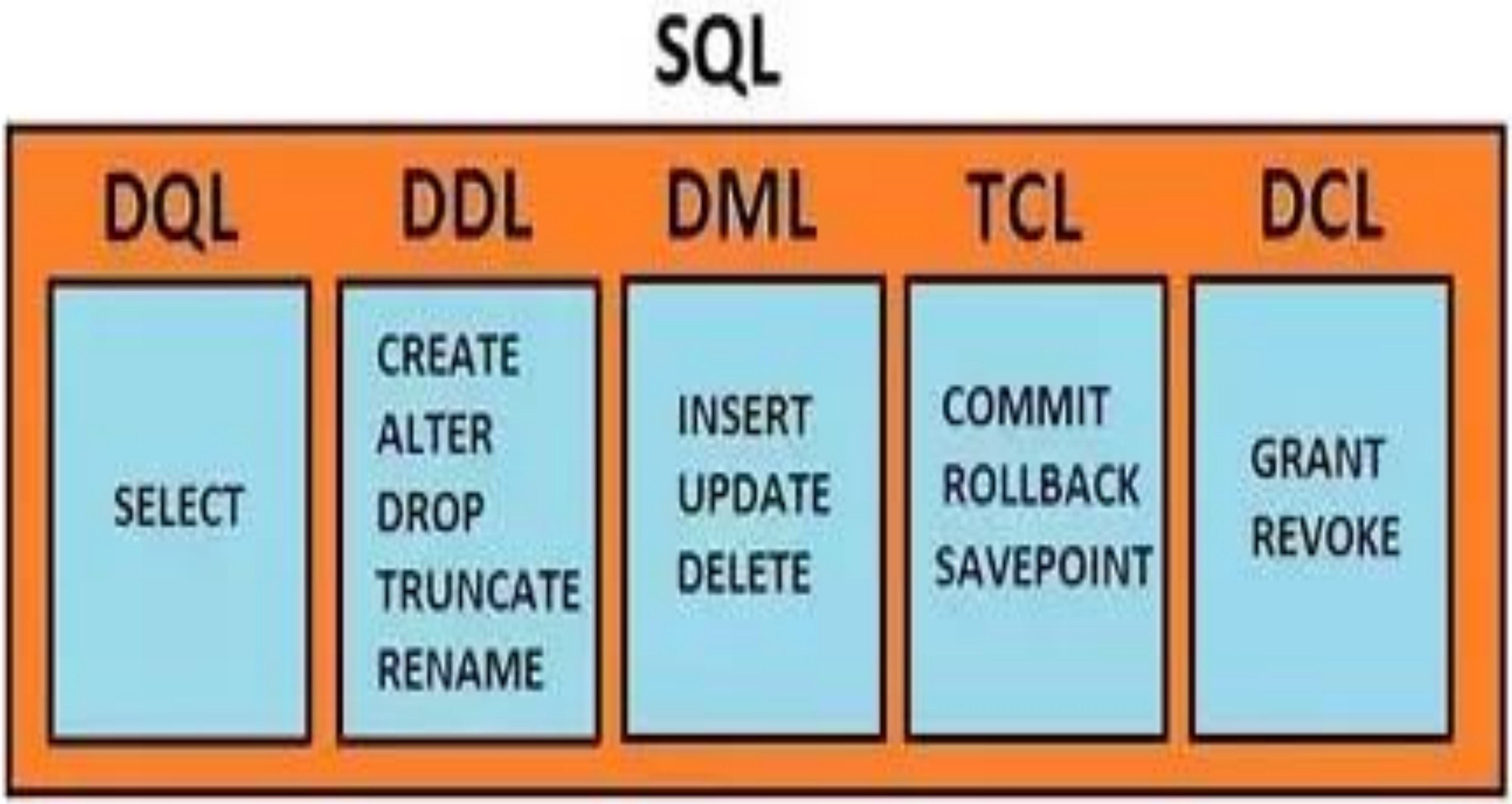
DBA (Database Administrator)

- DBA can be a single person or it can be a group of person. Database Administrator is responsible for everything that is related to database. He makes the policies, strategies and provides technical supports.
- **System Analyst(DB Architect)**
- System analyst is responsible for the design, structure and properties of database. All the requirements of the end users are handled by system analyst. Feasibility, economic and technical aspects of DBMS is the main concern of system analyst.

Data Base Administrator

- A database administrator (DBA) directs or performs all activities related to maintaining a successful database environment. Responsibilities include designing, implementing, and maintaining the database system; establishing policies and procedures pertaining to the management, security, maintenance, and use of the database management system.

Database Languages



SQL

DQL	DDL	DML	TCL	DCL

SELECT	CREATE
	ALTER
	DROP
	TRUNCATE
	RENAME
		INSERT
		UPDATE
		DELETE
			COMMIT
			ROLLBACK
			SAVEPOINT
				GRANT
				REVOKE

1. **DRL (Data Retrieval Language)**

   This is to retrieve data from ORACLE/sql server. SELECT statement falls in this category.

2. **DDL (Data Definition Language)**

   These statements are used to create tables and databases and define field properties or table properties. Examples of commands that fall in this category are CREATE,ALTER and DROP statements

3. **DML (Data Manipulation Language)**

   The statements that falls under this category are used to update data or add or remove data from tables. UPDATE, DELETE and INSERT commands fall under this category.

4. **DCL (Data Control Language)**

   It is used to control who access the data. The commands that come under this category are GRANT and REVOKE

5. **TCL (Transaction Control Language)**

   This language is used to commit data and restore data. COMMIT and ROLLBACK falls under this category.

Questions to practice

- Write advantages and disadvantages of DBMS.
- Define and explain the purpose of database system
- Discuss any five types of notations used in E-R model.
- Explain different views of data abstractions in RDBMS.
- Discuss about any five types of database users.
- Explain any five roles of database administrator.
- Mention any two types of database languages and write commands used in each language.
- Compare file system and RDBMS.
- Write a note on any two data models.
- Differentiate between logical view and physical view.
- Construct an ER diagram for library manage
- ment system
- Construct an ER diagram for theater ticket booking system.

- What is schema?
- What is instance?
- List Sub Languages of SQL.
- What is database management system?