Database Management System

UNIT 1
INTRODUCTION

Today's Target

- Database
- DBMS
- File system VS DBMS
- AKTU PYQs

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

DATA

- Data is statically raw and unprocessed information.

DATABASE

- A database is an organized collection of data that is stored and managed on a computer system.
- It allows users to easily store, retrieve, and update information.

Example library database.

Database Management System (DBMS)

- It is a software that is designed to manage and organize data in a structured manner.
- It allows users to create, modify, and query a database, as well as manage the security and access controls for that database.
- DBMS provides an environment to store and retrieve data in convenient and efficient manner.

Key features of DBMS

- A DBMS is responsible for storing and retrieving data from the database, and can provide various methods for searching and querying the data.
- A DBMS provides mechanisms for controlling concurrent access to the database, to ensure that multiple users can access the data without conflicting with each other.
- A DBMS provides mechanisms for backing up and recovering the data in the event of a system failure.

Database Administrator (DBA)

- It is an individual or person responsible for controlling, maintaining, coordinating, and operating a database management system.
- Managing, securing, and taking care of the database systems is a prime responsibility.

Types of Database Administrator (DBA)

1.Administrative DBA: Their job is to maintain the server and keep it functional. They are concerned with data backups, security, troubleshooting, replication, migration, etc.

2.Cloud DBA: Nowadays companies prefer to save their work piece on cloud storage. As it reduces the chance of data loss and provides an extra layer of data security and integrity

3.Application DBA: They particularly manage all requirements of application components that interact with the database and accomplish activities such as application installation and coordination, application upgrades, database cloning, data load process management, etc

Role and Duties of Database Administrator (AKTU 2017-18)

1. They decide on economical hardware, based on cost, performance, and efficiency of hardware, and best suits the organization.
2. Data integrity needs to be checked and managed accurately as it protects and restricts data from unauthorized use.
3. Database Administrator is solely responsible for giving permission to access data available in the database.

4. To ensure data is not lost in case of failures (e.g., system crashes, data corruption), DBAs create regular backup strategies.

5. DBAs monitor database performance using metrics like query execution time, response time, and disk usage.

6. DBAs manage user accounts and roles, ensuring each user has appropriate permissions to access specific parts of the database based on their role in the organization.

7. DBAs are responsible for maintaining detailed documentation of database configurations, schemas, procedures, backup policies, and disaster recovery plans.

8. DBAs troubleshoot performance issues, connectivity problems, and unexpected database errors.

Best Database Management System for Software Developers

1. Oracle.
2. MySQL.
3. PostgreSQL.
4. MongoDB.
5. Redis.
6. ElasticSearch.
7. MariaDB.

Database Users

- A Database User is defined as a person who interacts with data daily, updating, reading, and modifying the given data.

Types of Database Users

1.Database Administrators (DBA):

Manage the database system, perform backups, ensure security, and monitor performance.

2. Application Programmers

- Application programmers in a database are users who write software programs that interact with the database.
- They create and manage applications that perform tasks like adding, retrieving, updating, or deleting data, using languages

like SQL to communicate with the database.

3.System analyst

- System analysts are users who analyze and design database systems.
- They focus on understanding business requirements and translating them into technical database solutions.
- They ensure that the database structure meets the needs of the organization and optimize performance, security, and data flow.

4.Naive user

- Naive users are users who interact with the database through simple interfaces without knowing how the database works.
- Example: Withdrawing money from an ATM or booking tickets online.

5.Sophisticated users

- They are users who interact directly with the database using advanced tools or query languages like SQL.

. They understand the database's structure and can run complex queries, analyze data, and perform tasks without needing detailed knowledge of how the database is managed internally.

6.Database designer

- A database designer is responsible for creating the structure of a database.
- They define how data will be stored, organized, and related within the system by designing tables, relationships, constraints.

7.Specialized users

They are the user who write custom database applications or perform complex operations not supported by standard query languages.

8.Casual users, or temporary users

They are the individuals who access the database occasionally without needing detailed knowledge of its internal structure.

Basics	File System	DBMS
Structure	The file system is a way of arranging the files in a storage medium within a computer.	DBMS is software for managing the database.
Data Redundancy	Redundant data can be present in a file system.	In DBMS there is no redundant data.
Backup and Recovery	It doesn't provide Inbuilt mechanism for backup and recovery of data if it is lost.	It provides in house tools for backup and recovery of data even if it is lost.
Query processing	There is no efficient query processing in the file system.	Efficient query processing is there in DBMS.
Complexity	It is less complex as compared to DBMS.	It has more complexity in handling as compared to the file system.

Basics	File System	DBMS
Security Constraints	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file systems.
Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.
User Access	Only one user can access data at a time.	Multiple users can access data at a time.
Sharing	Data is distributed in many files. So, it is not easy to share data.	Due to centralized nature data sharing is easy
Attributes	To access data in a file , user requires attributes such as file name, file location.	No such attributes are required.

Database Management System

UNIT 1

INTRODUCTION

Today's Target

- Data Abstraction in DBMS
- DBMS Architecture
- Data Independence in DBMS

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

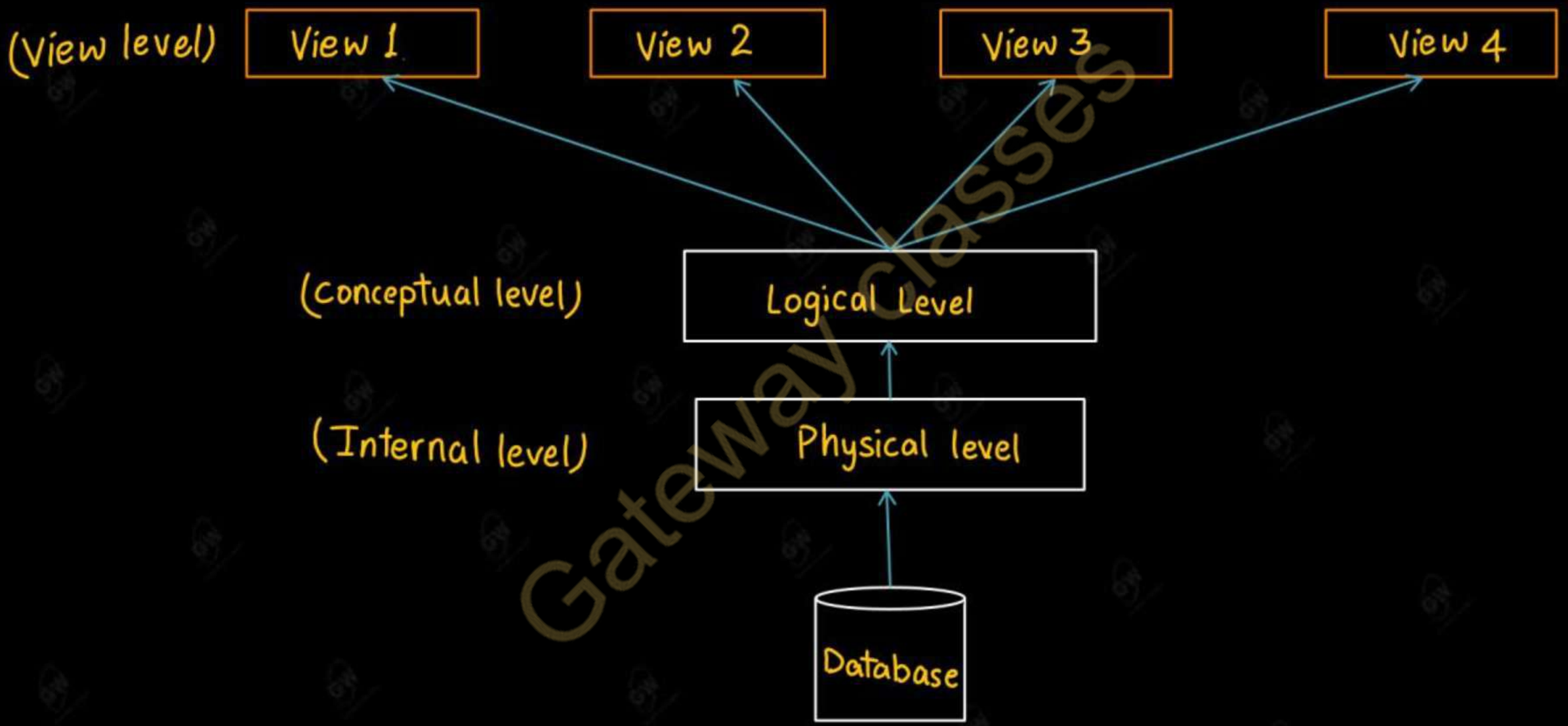
Data Abstraction in DBMS

- 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.

Levels of Abstraction in DBMS

There are three levels of data abstraction in DBMS .

1. Physical or internal level.
2. logical or conceptual level.
3. view or external level.



Physical or Internal Level

- It is the lowest level of data abstraction which defines how data is stored in database .
- It defines data structures used to store data and methods to access data in database.

Logical or Conceptual Level

- It is intermediate level present next to physical level.
- It defines what data is present in database and their relationships between them .
- It is less complex as compared to physical level.

View or External Level

- It is the highest level in abstraction.
- There are different levels of views and each view defines only a part of whole data required to user.

Physical or Internal Level

- It is very complex to understand and hence kept hidden from user.
- Database administrator decides how and where to store the data in database.
- Physical level deals with actual storage details like data organization, disk space allocation and data access methods.

Logical or Conceptual Level

- Programmers generally work at this level and depending on data, structure of tables, relationships and their constraints is decided at this level.

View or External Level

- This level defines many views of same database for simple view to user.
- This is the highest level and easiest to understand for user.

DBMS Architecture

DBMS architecture depends upon how users are connected to the database to get their request done .

Types of DBMS Architecture

1-Tier Architecture

2-Tier Architecture

3-Tier Architecture

1-Tier Architecture

- In 1-Tier Architecture the database is directly available to the user
- The user can directly sit on the DBMS and use it that is, the client, server, and Database are all present on the same machine.
- Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.
- The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.
- to learn SQL we set up an SQL server and the database on the local system. This enables us to directly interact with the relational database and execute operations

Advantages of 1-Tier Architecture

1.Simple Architecture: 1-Tier Architecture is the most simple architecture to set up, as only a single machine is required to maintain it.

2. Cost-Effective: No additional hardware is required for implementing 1-Tier Architecture, which makes it cost-effective.

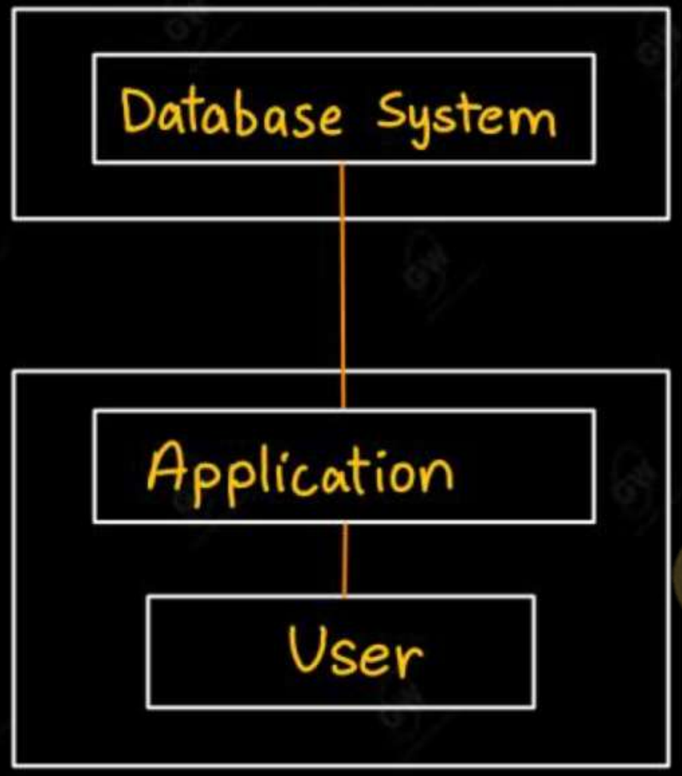
3.Easy to Implement: 1-Tier Architecture can be easily deployed, and hence it is mostly used in small projects.

2-Tier Architecture

- The 2-Tier architecture is same as basic client-server model.
- 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 server side is responsible for providing query processing and transaction management functionalities.

On the client side, the user interfaces and application programs are run.

- The application on the client side establishes a connection with the server side to communicate with the DBMS.



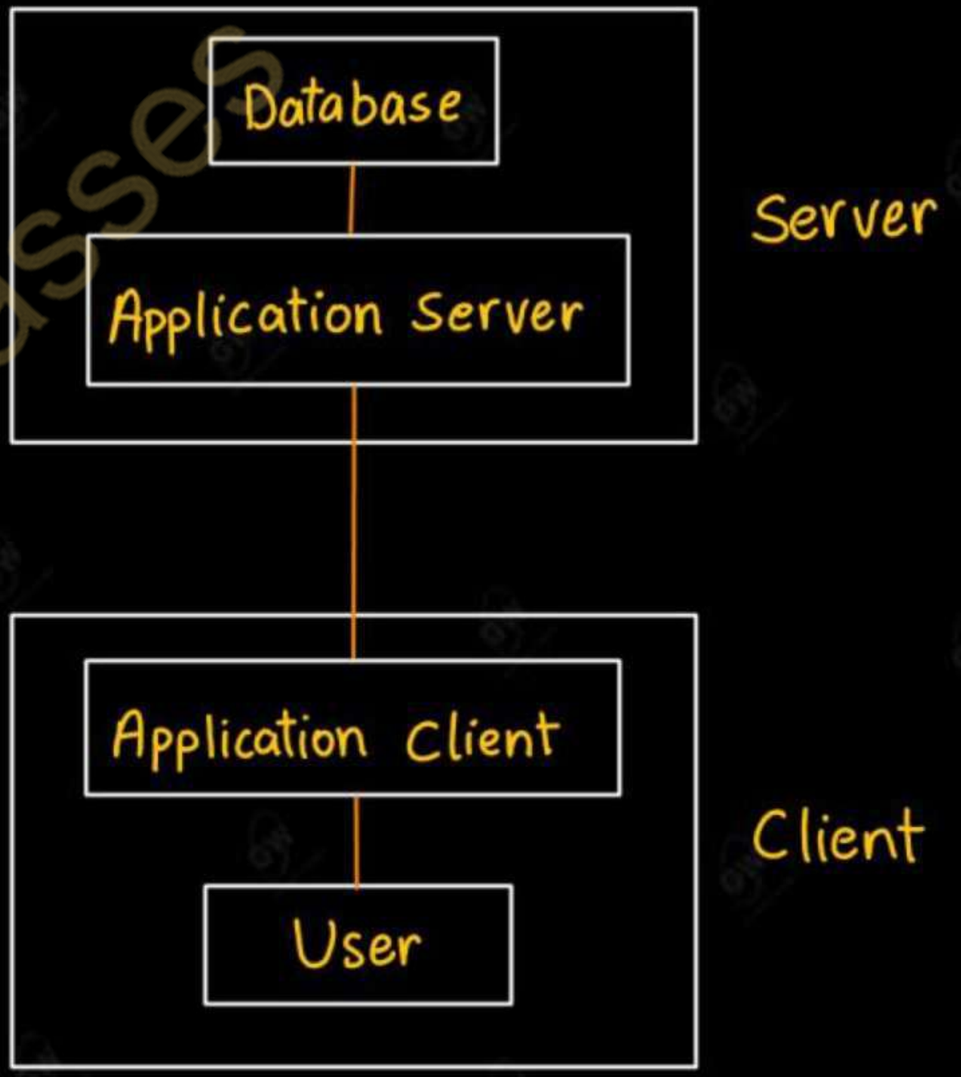
Advantages of 2-Tier Architecture

- **Easy to Access:** 2-Tier Architecture makes easy access to the database, which makes fast retrieval.
- **Scalable:** We can scale the database easily, by adding or upgrading hardware.
- **Low Cost:** 2-Tier Architecture is cheaper than 3-Tier Architecture and Multi-Tier Architecture.
- **Easy Deployment:** 2-Tier Architecture is easier to deploy than 3-Tier Architecture.
- **Simple:** 2-Tier Architecture is easily understandable as well as simple because of only two components

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.



Advantages of 3-Tier Architecture

1. **Enhanced scalability:** Scalability is enhanced due to the distributed deployment of application servers. Now, individual connections need not be made between the client and server.
2. **Data Integrity:** 3-Tier Architecture maintains Data Integrity. Since there is a middle layer between the client and the server, data corruption can be avoided/removed.
3. **Security:** 3-Tier Architecture Improves Security. This type of model prevents direct interaction of the client with the server thereby reducing access to unauthorized data.

Disadvantages of 3-Tier Architecture

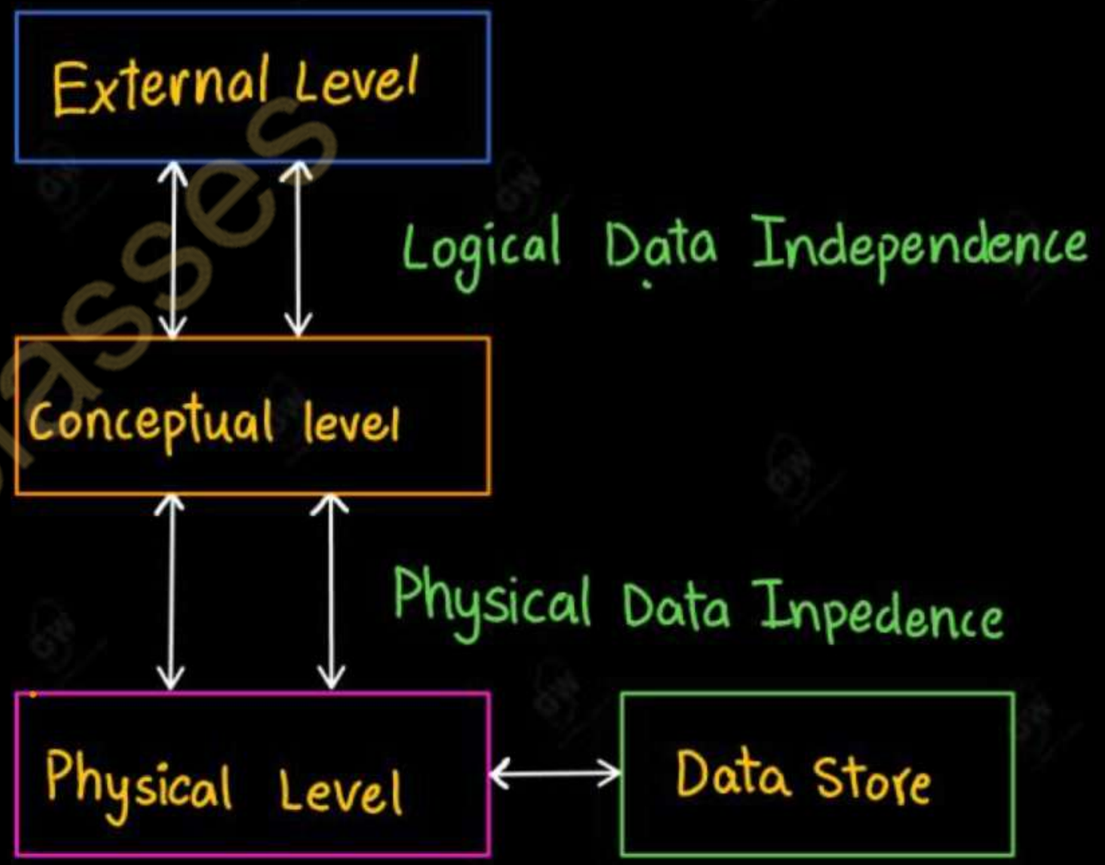
1. **More Complex:** **3-Tier Architecture is more complex in comparison to 2-Tier Architecture. Communication Points are also doubled in 3-Tier Architecture.**
2. **Difficult to Interact:** **It becomes difficult for this sort of interaction to take place due to the presence of middle layers.**

Data Independence in DBMS

Data independence is a property of a database management system by which we can change the database schema at one level of the database system without changing the database schema at the next higher level.

There are two types of data independence.

1. logical data independence
2. Physical data independence



Physical data independence

- Physical Data Independence can be defined as the ability to change the physical level without affecting the logical or Conceptual level.
- Physical data independence gives us the freedom to modify the - Storage device, File structure, location of the database, etc. without changing the definition of conceptual or view level.

changes can be done at the physical layer without affecting the conceptual layer -

- Changing the storage devices like SSD, hard disk and magnetic tapes, etc.
- Changing the access technique and modifying indexes.
- Changing the compression techniques or hashing algorithms.

Logical data independence

- Logical Data Independence is a property of a database that can be used to change the logic behind the logical level without affecting the other layers of the database.
- Logical data independence is usually required for changing the conceptual schema without having to change the external schema or application programs.
- It allows us to make changes in a conceptual structure like adding, modifying, or deleting an attribute in the database.

These changes can be done at a logical level without affecting the application program or external layer.

- Adding, deleting, or modifying the entity or relationship.
- Merging or breaking the record present in the database

Advantages

- As the database grows, data independence facilitates the addition of new data elements or tables without affecting existing queries or applications.
- Developers can focus on designing and building applications without needing to worry about changes in the underlying database structure.
- Data independence allows for security measures and access controls to be implemented at the logical level, protecting sensitive data from unauthorized access or modification
- As technology evolves, the physical storage and organization of data may need to change to take advantage of new hardware or software capabilities. Data independence allows these changes to be made without affecting the logical schema.
- Changes to the logical schema do not impact the application programs or queries that rely on the database. This means that existing applications can continue to function correctly even when the database structure changes, reducing the risk of disruptions.

Disadvantages

- Managing data independence may require additional system resources, such as storage space and processing power, to handle the various schema layers and translations between them.
- Database administrators and developers may require specific training and expertise to effectively manage data independence in a DBMS.
- While data independence simplifies
- schema changes, it may not eliminate all complexities associated with data migration. Migrating data between different versions of the database organization or across different DBMS platforms can still be complex and time-consuming.
- Changes in the logical schema, if not managed carefully, can lead to data integrity issues. Ensuring that data remains consistent and that referential integrity constraints are maintained can be challenging when altering the logical schema.

Database Management System

UNIT 1
INTRODUCTION

Today's Target

- Data models
- Database schema and instances
- AKTU PYQs

By PRAGYA RAJVANSHI

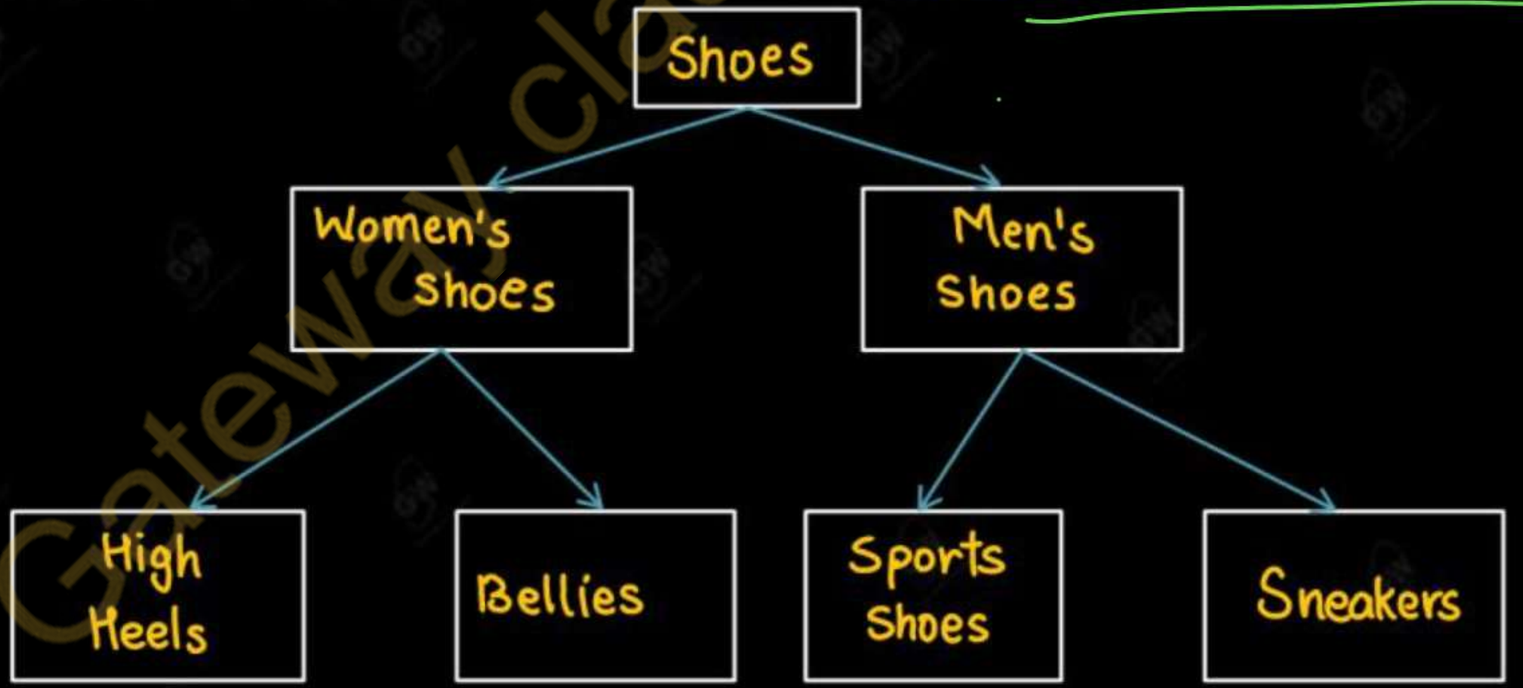
B.Tech, M.Tech(C.S.E.)

Data models

- Data Model gives us an idea that. how the final system will look like after its complete implementation.
- Data Models are used to show how data is stored, connected, accessed and updated in the database management system.

1. Hierarchical Model

- Hierarchical Model was the first DBMS model.
- This model organizes the data in the hierarchical tree structure.
- The hierarchy starts from the root which has root data and then it expands in the form of a tree adding child node to the parent node.



Features of a Hierarchical Model

One-to-many relationship: The data here is organized in a tree-like structure where the one-to-many relationship is between the datatypes.

Parent-Child Relationship: Each child node has a parent node but a parent node can have more than one child node.

Deletion Problem: If a parent node is deleted then the child node is automatically deleted.

Pointers: Pointers are used to link the parent node with the child node and are used to navigate between the stored data

Advantages:

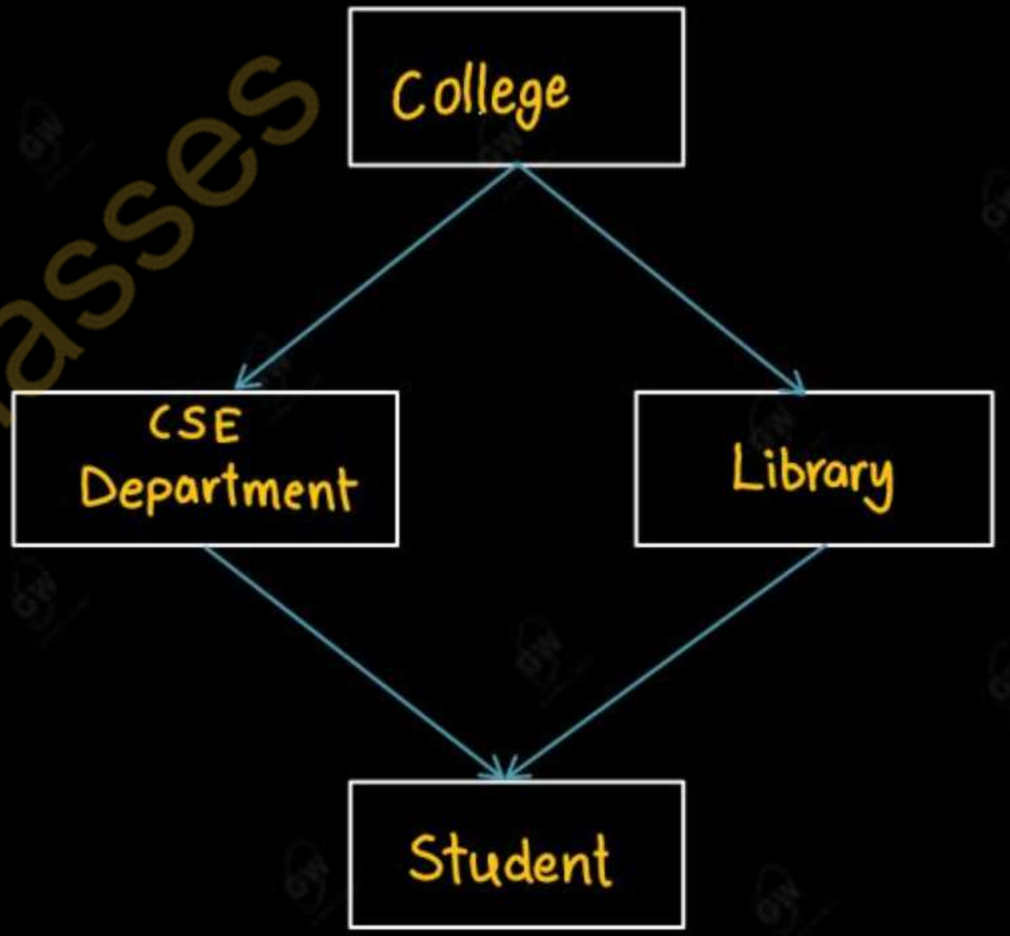
- Any change in the parent node is automatically reflected in the child node so, the integrity of data is maintained.

Disadvantages:

- Complex relationships are not supported.
- It does not support more than one parent of the child node.

2.Network Model

- This model is the same as the hierarchical model, the only difference is that a record can have more than one parent.
- It replaces the hierarchical tree with a graph.
- The data can be accessed faster as compared to the hierarchical model. This is because the data is more related in the network model and there can be more than one path to reach a particular node. So the data can be accessed in many ways.
- As there are more relationships so there can be more than one path to the same record.



Features of a Network Model

- In this model, as there are more relationships so data is more related. This model has the ability to manage one-to-one relationships as well as many-to-many relationship.
- The operations on the network model are done with the help of the circular linked list.

Advantages:

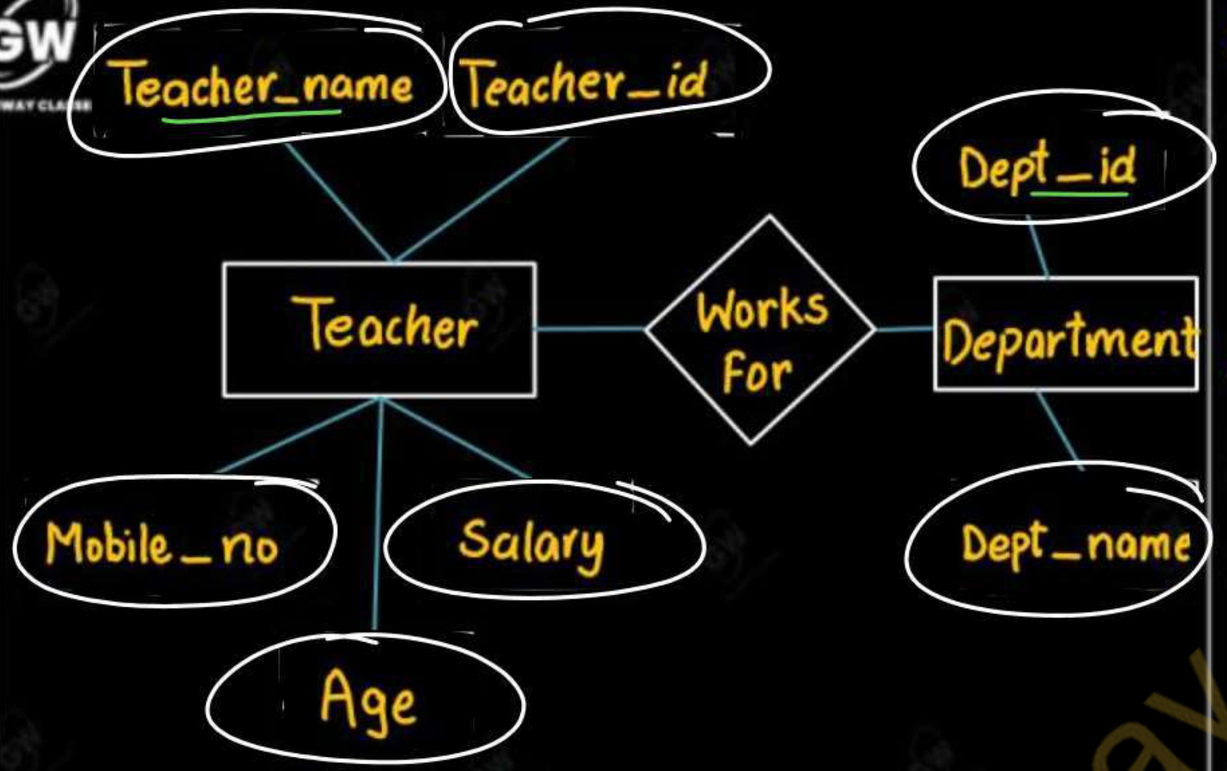
- The data can be accessed faster.
- data integrity is present.

Disadvantages:

- Any change like updation , deletion, insertion is very complex
- As more and more relationships need to be handled the system might get complex.

3.Entity-Relationship Model

- In this model, we represent the real-world problem in the pictorial form to make it easy for the stakeholders to understand.
- It is also very easy for the developers to understand the system by just looking at the ER diagram. We use the ER diagram as a visual tool to represent an ER Model.
- This model is used widely by the database designers for communicating their ideas



Features of ER Model

1.Graphical Representation for Better Understanding: It is very easy and simple to understand so it can be used by the developers to communicate with the stakeholders.

2. *Database Design:* This model helps the database designers to build the database and is widely used in database design.

Advantages of ER Model

- ER Model is very easy to build.
- This model is used widely by the database designers for communicating their ideas.
- This model maps well to the relational model and can be easily converted relational model by converting the ER model to the table.

Disadvantages of ER Model

- Some information might be lost or hidden in the ER model. As it is a high-level view so there are chances that some details of information might be hidden.

4.Relational Model

Relational Model is the most widely used model. In this model, the data is maintained in the form of a two-dimensional table. All the information is stored in the form of row and columns.

Emp_id	Emp_name	Job_name	Salary	Mobile_no	Dep_id	Project_id
AfterA001	Avi	Engineer	100000	9016.....	2	99
AfterA02	Alex	Analyst	50000	7462....	3	100
AfterA03	Adarsh	Manager	80000	8923....	2	65

Employee

Advantages

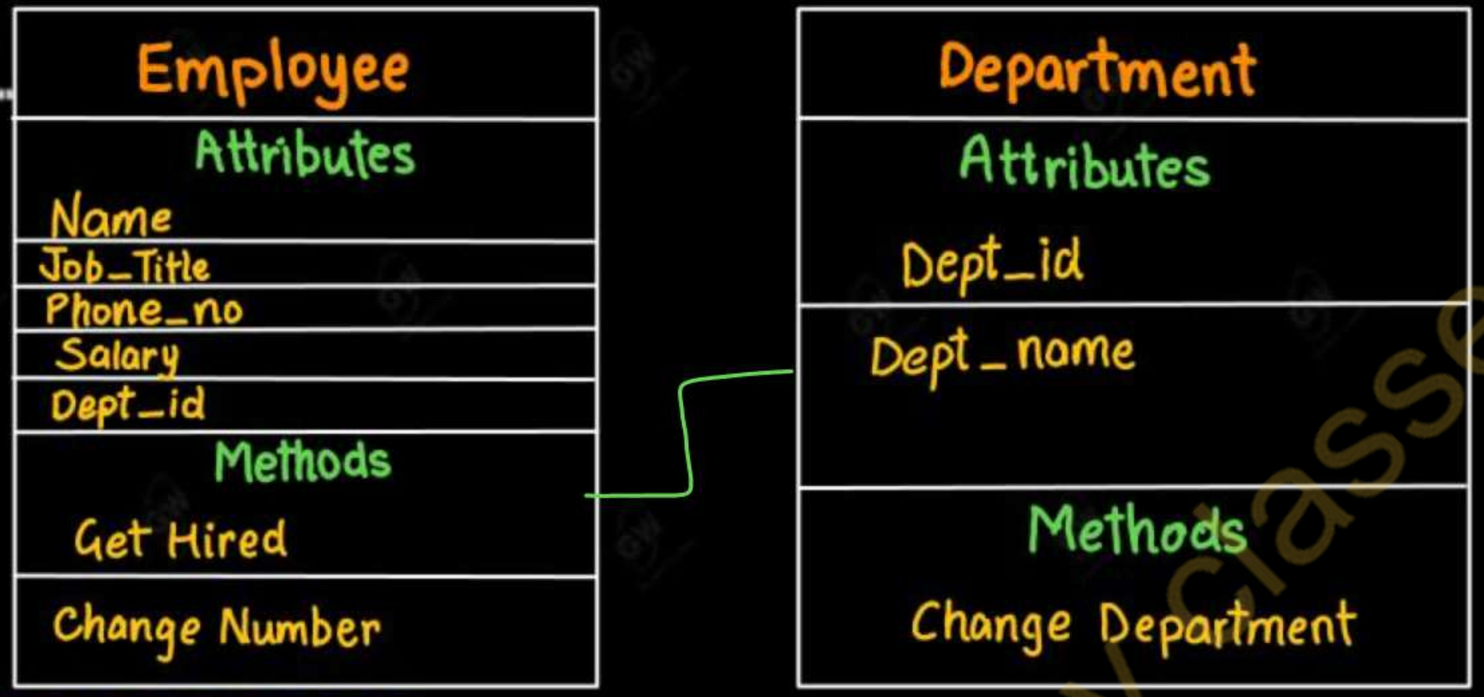
- This model is more simple as compared to the network and hierarchical model.
- This model can be easily scaled as we can add as many rows and columns we want.

Disadvantages

- For hiding the complexities and making things easier for the user this model requires more powerful hardware computers and data storage devices.

5.Object-Oriented Data Model

- In this model, **both the data and relationship** are present in a single structure known as an object.
- We can **store audio, video, images, etc** in the database which was not possible in the relational model(although you can store audio and video in relational database, it is advised not to store in the relational database).
- In this model, **two are more objects are connected through links**. We use this link to relate one object to other objects.



6.FLAT DATA MODEL

- Flat Data Model is a simple, two-dimensional model where data is stored in a single table or file, with no relationships between records.
- Each record has a fixed number of fields (columns), and all records follow the same structure.
- It's often used in spreadsheets or CSV files, where each row represents a single record, and each column represents a specific attribute of that record.

7.Object-Relational Model

- The Object-Relational Model integrates features from both object-oriented and relational models.
- It extends the relational model by allowing complex data types, like objects, to be stored in relational databases.
- This model supports inheritance, encapsulation, and polymorphism while maintaining traditional SQL for querying.

8. The Associative Data Model

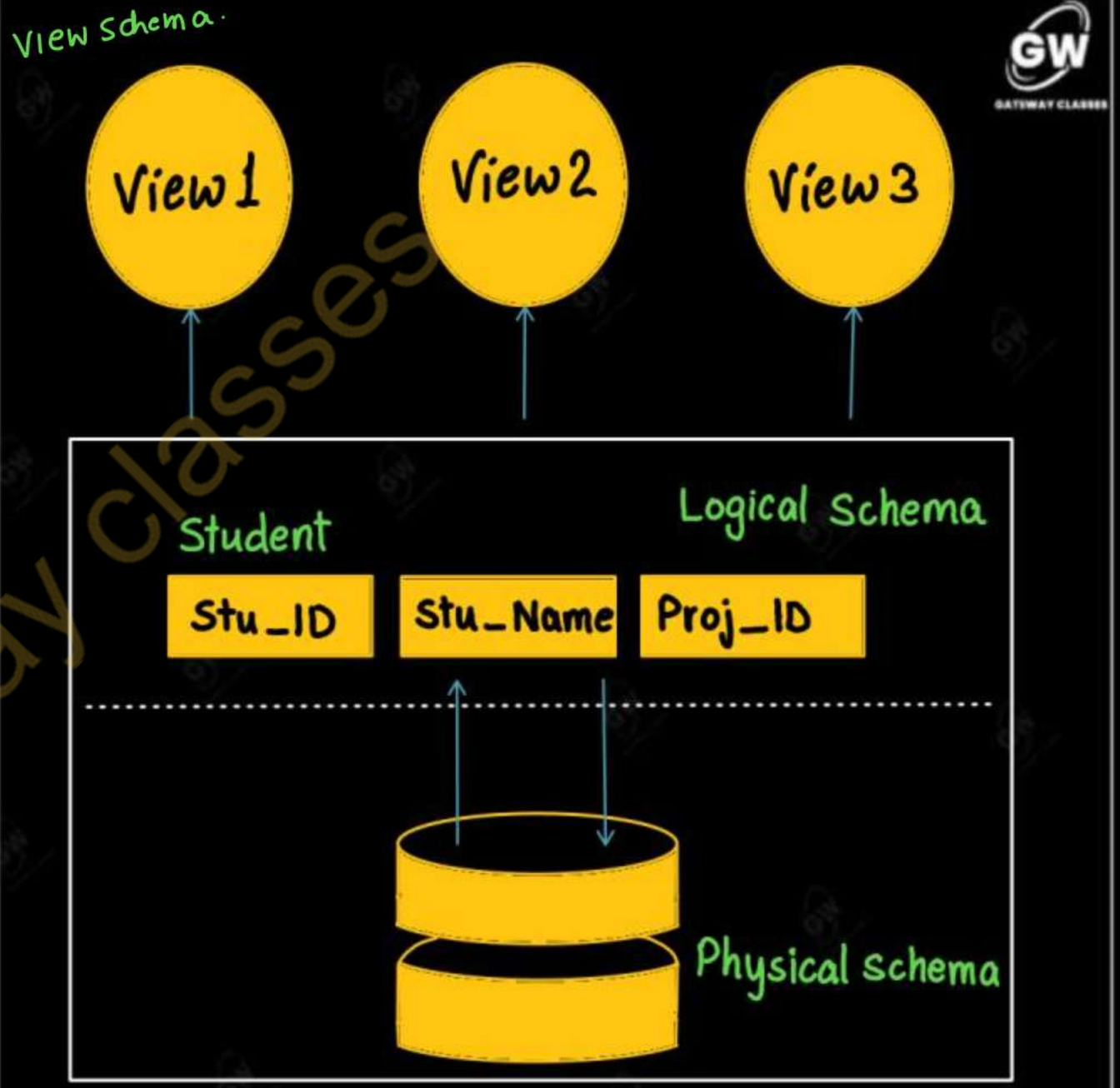
- It stores data as two parts: items (things) and associations (connections between things).
- Instead of tables or hierarchies, it focuses on how things are related to each other, making it easier to handle complex relationships like many-to-many links.

9. Context Data Model

Context Data Model is a collection of several models. This consists of models like network model, relational models etc. Using this model we can do various types of tasks which are not possible using any model alone.

Database Schema

- A database schema is the skeleton structure that represents the logical view of the entire database
- It defines how the data is organized and how the relations among them are associated.
- It formulates all the constraints that are to be applied on the data.
- A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database.



A database schema can be divided broadly into three categories –

1.Physical Database Schema –

- This schema concerns to the actual storage of data and its form of storage like files, indices, etc.
- It defines how the data will be stored in a secondary storage.

2.Logical Database Schema

- This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.
- The Logical schema describes how the data is stored in the form of tables & how the attributes of a table are connected.
- Using ER modeling the relationship between the components of the data is maintained.

3.View Schema

- It represents the way data is presented to users in a database, typically through views.
- A view is a virtual table generated by querying data from one or more base tables. Unlike actual tables, views don't store data physically; instead, they display data dynamically based on the underlying query

Database instances

- A database instance is a state of operational database with data at any given time.
- It contains a snapshot of the database. Database instances tend to change with time.
- A DBMS ensures that its every instance (state) is in a valid state, by diligently following all the validations, constraints, and conditions that the database designers have imposed.

Database Management System

UNIT 1
INTRODUCTION

Today's Target

- Keys and its types
- Database language
- AKTU PYQs

By PRAGYA RAJVANSHI

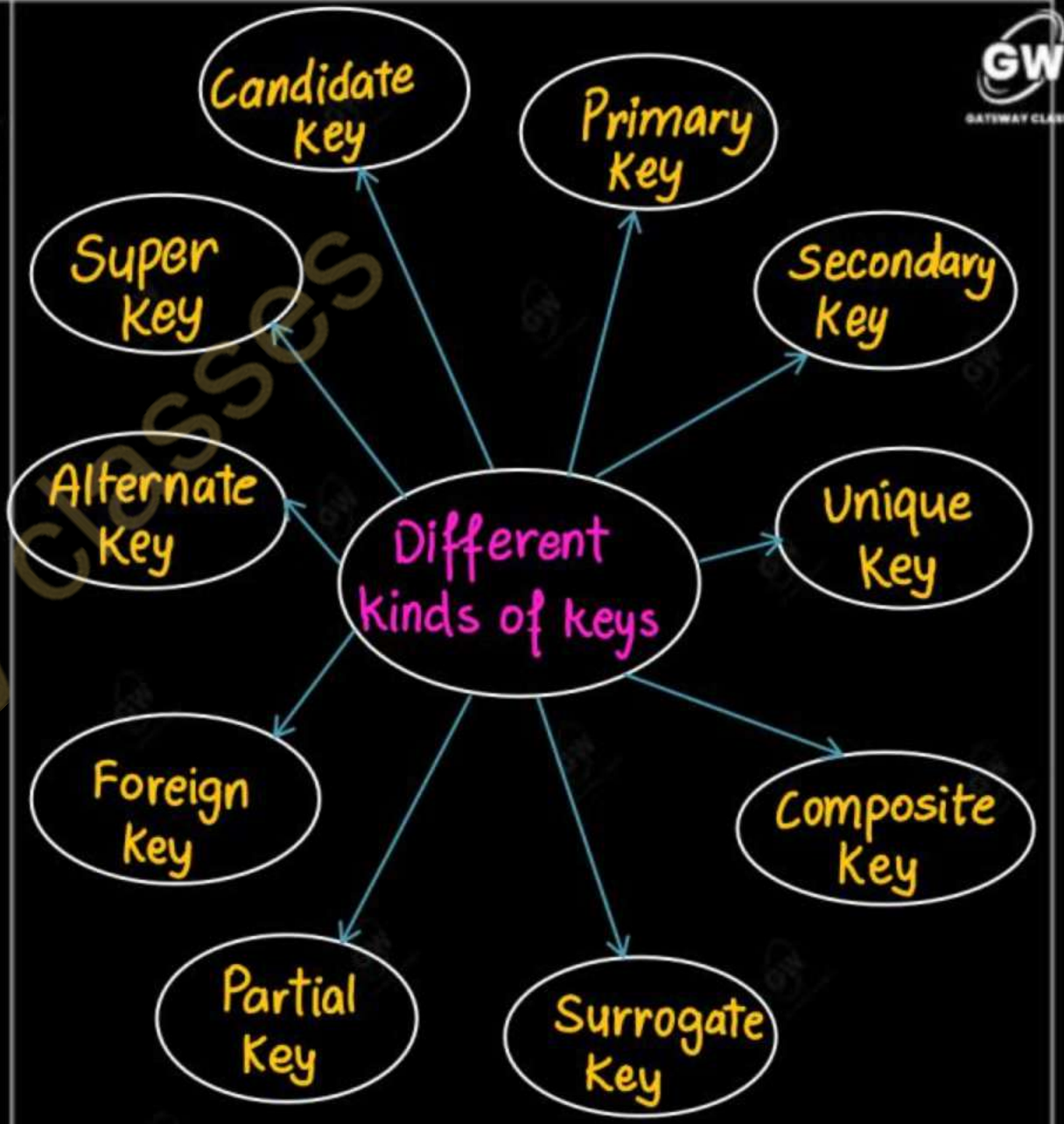
B.Tech, M.Tech(C.S.E.)

Keys & Types of keys

A key is a set of attributes that can identify each tuple uniquely in the given relation.

Note

- In DBMS, a relation is a table that consists of rows and columns.
- Each row represents a record (tuple), and each column represents an attribute (field).
- The relation is the fundamental building block in a relational database.



Student (Relation)

Roll	Name	Gen	%
1	A	F	28%
2	B	F	62%
3	B	M	28%
4	C	M	29%

tuple
Record

Roll, Name, Gender, %
Attribute

Gender F

A
B

F

BX

Name	Fathe	Adde
A	X	165
B	Y	165
A	Z	92
B	X	82

(Name, Fathe)
↓
K

A X

Note:

- The terms 'relation' and 'table' are used interchangeably.
- The terms 'tuple' and 'record' are used interchangeably.

1. Super Key-

- A super key is a set of attributes that can identify each tuple uniquely in the given relation.
- A super key is not restricted to have any specific number of attributes

Student schema

Student (roll , name , sex , age , address , class , section)

examples of super keys

(roll , name , sex , age , address , class , section)

(class , section , roll)

(class , section , roll , sex)

(name , address)

(roll)

Note:

- All the attributes in a super key are definitely sufficient to identify each tuple uniquely in the given relation but all of them may not be necessary.

Student

Roll	name	sex	age	Address	Class	section
1	W	F	19	256	CSE	A
2	W	F	20	659	CSE	B
3	B	M	23	423	ME	C
4	Y	M	24	332	CSE	D
5	Z	F	25	659		E
6	Z	F	26	238	CW	
7	X	M	21	999	I	A

Roll
Address, name

W256
W 659

name Address (Rollname)
W659

25
659

2. Candidate Key-

- A minimal super key is called as a candidate key.
- A set of minimal attribute that can identify each tuple uniquely in the given relation is called as a candidate key.

Student schema-

Student (roll , name , sex , age , address , class , section)

Example of candidate keys

(name , address)

(roll)

NOTE:

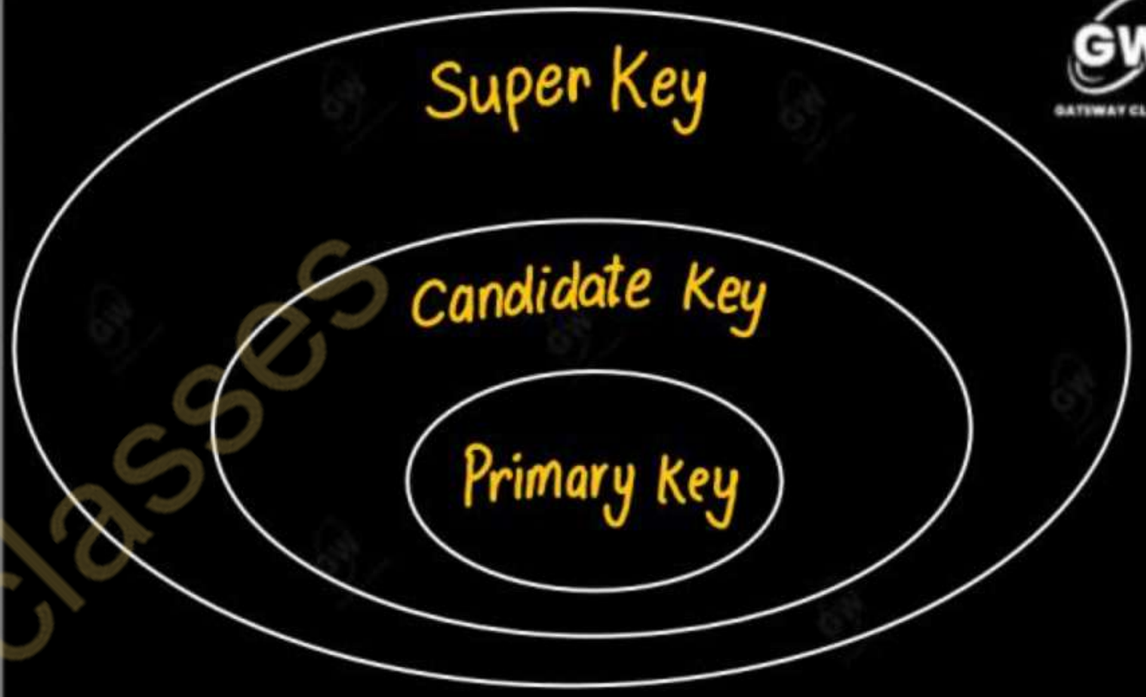
- All the attributes in a candidate key are sufficient as well as necessary to identify each tuple uniquely.
- Removing any attribute from the candidate key fails in identifying each tuple uniquely.
- The value of candidate key must always be unique.
- The value of candidate key can never be NULL.
- It is possible to have multiple candidate keys in a relation.
- Those attributes which appears in some candidate key are called as prime attributes.

3. Primary Key-

- A primary key is a candidate key that the database designer selects while designing the database.

Note:

- The value of primary key can never be NULL.
- The value of primary key must always be unique.
- The values of primary key can never be changed i.e. no updation is possible.
- The value of primary key must be assigned when inserting a record.
- A relation is allowed to have only one primary key.

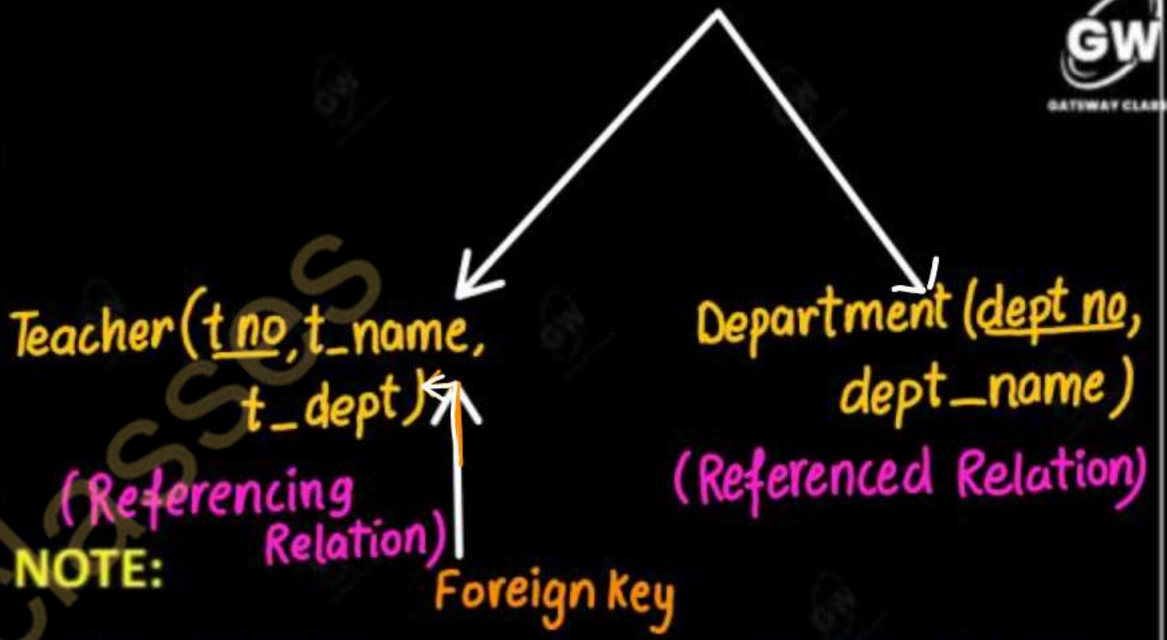


4. Alternate Key-

Unimplemented candidate keys are called as alternate keys.

5. Foreign Key-

- An attribute 'X' is called as a foreign key to some other attribute 'Y' when its values are dependent on the values of attribute 'Y'.
- The attribute 'X' can assume only those values which are assumed by the attribute 'Y'.
- Here, the relation in which attribute 'Y' is present is called as the referenced relation.
- The relation in which attribute 'X' is present is called as the referencing relation.
- The attribute 'Y' might be present in the same table or in some other table.



NOTE:

- Foreign key references the primary key of the table.
- Foreign key can take only those values which are present in the primary key of the referenced relation.

teacher

t_no	t_no	t.dept.
A1	XY	1
A2	ZX	1
A3	WX	2
A4	ZX	3
A5	ZX	~~6~~

Department

dept no	Dept name
1	CSE
2	AIML
3	DS
4	ME
5	Civil

Foreign key can take the NULL value.

- There is no restriction on a foreign key to be unique..
- Referenced relation may also be called as the master table or primary table.
- Referencing relation may also be called as the foreign table.

Partial key

- Partial key is a key using which all the records of the table can not be identified uniquely.

Department (Emp_no , Dependent_name , Relation)

Emp_no is partial key

Emp_no	Dependent_name	Relation
E1	Suman	Mother
E1	Ajay	Father
E2	Vijay	Father
E2	Ankush	Son

7. Composite Key-

- It is a combination of two or more columns in a table that uniquely identifies a row.
- None of the individual columns can uniquely identify the row on their own.

8. Unique Key-

- It is unique for all the records of the table.
- Once assigned, its value can not be changed i.e. it is non-updatable.
- It may have a NULL value.

Example email id

9. Surrogate key

A column that is not generated from the data in the database is known as a surrogate key. Rather, the DBMS generates a unique identifier for you. In database tables, surrogate keys are frequently utilized as primary keys.

Features of the Surrogate Key

- It is automatically generated by the system.
- It holds an anonymous integer.
- It contains a unique value for all records of the table.
- The value can never be modified by the user or application.
- The surrogate key is called the factless key as it is added just for our ease of identification of unique values and contains no relevant fact(or information) that is useful for the table

registration_no	name	percentage
210101	Harry	90
210102	Maxwell	65
210103	Lee	87
210104	Chris	76

registration_no	name	percentage
CS107	Taylor	49
CS108	Simon	86
CS109	Sam	96
CS110	Andy	58

Surr_no	registration_no	name	percentage
1	210101	Harry	90
2	210102	Maxwell	65
3	210103	Lee	87
4	210104	Chris	76
5	CS107	Taylor	49
6	CS108	Simon	86
7	CS109	Sam	96
8	CS110	Andy	58

Secondary Key

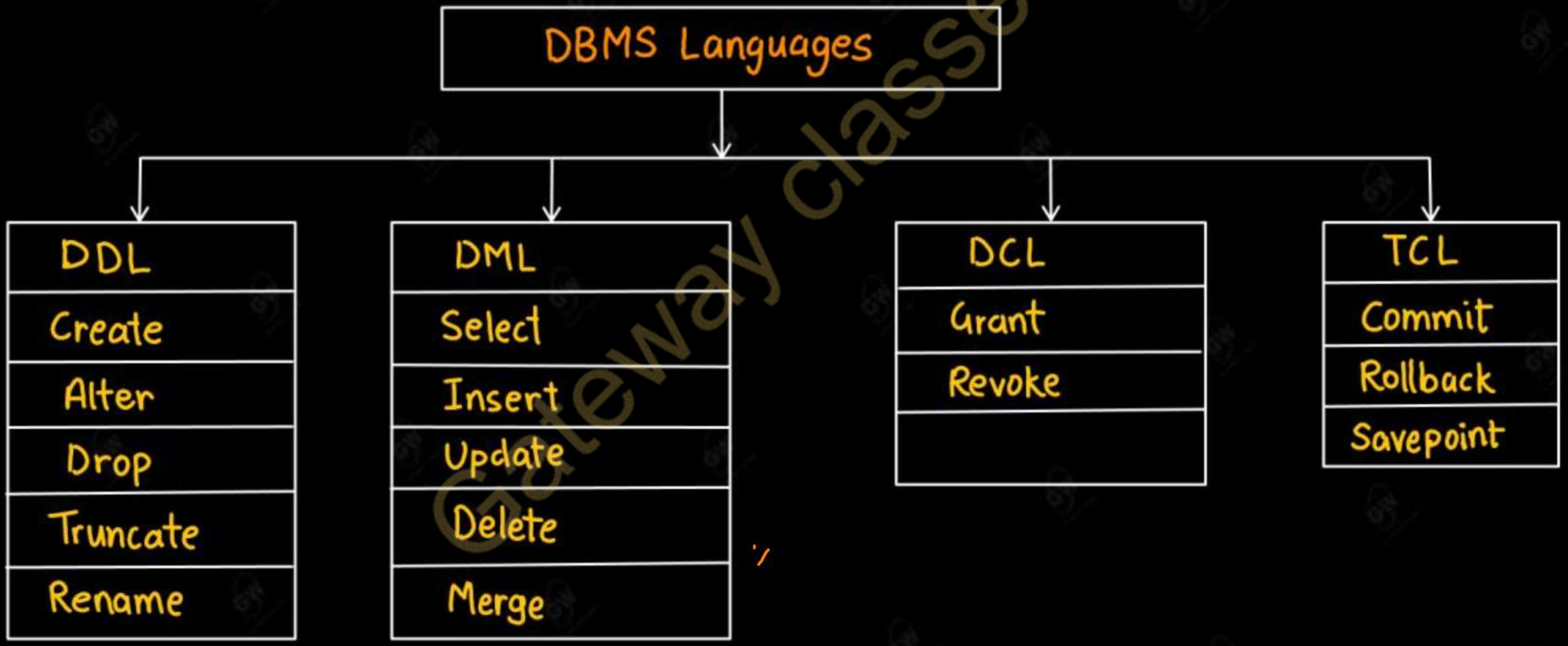
- Secondary Key is the key that has not been selected to be the primary key.
- However, it is considered a candidate key for the primary key.
- A candidate key not selected as a primary key is called secondary key.
- Note: Secondary Key is not a Foreign Key.

Student_ID	Student_Enroll	Student_Name	Student_Age	Student_Email
096	9122717	Manish	25	aaa@gmil.com
055	9122655	Manan	23 ..	abc@gmail.com
067	9122699	Shreyas	28	pqr@gmail.com

- **Student_ID, Student_Enroll** and **Student_Email** are the candidate keys. Searching Indexing
- **Student_ID** as primary key, therefore **Student_Enroll** and **Student_Email** will be Secondary Key (candidates of primary key)

Database Languages in DBMS

- In DBMS, a database language refers to the set of languages used to define, manipulate, control, and retrieve data from a database.



DDL (Data Definition Language)

- DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema.
- It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.
- DDL is a set of SQL commands used to create, modify, and delete database structures but not data.

CREATE	It is used to create new database objects like tables, views, indexes, or databases
ALTER	It is used to modify an existing database object like a table by adding, deleting, or modifying columns.

TRUNCATE	It is used to delete all rows from a table quickly, but it keeps the table structure intact for future use.
RENAME	It is used to change the name of a database object, like a table or column.
DROP	It is used to delete database objects like tables, databases, views, or indexes permanently.

DML (Data Manipulation Language)

- It is a way to access and manipulate data stored within the database.
- It includes operations such as inserting, extracting, or updating the data within the database.

INSERT	Adds new data into a table
UPDATE	Modifies existing data in a table.
DELETE	Removes data from a table.
SELECT DQL	It is used to retrieve data from a database. It allows querying and fetching data from one or more tables.
MERGE	insert new records or modify existing records depending upon whether condition matches.

DCL (Data Control Language)

- DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.

GRANT	Assigns new privileges to a user account, allowing access to specific database objects, actions, or functions.
REVOKE	Removes previously granted privileges from a user account, taking away their access to certain database objects or actions.

TCL (Transaction Control Language)

- TCL (Transaction Control Language) in SQL is used to manage transactions within a database.
- Transactions are sequences of one or more SQL operations executed as a single unit of work.

TCL commands help ensure data integrity and consistency.

COMMIT	Saves all changes made during the current transaction. It makes the transaction's changes permanent.
ROLLBACK	Reverts all changes made during the current transaction. It undoes the transaction if something goes wrong.
SAVEPOINT	Sets a point within a transaction to which you can later roll back. Useful for partial rollbacks.

AKTU PYQs

Q.1	What are the different types of models in DBMS? Explain them.	AKTU 2022-23 AKTU 2020-21 AKTU 2018-19 AKTU 2017-18
Q.2	Explain different types of keys Difference between primary key and unique key. Explain super key, Primary key, candidate key.	AKTU 2020-21 AKTU 2018-19/19-20 AKTU 2017-18 AKTU 2021-22
Q.3	What is the significance of physical data independence	AKTU 2020-21
Q.4	List four function of database administrator(DBA)	AKTU2022-21
Q.5	Explain DML,DDL	AKTU 2017-18/18-19/20-21
Q6	What is instance and schema of database(DB)	AKTU 2017-18 AKTU 2018-19
Q7	Difference between physical and logical independence.	AKTU 2022-23

AKTU PYQs

Q.8	What is data abstraction? Explain the level of data abstraction OR How data abstraction is achieve in DBMS?	AKTU 2022-23 AKTU 2020-21
Q.9	Briefly discuss about the difference between data system and file system(10 key points)	AKTU 2020-21 AKTU 2019-20 AKTU 2018-19
Q.10	What is data independence in DBMS . OR What is data independence and its types	AKTU 2020-21 AKTU 2021-22 AKTU 2019-20 AKTU 2017-18
Q.11	Explain the difference between physical level, conceptual level and view level	AKTU 2017-18
Q.12	What is the role of database administrator	AKTU 2017-18
Q.13	What is database language	AKTU 2018-19

Database Management System

UNIT 1
INTRODUCTION

Today's Target

- ER diagram
- AKTU PYQs

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

ER diagram

- ER diagram or Entity Relationship diagram is a conceptual model that gives the graphical representation of the logical structure of the database.
- It shows all the constraints and relationships that exist among the different components.

Components of ER diagram-

An ER diagram is mainly composed of following three components-

Entity Sets, Attributes, Relationship Set

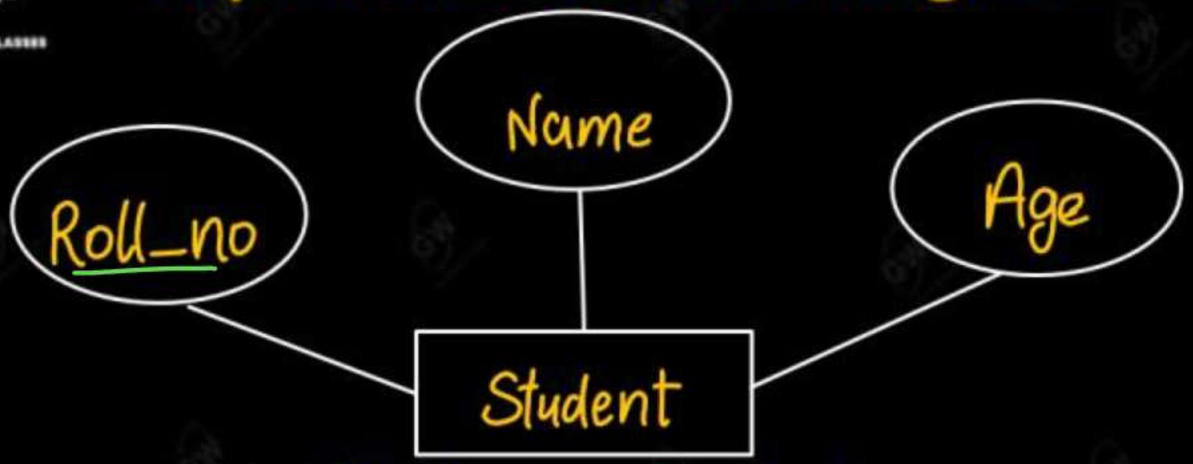
STUDENT

Roll_no	Name	Age
1	Akhil	20
2	Rahul	19
3	Pooja	20
4	Aarti	19

entity

This complete table is referred to as "Student Entity Set" and each row represents an "entity".

Representation as ER Diagram-



ER Diagram Symbols-

1. For Entity Sets-

- An entity set is a **set of same type of entities.**
- An entity refers to any **object** having-

1.Either a **physical existence** such as a particular **person, office, house or car.**

2. A conceptual existence such as **a department or a company.**

1. 1. Strong Entity Set-

- A strong entity set possess its own **primary key.**
- It is represented using a **single rectangle.**

1.2. Weak Entity Set-

- A weak entity set do not possess its own **primary key.**
- It is represented using a **double rectangle.**

Strong Entity set

Weak Entity Set

Example of Strong entity

- **Double line** between Student and relationship set signifies **total participation.**
- It suggests that **each student** must be enrolled in at **least one course.**

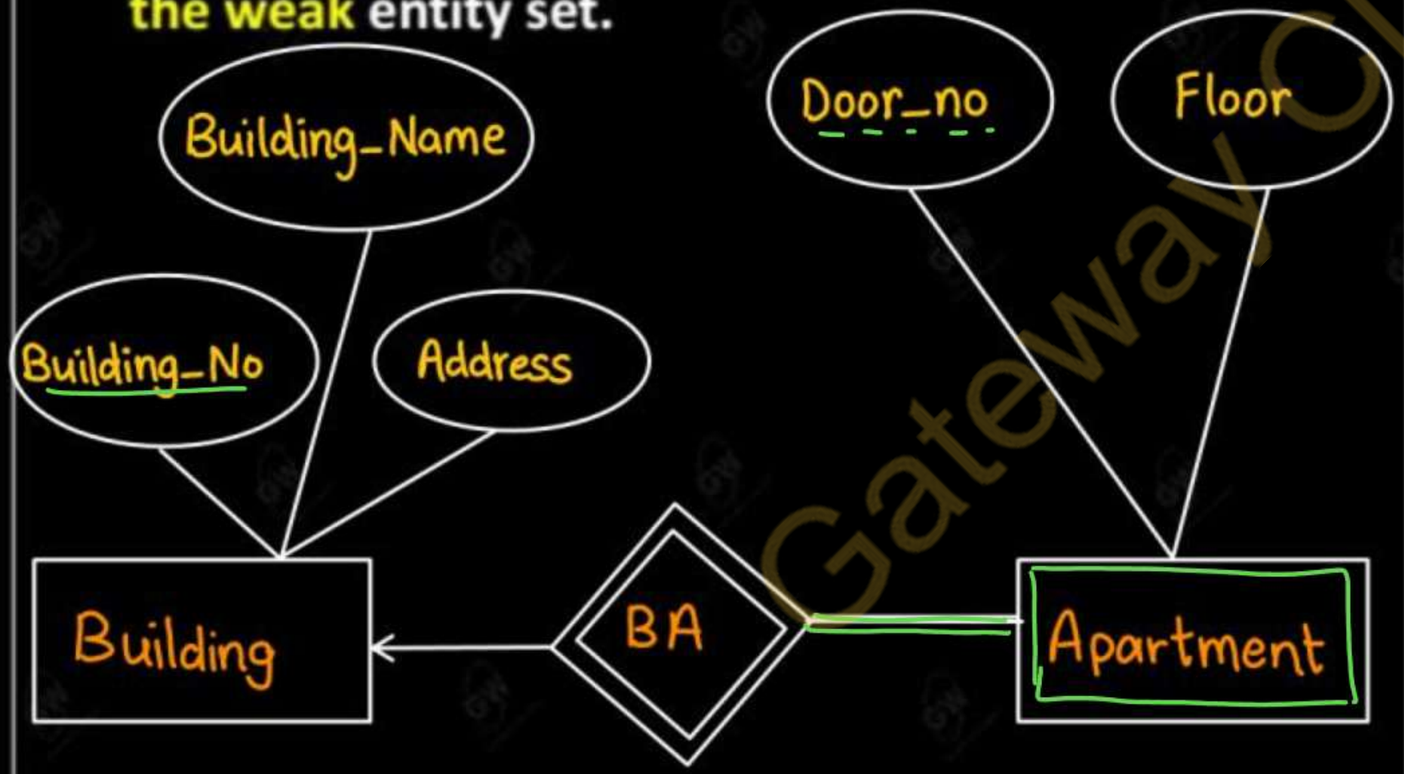
- **Single line** between Course and relationship set signifies **partial participation.**
- It suggests that there might exist some courses for which no enrollments are made.

Weak Entity Set-

- It contains a **partial key** called as a **discriminator.**
- Discriminator can **identify a group of entities** from the entity set.
- Discriminator is represented by underlining with a **dashed line.**

The combination of discriminator and primary key of the strong entity set makes it possible to uniquely identify all entities of the weak entity set.

- Thus, this combination serves as a primary key for the weak entity set.



- Double line between Apartment and relationship set signifies total participation.
- It suggests that each apartment must be present in at least one building.
- Single line between Building and relationship set signifies partial participation.
- It suggests that there might exist some buildings which has no apartment.

Primary key of weak entity set

= Its own discriminator + Primary key of strong entity set

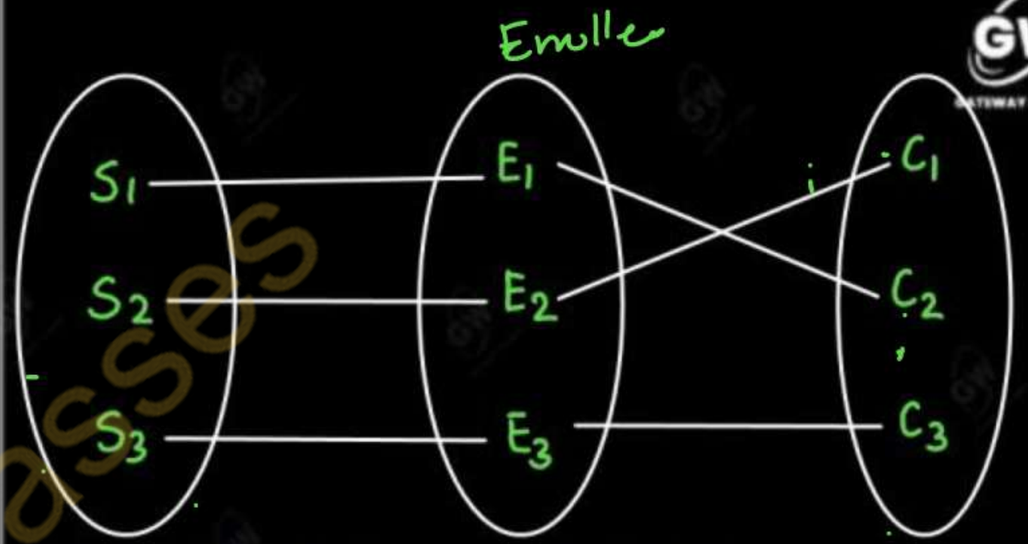
Strong Entity	Weak Entity
Strong entity always has a primary key.	While a weak entity has a partial discriminator key.
Strong entity is not dependent on any other entity.	Weak entity depends on strong entity.
Strong entity is represented by a single rectangle.	Weak entity is represented by a double rectangle.
Two strong entity's relationship is represented by a single diamond.	While the relation between one strong and one weak entity is represented by a double diamond.
Strong entities have either total participation or partial participation.	A weak entity has a total participation constraints.

Relationship

- A relationship is defined as an association among several entities.

Relationship Set-

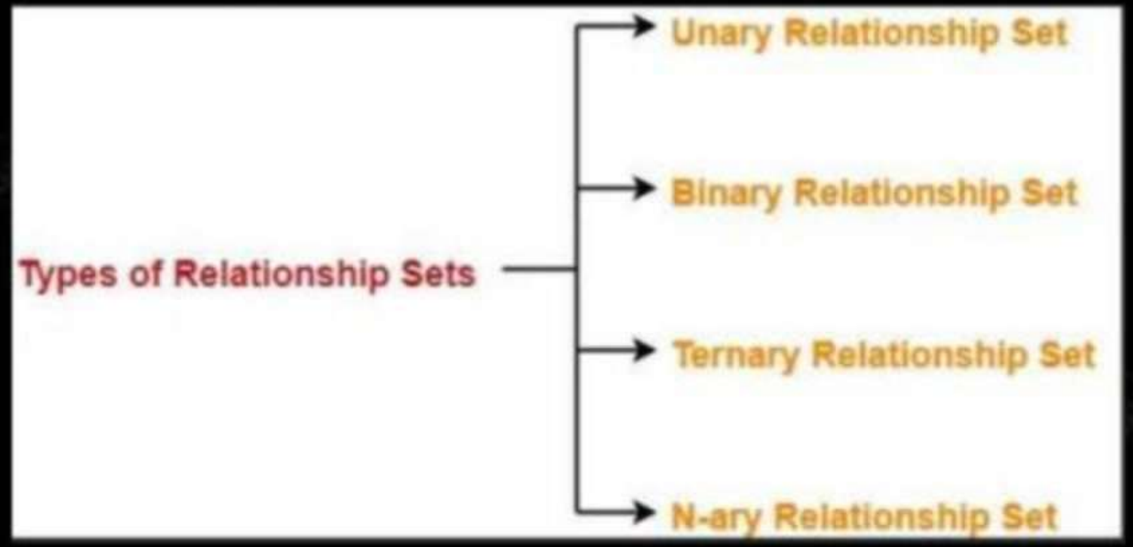
- A relationship set is a set of relationships of same type.
- Relationship set depicts S1 as enrolled in C2, S2 as enrolled in C1, and S3 as registered in C3.



Degree of a Relationship Set-

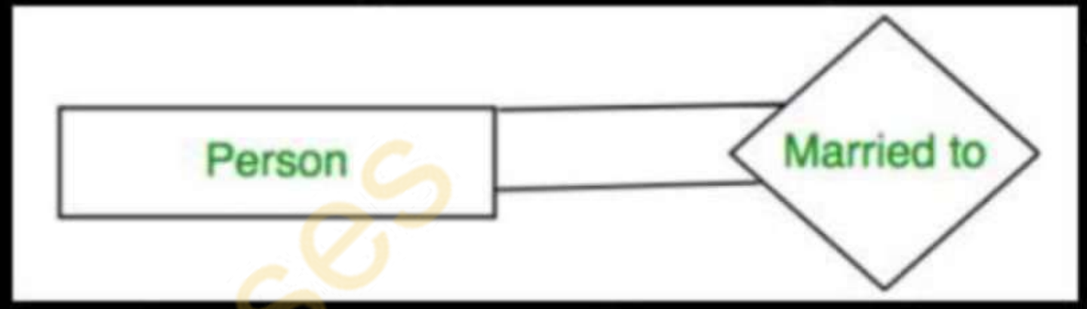
- The number of entity sets that participate in a relationship set is termed as the degree of that relationship set.

Types of Relationship Sets-



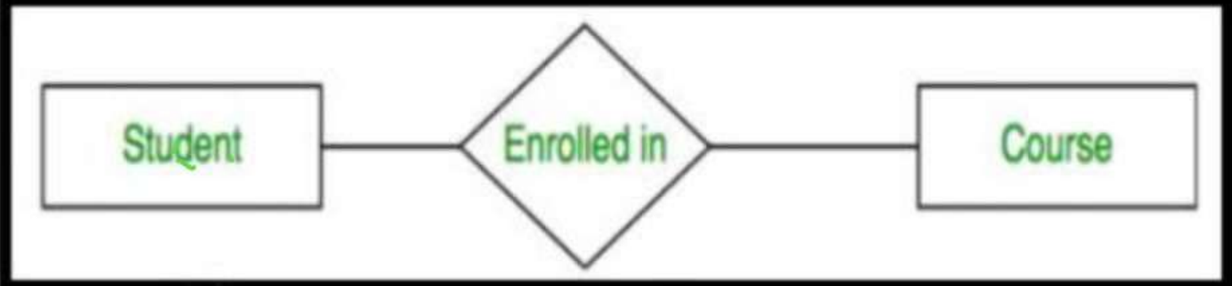
1. Unary Relationship: When there is only **ONE entity set** participating in a relation, the relationship is called a unary relationship.

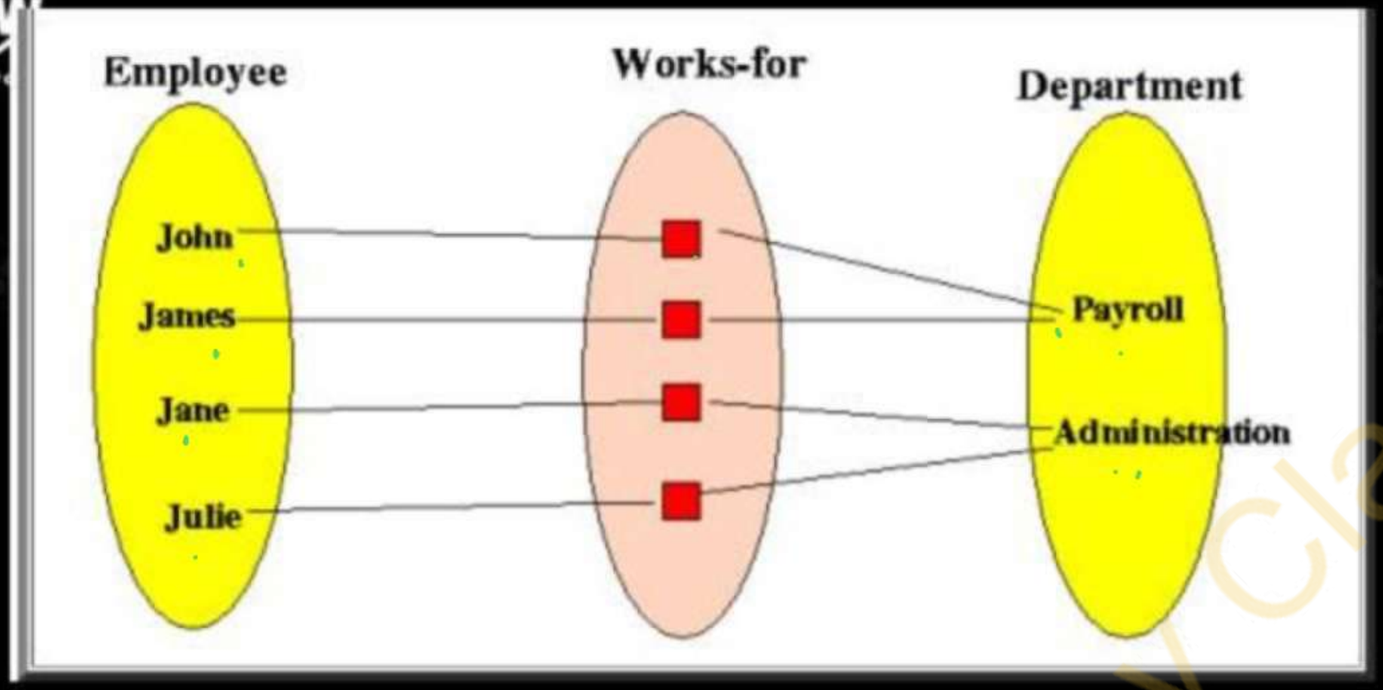
For example, **one person is married to only one person.**



2. Binary Relationship: When there are two entities set participating in a relationship, the relationship is called a binary relationship.

For example, **a Student is enrolled in a Course.**





3. Ternary Relationship Set-

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

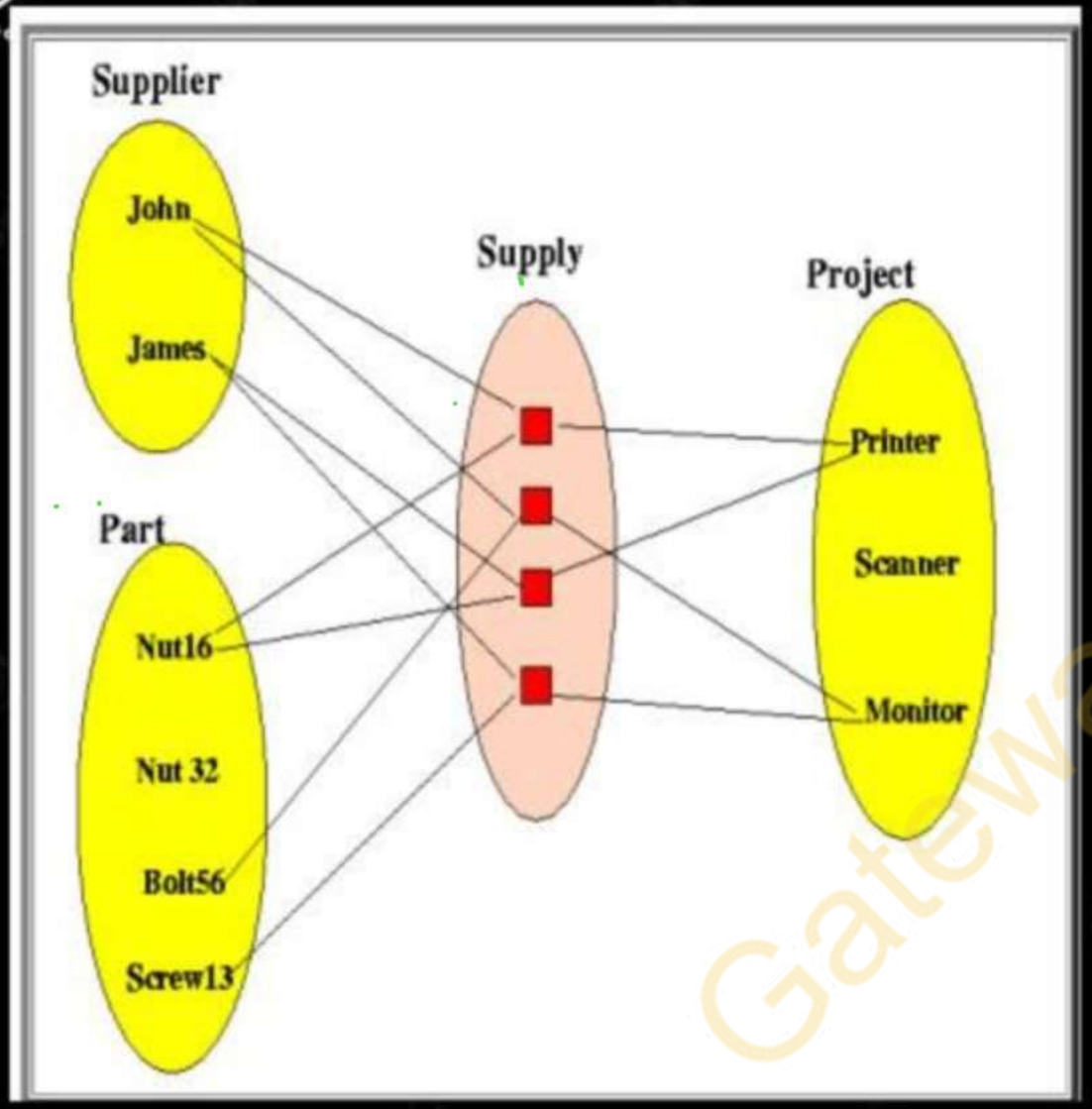
Supplier = {John, James}

Part = {Nut16, Nut32, Bolt56, Screw13}

Project = {Printer, Scanner, Monitor}

Relationship type: a *certain* supplier supplies a *certain* part to a *certain* project.

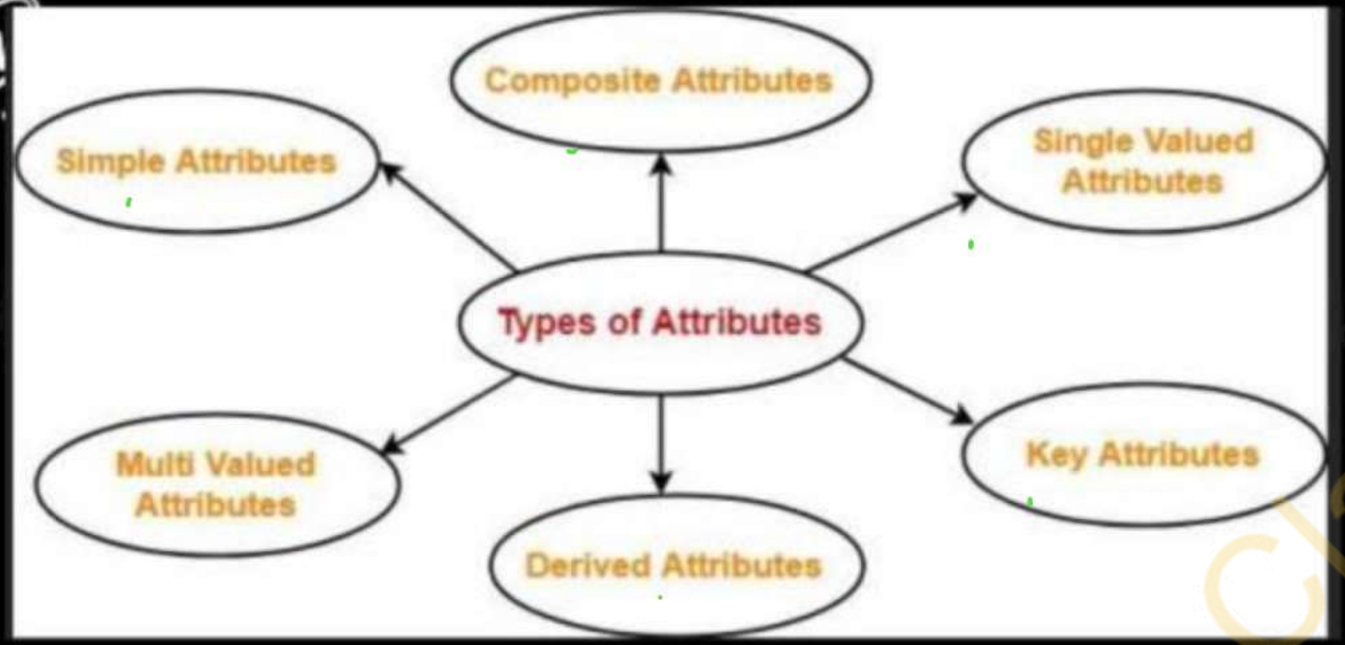
(John, Nut16, Printer) - represents the fact that John supplies the part "Nut16" to the project "Printer"



N-ary Relationship: When there are n entities participating in a relationship, the relationship is called an n-ary relationship

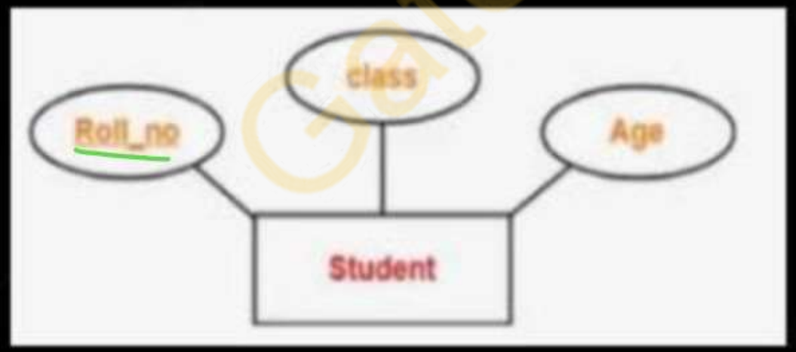
Attribute

- Attributes are the descriptive properties which are owned by each entity of an Entity Set..
- For example, Roll_No, Name, DOB, Age, Address, and Mobile_No are the attributes that define entity type Student.



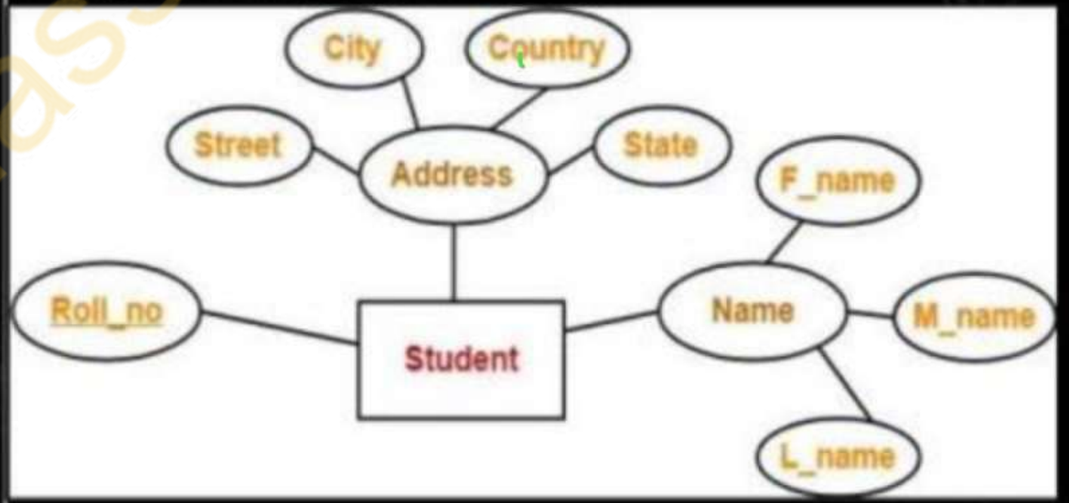
1. Simple Attributes-

Simple attributes are those attributes which can not be divided further.



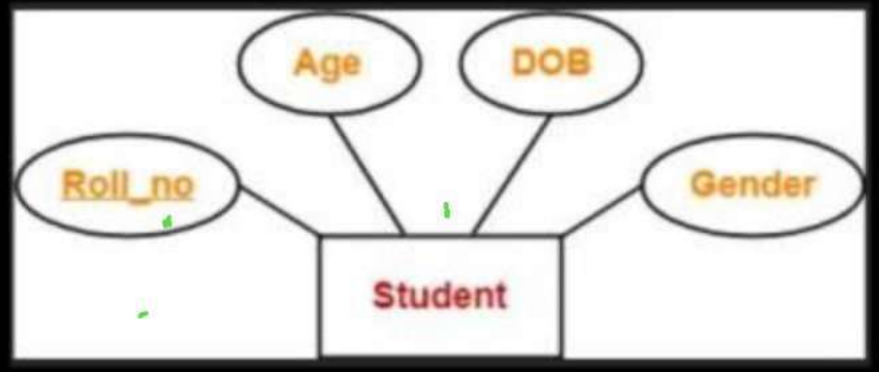
2. Composite Attributes-

Composite attributes are those attributes which are composed of many other simple attributes.



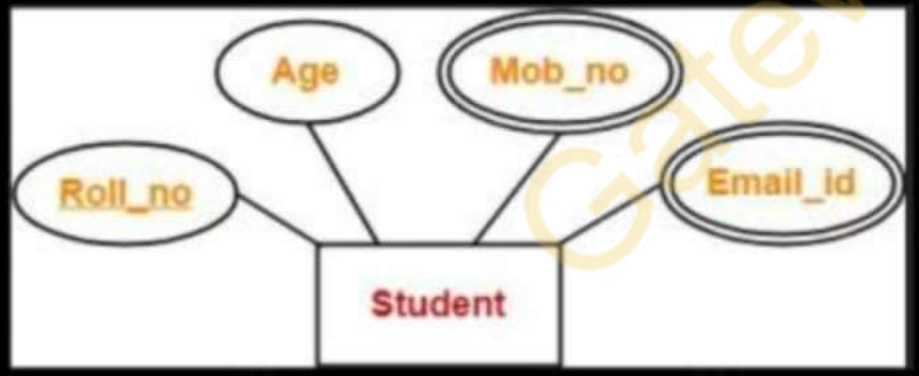
3. Single Valued Attributes-

Those attributes which can take only one value for a given entity from an entity set.



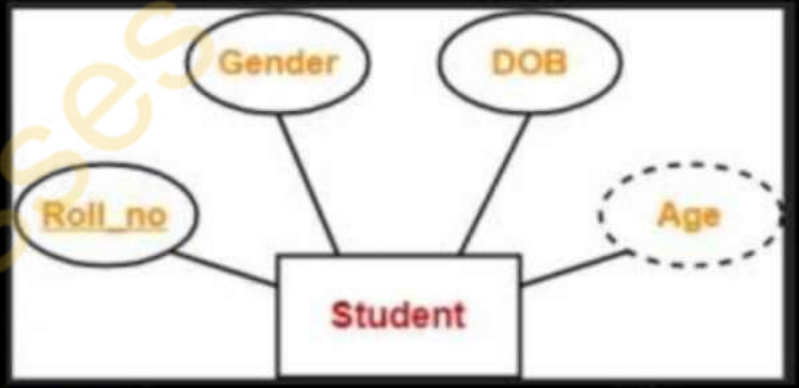
4. Multi Valued Attributes-

Those attributes which can take more than one value for a given entity from an entity set.



5. Derived Attributes-

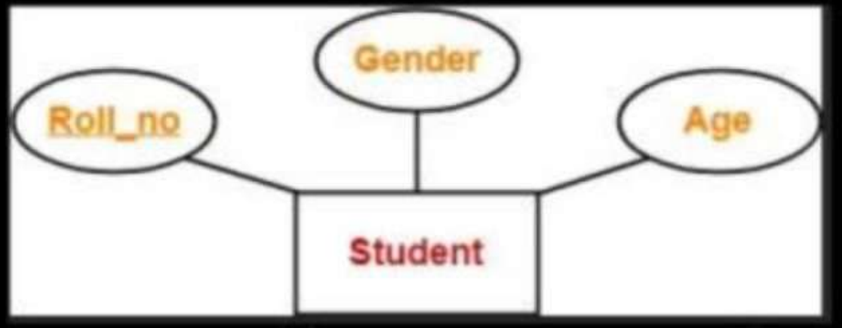
Those attributes which can be derived from other attribute.



6. Key Attributes-

Those attributes which can identify an entity uniquely in an entity set.

.



Constraints on Relationships

Relationships have a number of constraints

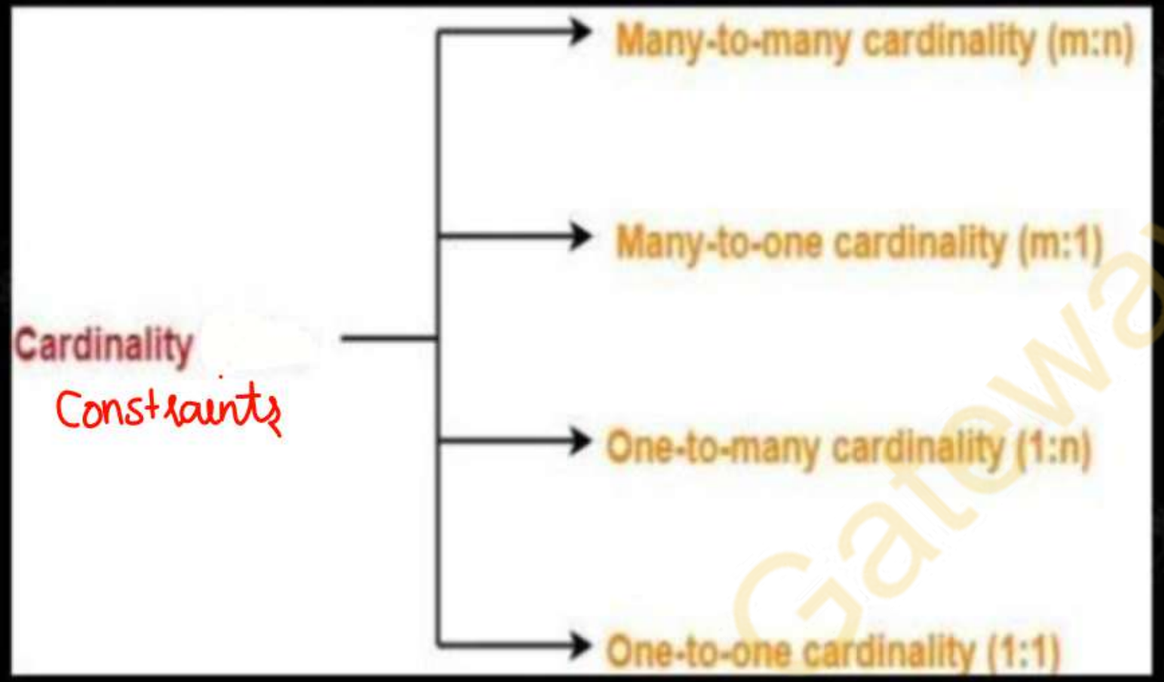
It is important to discover all constraints of the relationships because these constraints will help us in the design of "good" database schemas.

There are a two types of constraints:

- Cardinality ratio constraints
- Participation constraints

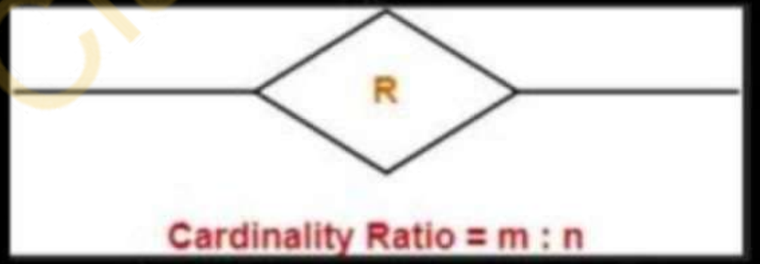
Cardinality Constraint-

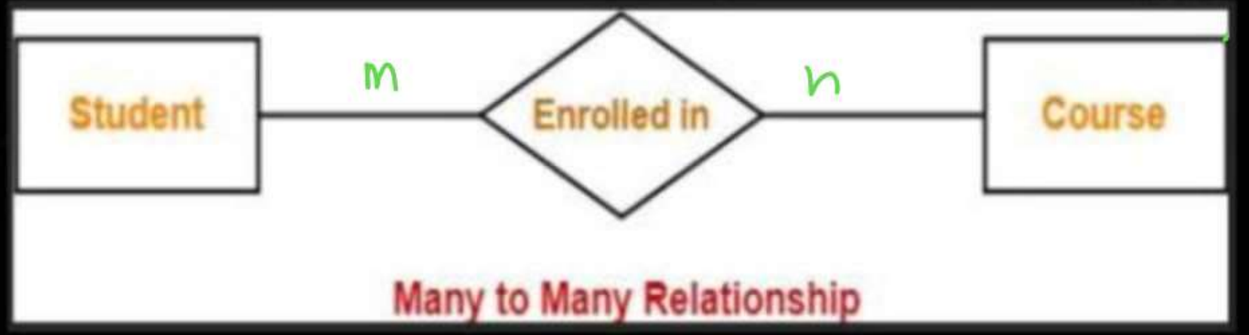
Cardinality constraint defines the maximum number of relationship instances in which an entity can participate.



1. Many-to-Many Cardinality-

- An entity in set A can be associated with any number (zero or more) of entities in set B.
- An entity in set B can be associated with any number (zero or more) of entities in set A



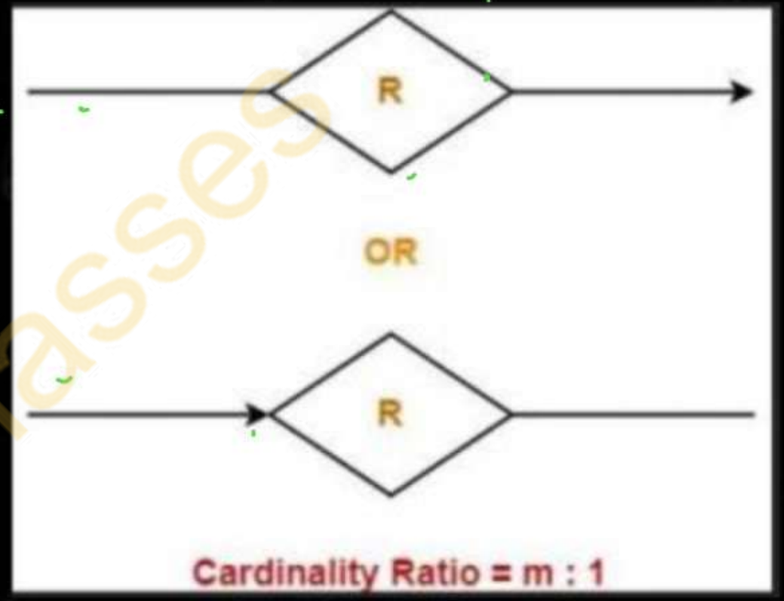


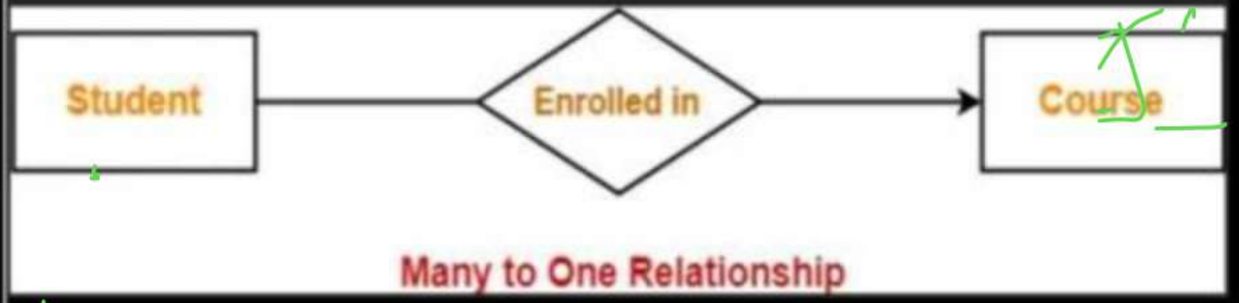
One student can enroll in any number (zero or more) of courses.

- One course can be enrolled by any number (zero or more) of students.

2. Many-to-One Cardinality-

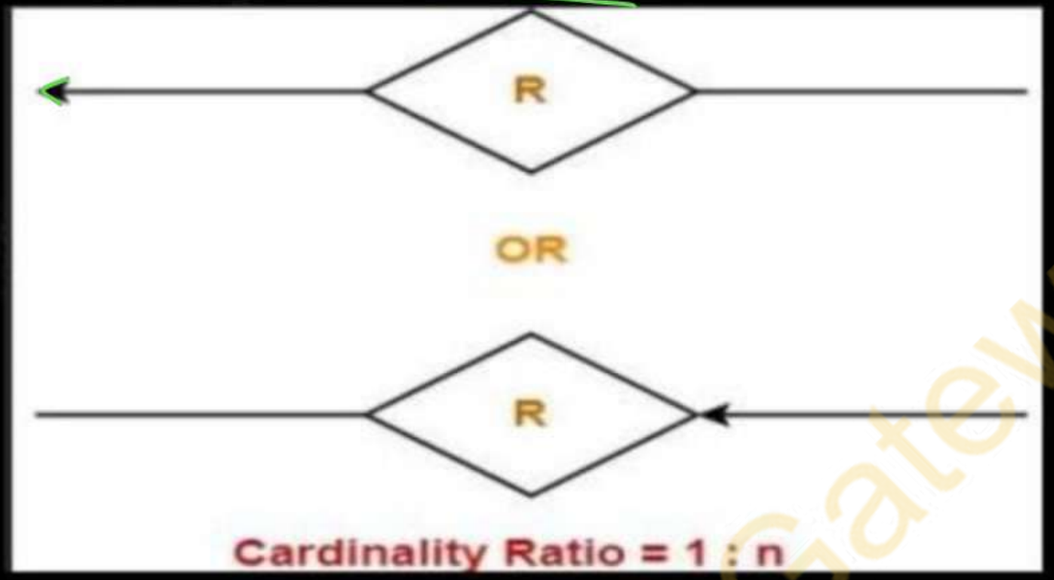
- An entity in set A can be associated with at most one entity in set B.
- An entity in set B can be associated with any number (zero or more) of entities in set A.
- One student can enroll in at most one course.
- One course can be enrolled by any number (zero or more) of students.

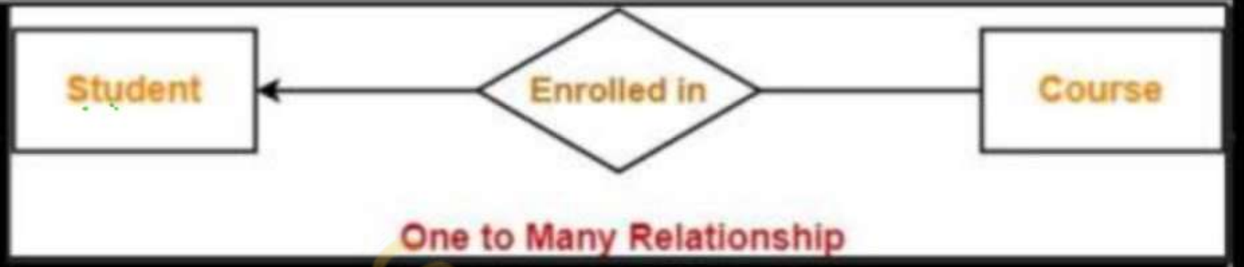




3. One-to-Many Cardinality-

- An entity in set A can be associated with any number (zero or more) of entities in set B.
- An entity in set B can be associated with at most one entity in set A.

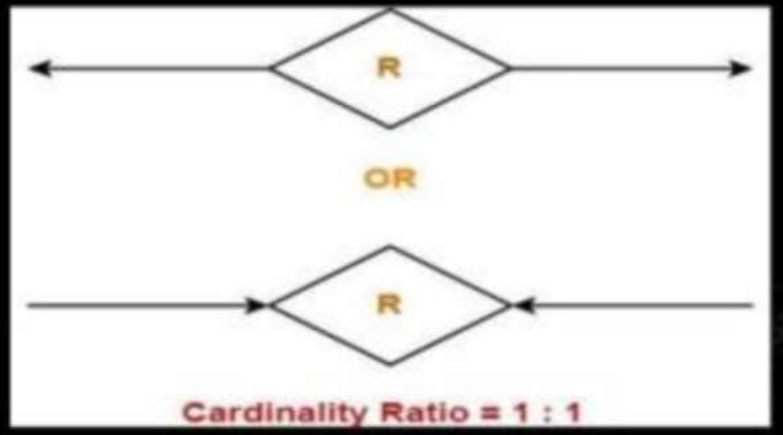


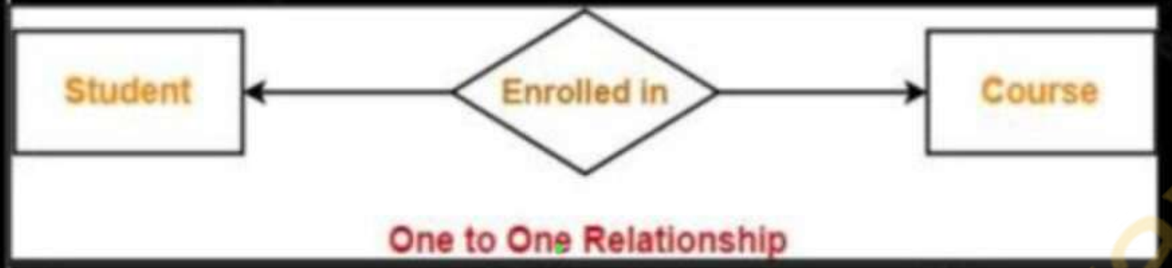


- One student can enroll in any number (zero or more) of courses.
- One course can be enrolled by at most one student.

4. One-to-One Cardinality-

- An entity in set A can be associated with at most one entity in set B.
- An entity in set B can be associated with at most one entity in set A.

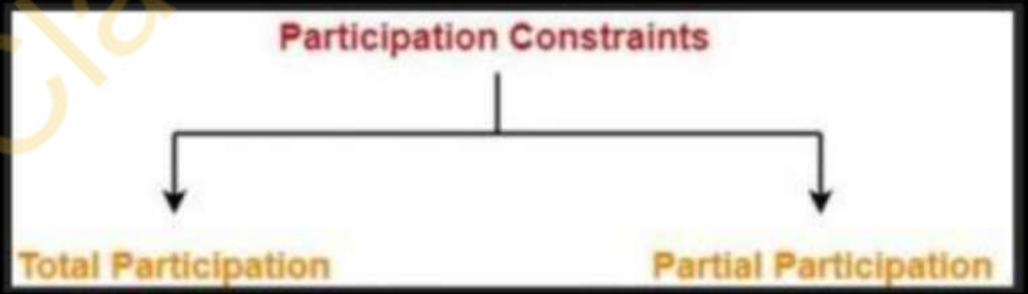




- One student can enroll in at most one course.
- One course can be enrolled by at most one student

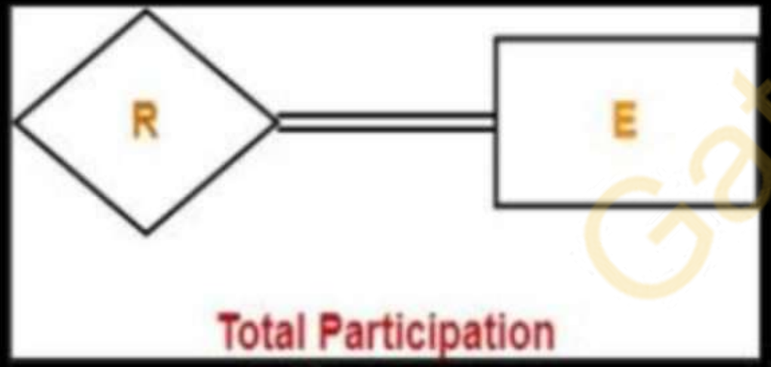
Participation Constraints-

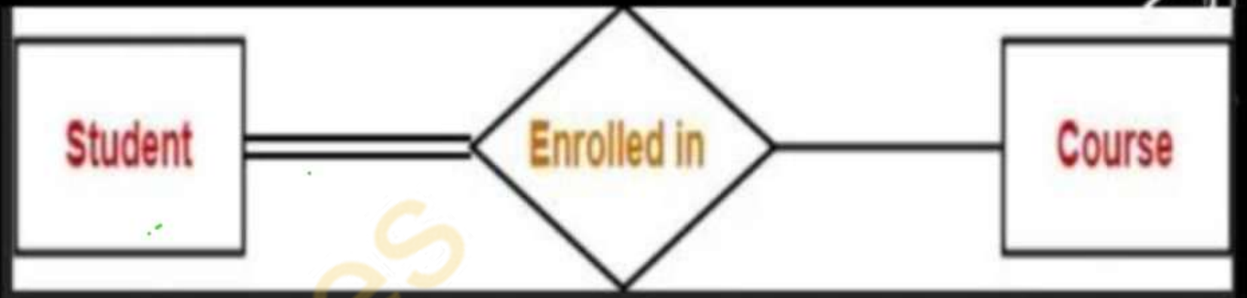
Participation constraints define the least number of relationship instances in which an entity must compulsorily participate.



1. Total Participation-

- It specifies that each entity in the entity set must compulsorily participate in at least one relationship instance in that relationship set.
- That is why, it is also called as mandatory participation.

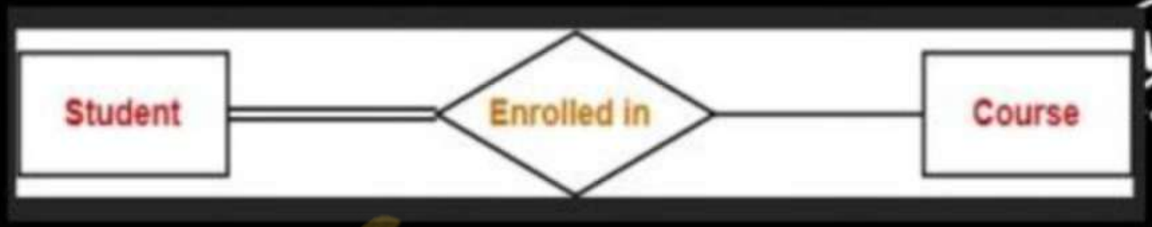




- It specifies that each student must be enrolled in at least one course.
- Total participation is represented using a double line between the entity set and relationship set.

2. Partial Participation-

- 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.



- Single line between the entity set "Course" and relationship set "Enrolled in" signifies partial participation.
- It specifies that there might exist some courses for which no enrollments are made.

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.

AKTU QUESTIONS		
Q.1	Distinguish between strong entity set with weak entity set .Draw an ER diagram to illustrate weak entity set.	AKTU 2022-23 AKTU 2019-20 AKTU 2018-19
Q.2	Explain ER diagram in detail.	AKTU 2020-21
Q.3	What is the key of a relation? Explain the difference between primary key, super key and candidate key.	AKTU 2015-16
Q.4	Give example of a simple , composite, attribute of an entity	AKTU2015-16

Database Management System

UNIT 1
INTRODUCTION

Today's Target

- Converting ER diagram to tables
- AKTU PYQs

By PRAGYA RAJVANSHI

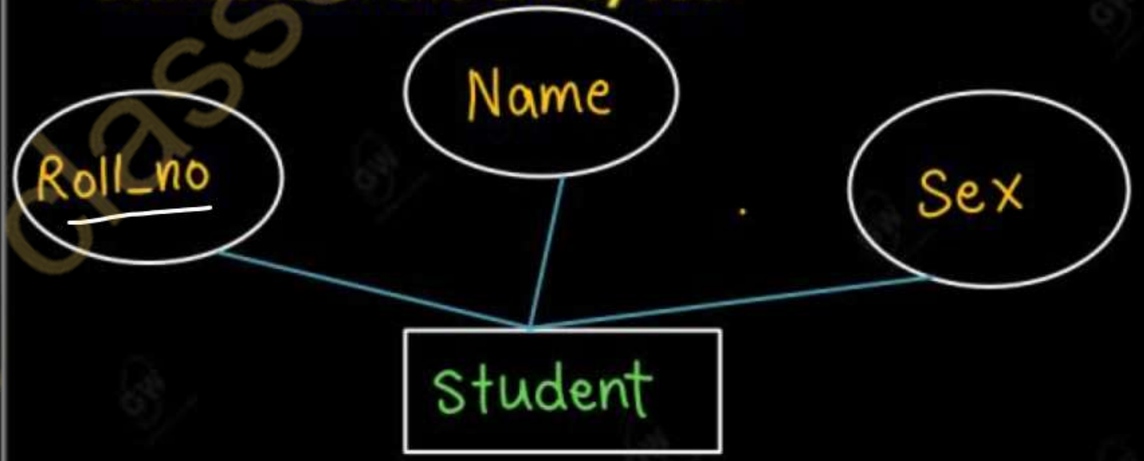
B.Tech, M.Tech(C.S.E.)

Converting ER Diagrams to Tables-

- ER diagram is converted into the tables in relational model.
- This is because relational models can be easily implemented by RDBMS like MySQL , Oracle etc.

Rule-01: For Strong Entity Set With Only Simple Attributes-

- A strong entity set with only simple attributes will require only one table in relational model.
- Attributes of the table will be the attributes of the entity set.
- The primary key of the table will be the key attribute of the entity set.



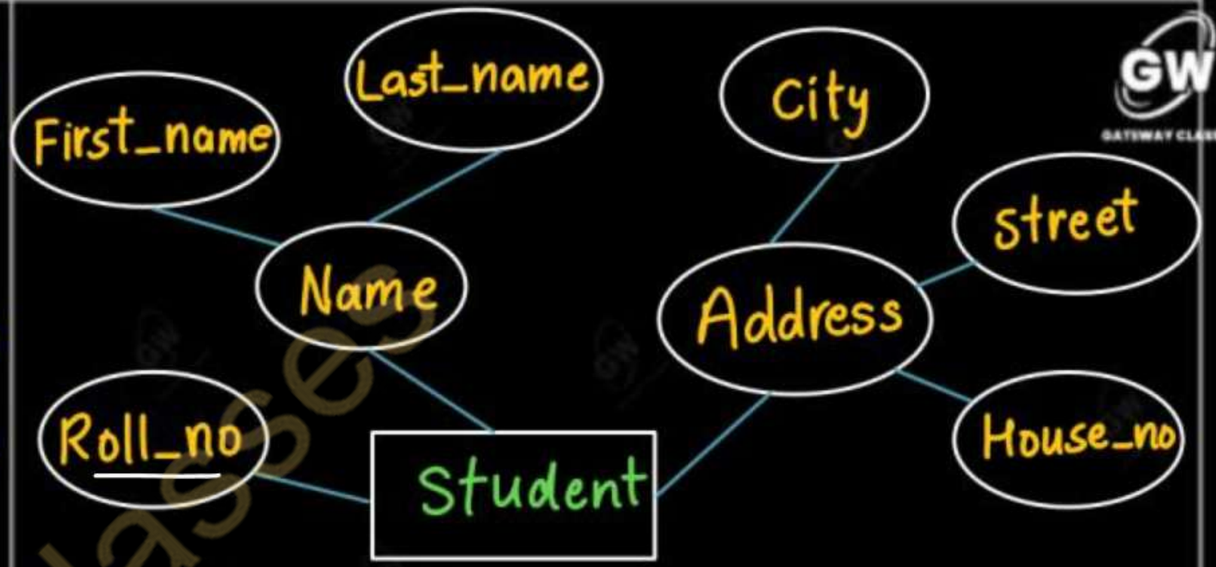
Student

Roll_no	Name	Sex

Schema : Student (Roll_no , Name , Sex)

Rule-02: For Strong Entity Set With Composite Attributes-

- A strong entity set with any number of composite attributes will require only one table in relational model.
- While conversion, simple attributes of the composite attributes are taken into account and not the composite attribute itself.



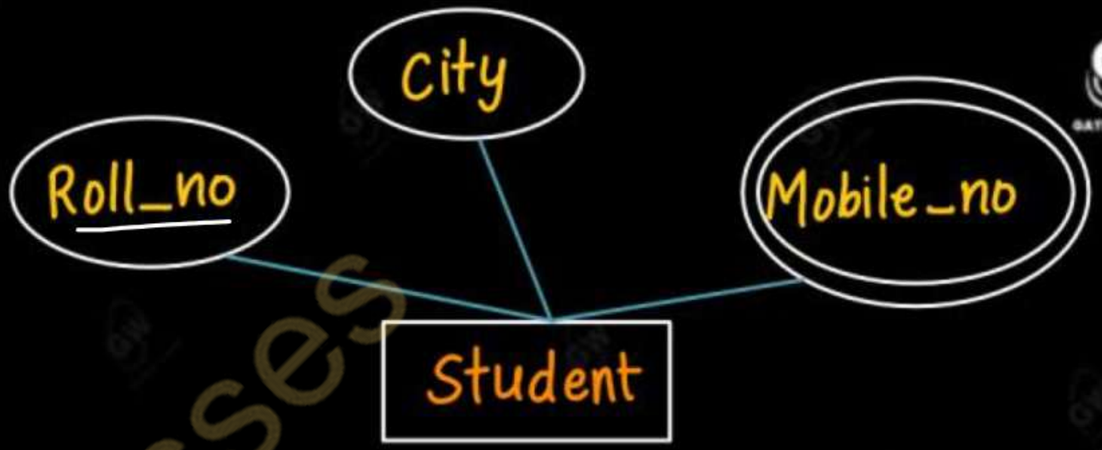
Student

Roll_no	First_name	Last_name	House_no	Street	City

Schema : Student (Roll_no , First_name , Last_name , House_no , Street , City)

Rule-03: For Strong Entity Set With Multi Valued Attributes-

- A strong entity set with any number of multi valued attributes will require two tables in relational model.
- One table will contain all the simple attributes with the primary key.
- Other table will contain the primary key and all the multi valued attribute



Roll_no	City

Roll_no	Mobile_no

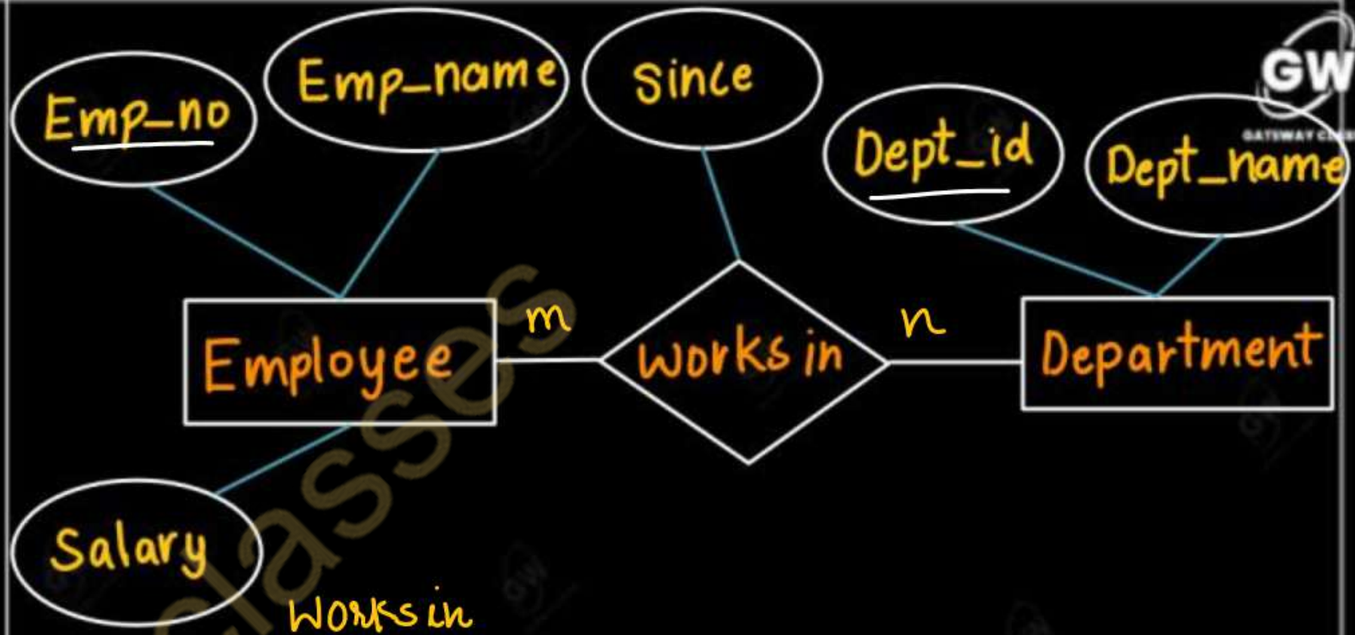
Rule-04: Translating

Relationship Set into a Table-

- A relationship set will require one table in the relational model.
- Attributes of the table are-

1. Primary key attributes of the participating entity sets
2. Its own descriptive attributes if any.

- Set of non-descriptive attributes will be the primary key.



Works in

Emp_no	Dept_id	since

Schema : Works in (Emp_no , Dept_id , since)

NOTE-

If we consider the overall ER diagram, three tables will be required in relational model-

- One table for the entity set "Employee"
- One table for the entity set "Department"
- One table for the relationship set "Works in"

Employee

Emp_no	Emp_name	salary

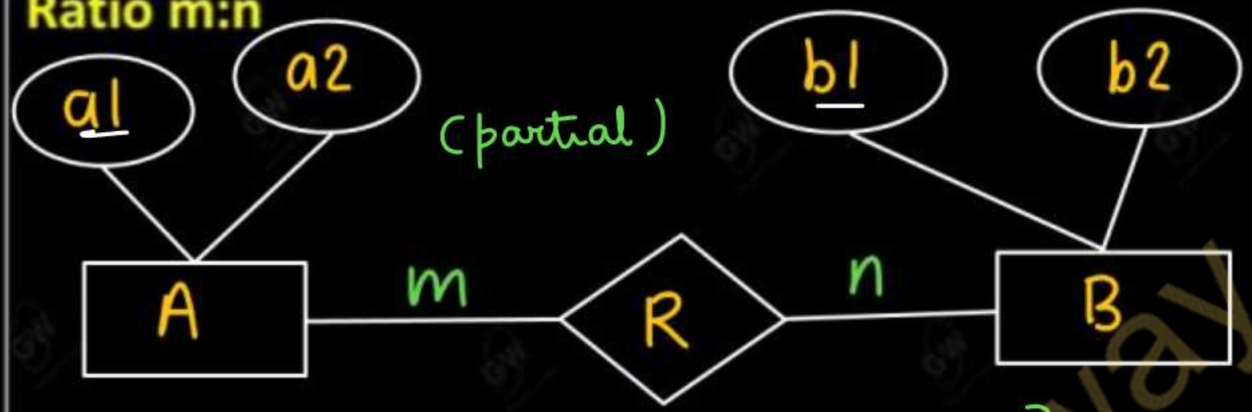
Department

Dept_id	Dept_name

Rule-05: For Binary Relationships

With Cardinality Ratios-

Case-01: For Binary Relationship With Cardinality Ratio m:n



Three tables will be required-

A (a1 , a2)

R (a1 , b1)

B (b1 , b2)

A

a1	a2

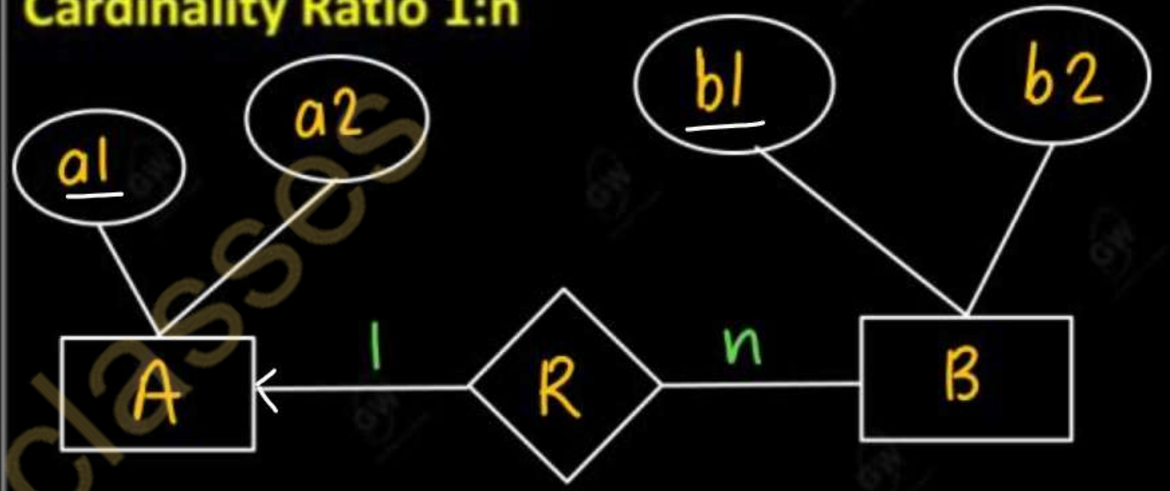
B

b1	b2

R

a1	b1

Case-02: For Binary Relationship With Cardinality Ratio 1:n



Two tables will be required-

A (a1 , a2)

BR (a1 , b1 , b2)

A

a1	a2

BR

b1	b2	a1

Case-03: For Binary Relationship With Cardinality Ratio m:1

n : 1

Here, two tables will be required-

AR (a1 , a2 , b1)

B (b1 , b2)

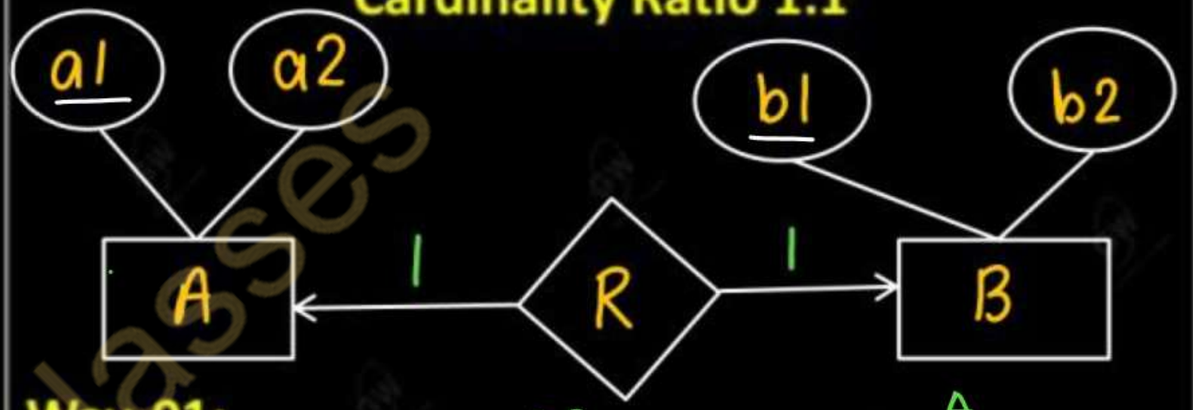
AR

a1	a2	b1

B

b1	b2

Case-04: For Binary Relationship With Cardinality Ratio 1:1



Way-01:

AR (a1 , a2 , b1)

B (b1 , b2)

AR

a1	a2	b1

B

b1	b2

A

a1	a2

BR

b1	b2	a1

Way-02:

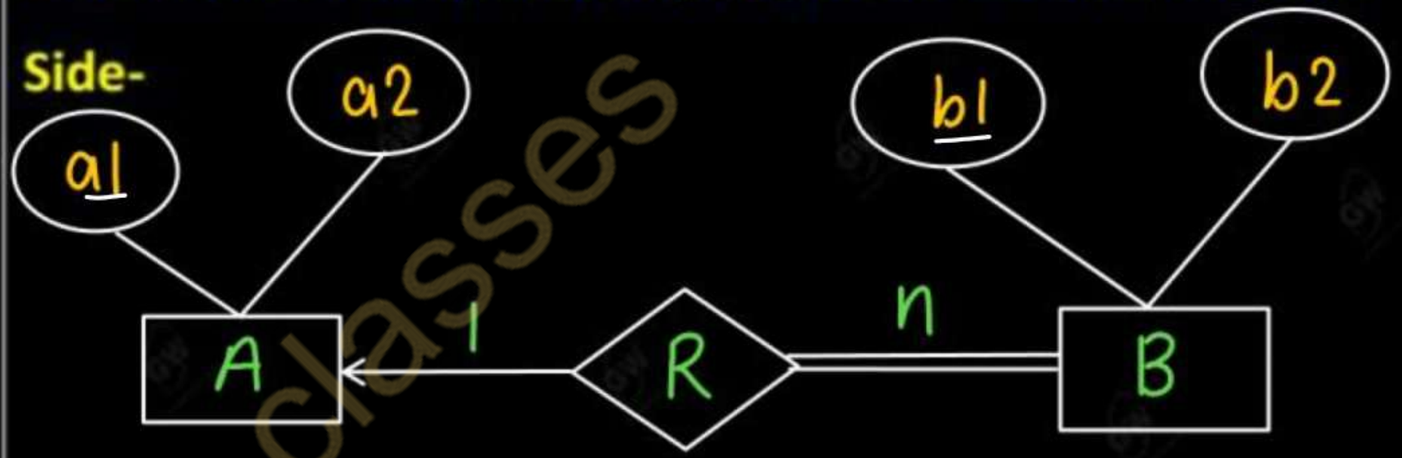
A (a1 , a2)

BR (a1 , b1 , b2)

Rule-06: For Binary Relationship With Both Cardinality Constraints and Participation Constraints-

- Cardinality constraints will be implemented as discussed in Rule-05.
- Because of the total participation constraint, foreign key acquires NOT NULL constraint i.e. now foreign key can not be null.

Case-01: For Binary Relationship With Cardinality Constraint and Total Participation Constraint From One Side-



Then, two tables will be required-

A (a1 , a2)

BR (a1 , b1 , b2)

A

a1	a2

BR

a1	b1	b2

Because of total participation, foreign key a1 has acquired NOT NULL constraint, so it can't be null now.

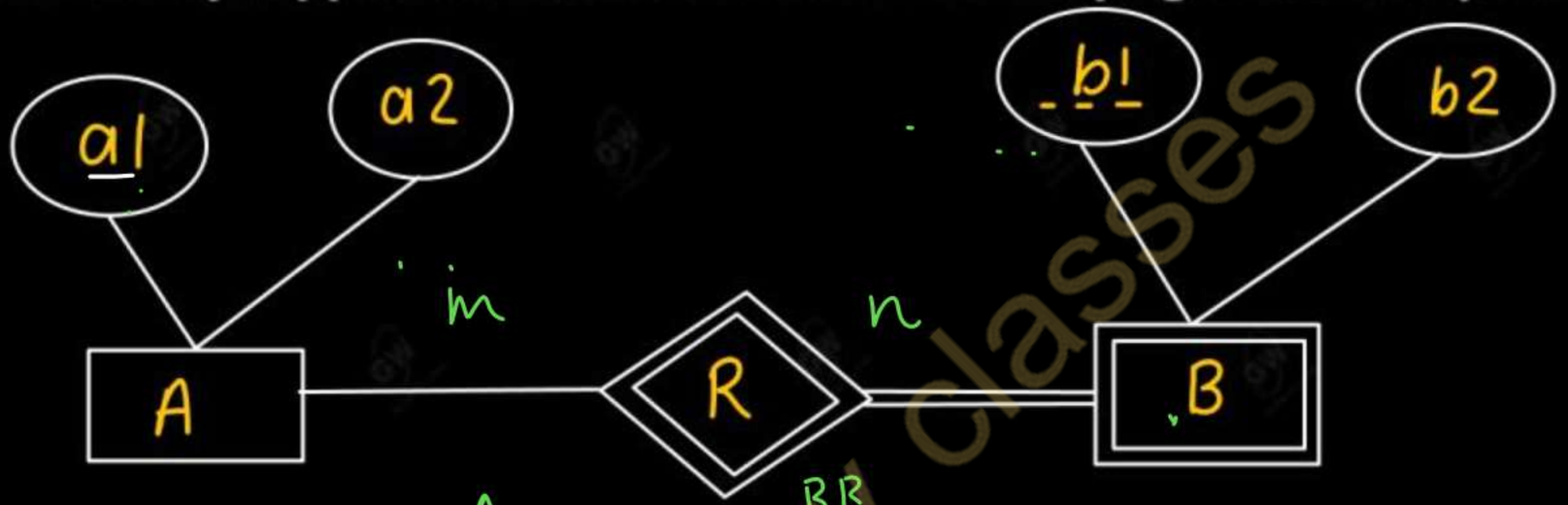
Case-02: For Binary Relationship With Cardinality Constraint and Total Participation Constraint From Both Sides-

- If there is a key constraint from both the sides of an entity set with total participation, then that binary relationship is represented using only single table.

ARB (a1 , a2 , b1 , b2)

. Rule-07: For Binary Relationship With Weak Entity Set-

Weak entity set always appears in association with identifying relationship with total participation constraint.

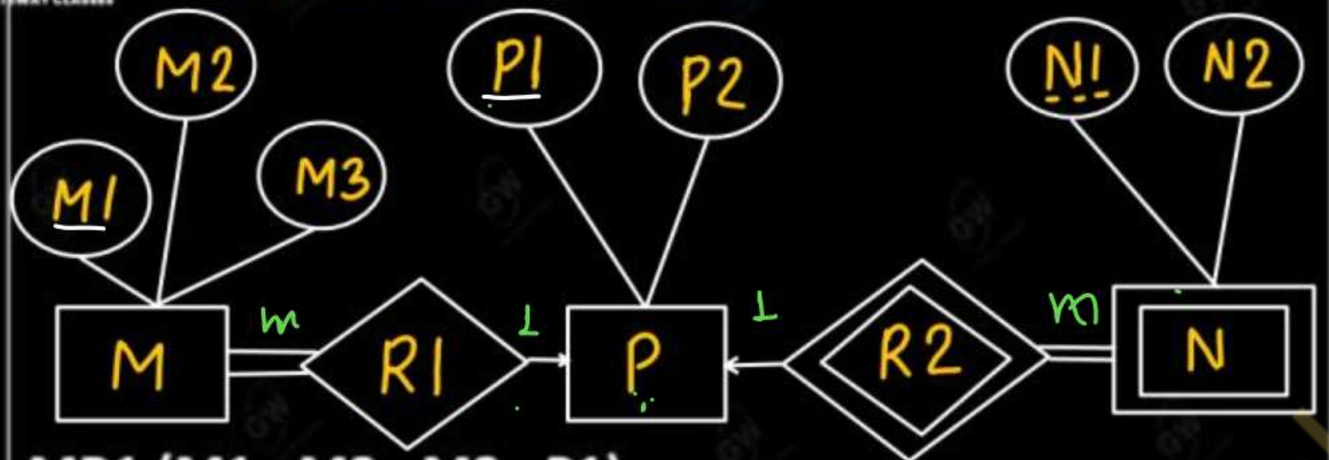


Two tables will be required-

A (a1 , a2)

BR (a1 , b1 , b2)

Practice problem 1:



MR1 (M1 , M2 , M3 , P1)

P (P1 , P2)

NR2 (P1 , N1 , N2)

MR1

M1	M2	M3	P1

P1 which is a foreign key Can't be NULL

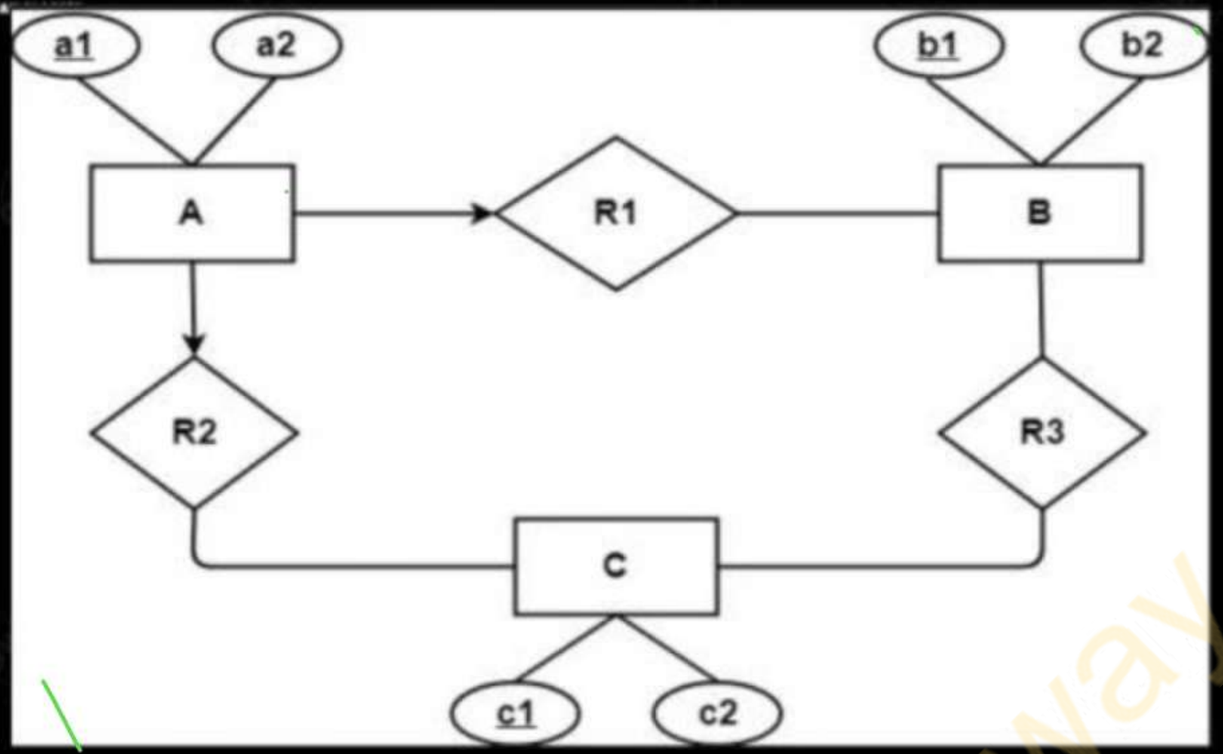
P1	P2

P

NR2

N1	P1	N2

Pratice Problem2



AR1R2 (a1 , a2 , b1 , c1)

B (b1 , b2)

C (c1 , c2)

R3 (b1 , c1)

AR1R2

a1	a2	b1	c1

B

b1	b2

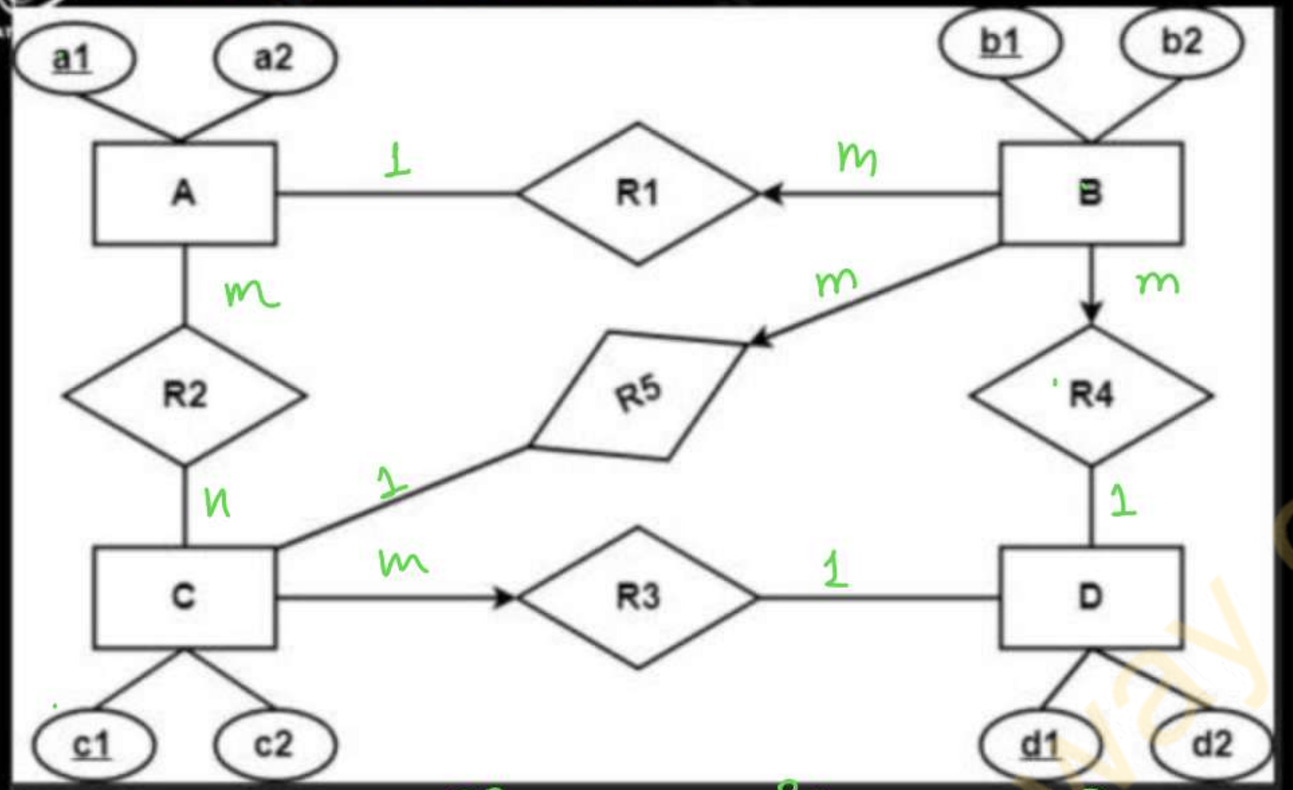
C

c1	c2

R3

b1	c1

Pratice Problem 3



A

a1	a2

CR3

c1	c2	d1

R2

a1	c1

D

d1	d2

BR1R4R5 (b1 , b2 , a1 , c1 , d1)

A (a1 , a2)

R2 (a1 , c1)

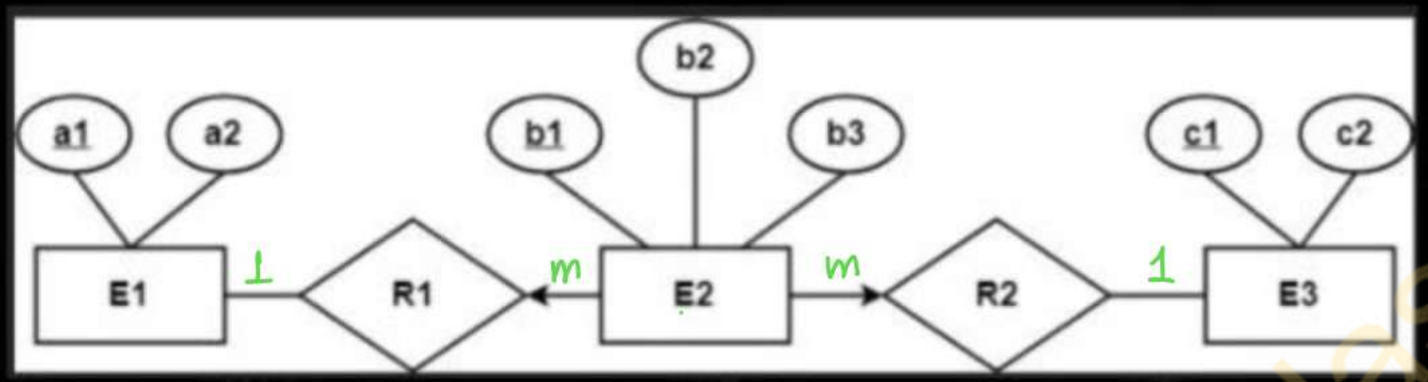
CR3 (c1 , c2 , d1)

D (d1 , d2)

BR1R4R5

b1	b2	a1	d1	c1

Pratice Problem 4



E1 (a1 , a2)

E2R1R2 (b1 , b2 , a1 , c1 , b3)

E3 (c1 , c2)

E1

a1	a2

E2 R1 R2

b1	b2	b3	a1	c1

E3

c1	c2

Database Management System

UNIT 1
INTRODUCTION

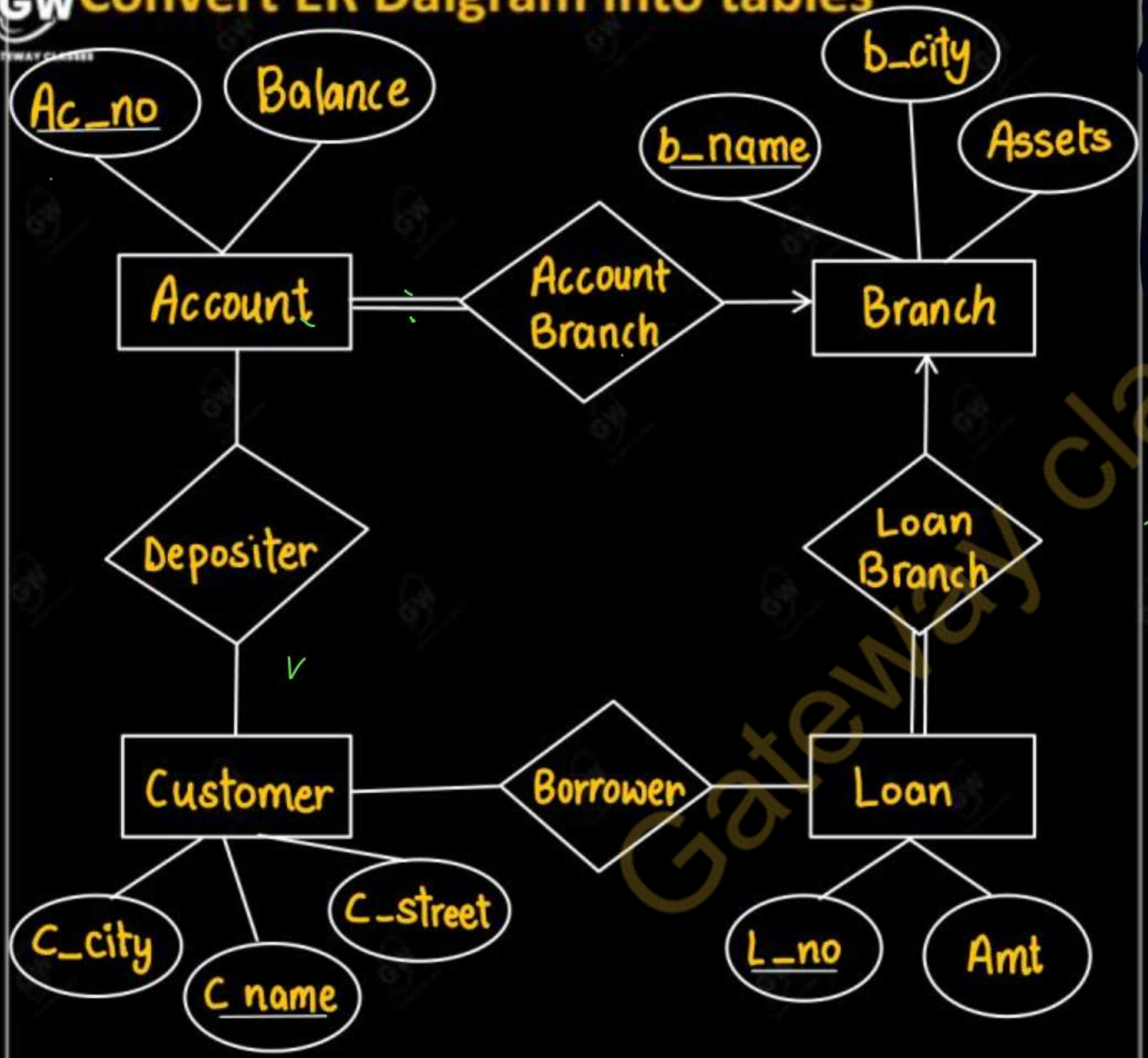
Today's Target

- Converting ER diagram to tables
- AKTU PYQs

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

Convert ER Daigram into tables



Account

Ac_no	Balance	b_name

Account (Ac_no, Balance, b_name)

Branch (b_name, b_city, Assests)

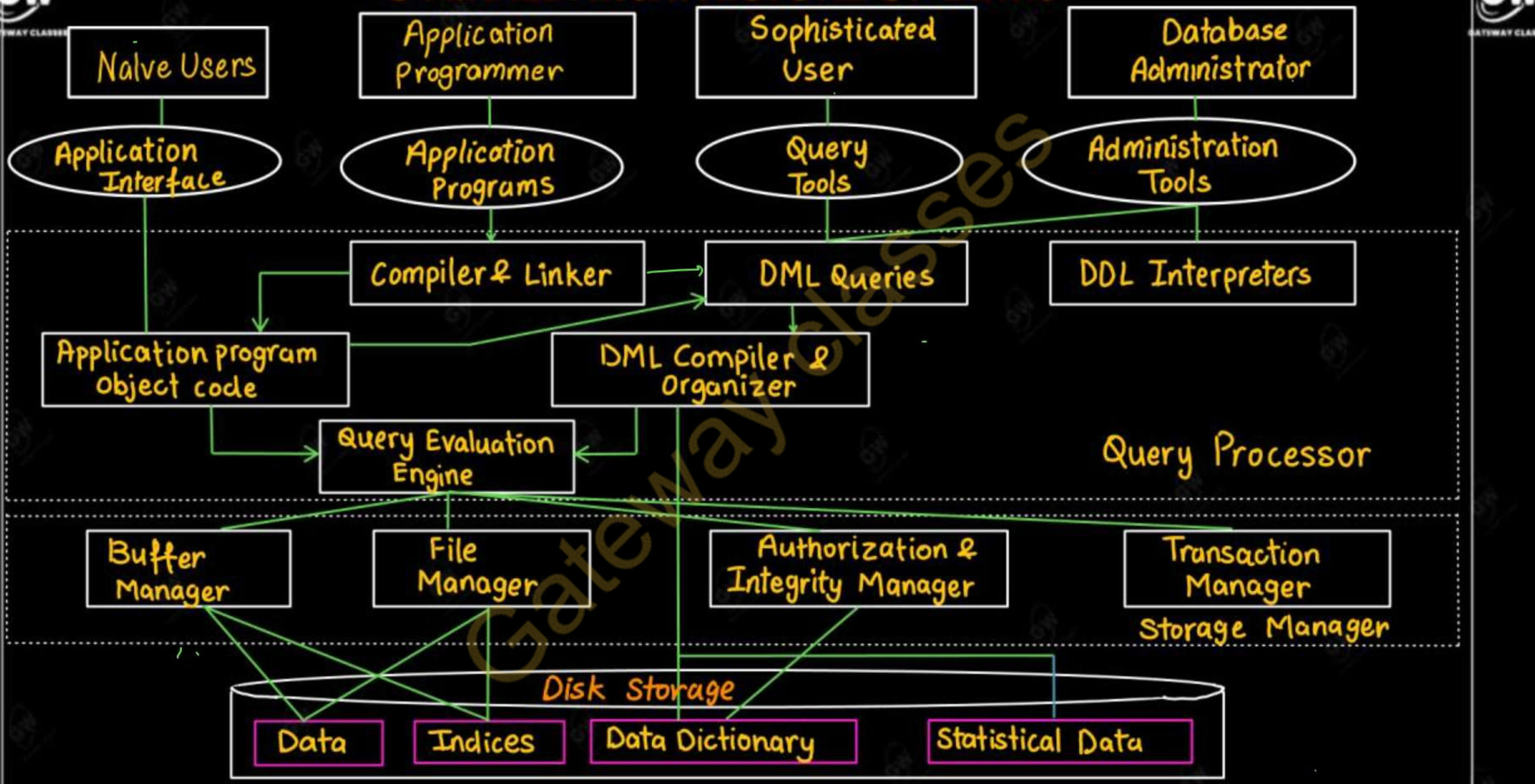
Loan (b_name, L_no, Amt)

Customer (C_city, C_name, C_street)

Borrower (C_name, L_no)

Depositer (C_name, Ac_no)

OVERALL ARCHITECTURE OF DBMS



The typical structure of DBMS is based on Relational data model.

- The top part of the architecture shows application interfaces used by naive users, application programs created by application programmers, query tools used by sophisticated users and administration tools used by database administrator.
- The lowest part of the architecture is for disk storage.
- The Middle two parts(Query processor and storage manager) are important components of database architecture

1Query Processor:

The query processor will accept query from user and solves it by accessing the database.

1.1.DDL (Data Definition Language) interpreter

Interprets DDL statements and store the definitions in the data dictionary, as metadata

- It processes and executes DDL commands like CREATE, ALTER, and DROP that define or modify database schema structures (tables, indexes, views).
- It parses these commands and translates them into the database's internal actions to create or alter objects in the database.

1.2.DML (Data Manipulation Language) compiler and organizer

- It is compiler that processes DML commands like SELECT, INSERT, UPDATE, and DELETE.
- It converts these high-level commands into low-level operations for accessing or modifying data in the database.
- The DML organizer refers to the System that arranges and optimizes the execution of these translated commands, ensuring efficient data retrieval and manipulation within the database.

1.3. Query evaluation engine

- It executes database queries by interpreting and optimizing SQL commands
- It works with the query optimizer to choose the best execution plan, retrieves data from storage, and processes the results efficiently based on the query.

1.4. Application Program Object Code

- It converts DML statements embedded in an application to normal procedure calls in the host language.

1.5.DML Queries

- Sophisticated or Specialized user's requests in a database query language.
- Each query is submitted to a query tools
- Query tools break down it into DML queries and low-level instructions.
- These are sent to the DML compiler and Organizer, and the query evaluation engine for further processing.

1.6. Compiler and Linker: -

- Application programmer writes program application.
- The source codes compiled by the compiler and linker-linked application program object code to DML queries and send to query evaluation engine.

2.Storage manager:

- Storage manager is the component of database system that provides interface between the low level data stored in the database and the application programs and queries submitted to the system.
- The storage manager is responsible for storing, retrieving, and updating data in the database. The storage manager components include-

2.1.Authorization and integrity manager: Validates the users who want to access the data and tests for integrity constraints.

2.2.Transaction manager: Ensures that the database remains in consistent despite of system failures and concurrent transaction execution proceeds without conflicting.

2.3.File manager: Manages allocation of space on disk storage and representation of the information on disk.

2.4. Buffer manager: Manages the fetching of data from disk storage into main memory. The buffer manager also decides what data to cache in main memory. Buffer manager is a crucial part of database system.

Storage manager implements several data structures such as:

- **Data files:** Used for storing database itself.
- **Data dictionary:** Used for storing metadata, particularly schema of database.
- **Indices:** Indices are used to provide fast access to data items present in the database.

Statistical Data – Records

DBMS Interface

A database management system (DBMS) interface is a user interface that allows the user to input queries to a database without using the query language itself.

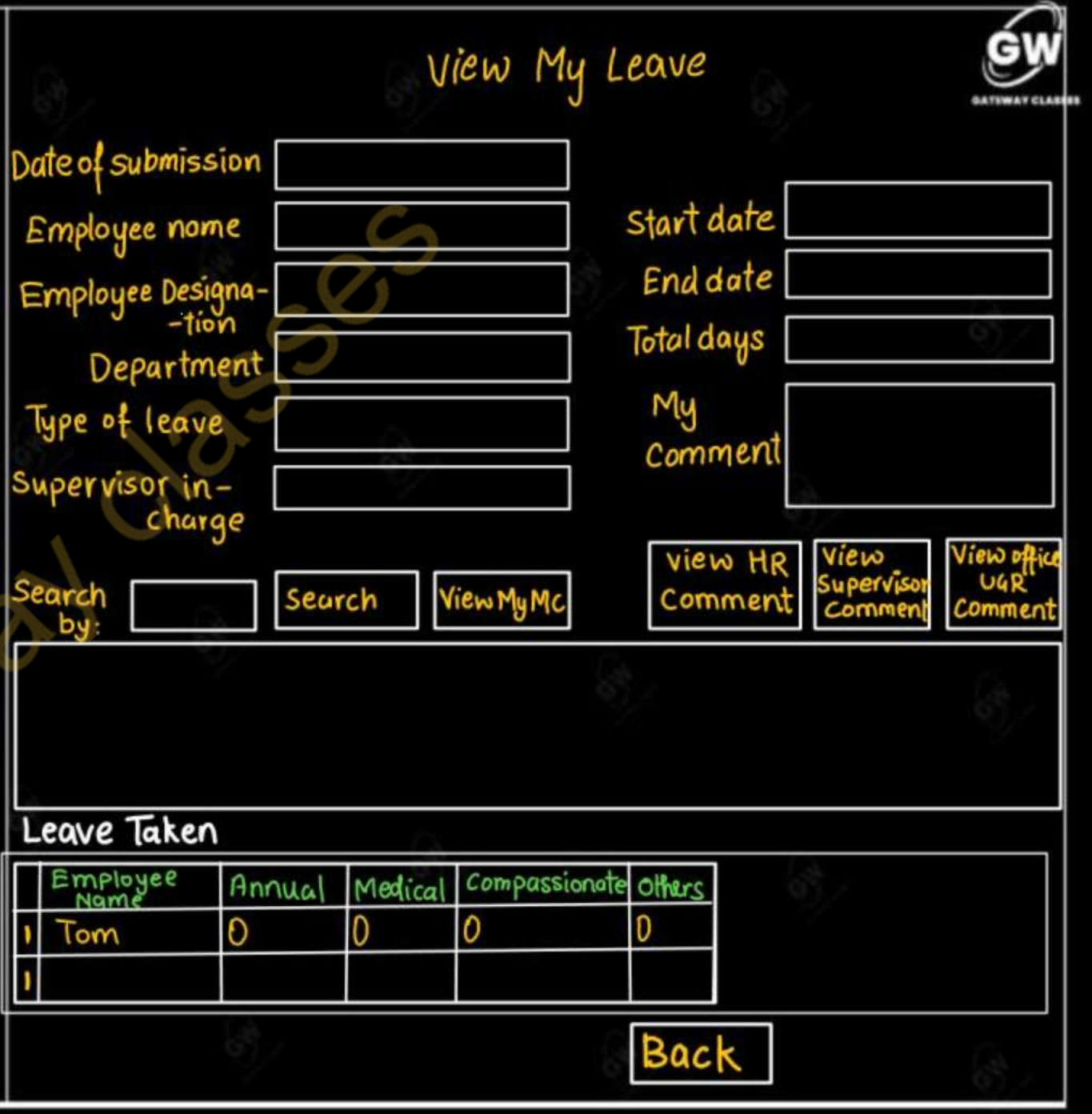
1. Menu-Based Interfaces

Seating prefrence

- These interfaces present the user with lists of options (called menus) that lead the user through the formation of a request
- The basic advantage of using menus is that they remove the tension of remembering specific commands and syntax of any query language.
- The query is basically composed step by step by collecting or picking options from a menu that is shown by the system.
- Pull-down menus are a very popular technique in Web-based interfaces. They are also often used in browsing interfaces which allow a user to look through the contents of a database in an exploratory and unstructured manner.

2.Forms-Based Interfaces

- A forms-based interface displays a form to each user.
- users can either fill out all the fields in a form to insert new data into a database or just fill in some fields. If some fields are left blank, the database system (DBMS) will automatically fill those blank fields with default values or similar data of the same type.
- These types of forms are usually designed or created and programmed for users that have no expertise.



3. Graphical User Interface

- A GUI typically displays a schema to the user in diagrammatic form.
- The user then can specify a query by manipulating the diagram. In many cases, GUI utilize both menus and forms.
- Most GUI use a pointing device such as a mouse, to pick a certain part of the displayed schema diagram

4. Natural Language Interface

- It allows users to make requests in plain English (or another language).
- The natural language interface refers to the words in its schema as well as to the set of standard words in a dictionary to interpret the request.
- If the interpretation is successful, the interface generates a query corresponding to the natural language and submits it to the DBMS for processing,
- otherwise, a dialogue is started with the user to clarify any provided condition or request. The main disadvantage of this is that the capabilities of this type of interface are not that advance.

5. Speech Input and Output Interfaces

- Speech is becoming more common for asking questions and getting answers in some applications.
- For simple tasks like checking a phone directory, flight times, or bank information, people can use speech to interact with the system.
- The system listens for specific words, turns them into data, and uses that data to run a query.
- Similarly, the system can convert results from text or numbers into spoken words to give users an answer.

6. Interfaces for Database Administrators (DBA)

- Most database system contains privileged commands that can be used only by the DBA's staff.
- These include commands for creating accounts, setting system parameters, granting account authorization, changing a schema, and reorganizing the storage structures of databases.

DBMS	RDBMS
No relationship between data.	Data is stored in the form of tables which are related to each other
Normalization is not present.	Normalization is present.
DBMS does not support distributed database	RDBMS support distributed database.
It stores data in either a navigational or hierarchical form.	It uses a tabular structure where the headers are the column names, and the rows contain corresponding values.
It deals with small quantity of data.	It deals with large amount of data

DBMS	RDBMS
Data redundancy is common in this model.	Keys and indexes do not allow Data redundancy.
Security is less	More security measures provided.
It supports single user.	It supports multiple users.
Data fetching is slower for the large amount of data.	Data fetching is fast because of relational approach.
Low software and hardware necessities	Higher software and hardware necessities

AKTU QUESTION

Q1	Explain briefly the overall structure of DBMS and explain its components on brief.	AKTU 2018-19 AKTU 2019-20
Q2	Difference between DBMS and RDBMS	AKTU 2018-19

Database Management System

UNIT 1
INTRODUCTION

Today's Target

- Difference between Procedural and non-procedural DML
- Three schema architecture
- Types of users

By PRAGYA RAJVANSHI
B.Tech, M.Tech(C.S.E.)

PDF notes are available in the App (Link in Description)

Difference between-

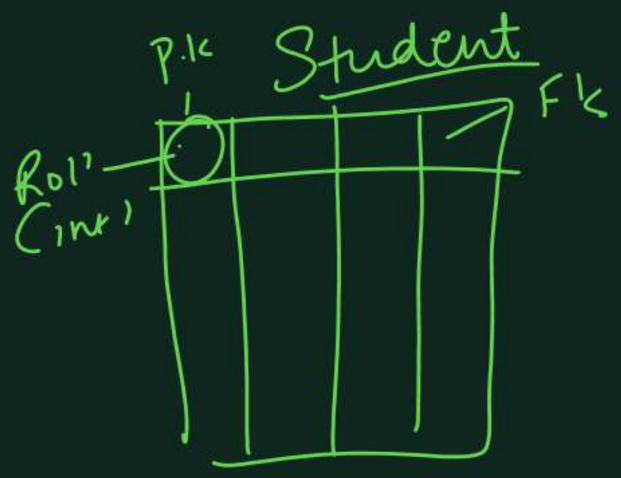
Feature	Procedural DML	Non-Procedural DML
Approach	Specifies *how* to retrieve data.	Specifies *what* data to retrieve.
Control over process	Full control over the retrieval process.	No control over how the query is executed.
Complexity	More complex, as it involves specifying steps.	Simpler, as only the result is described.
Examples	PL/SQL, T-SQL (with control structures).	SQL (SELECT, INSERT without procedures).
Performance	can be optimized manually by user.	System optimizes the query automatically.
Ease of use	Requires understanding of underlying data flow.	Easier for users with limited database knowledge.

List any four advantages of file system over database (2 Marks)

1. **Simplicity:** File systems are **easier to set up and use,** especially for simple data storage and retrieval tasks.
2. **Cost-effective:** File systems don't require **expensive software or maintenance,** making them a cheaper option.
3. **Low overhead:** File systems have **minimal resource requirements,** leading to better performance in small-scale application
4. **Direct file access:** Users have direct control over file storage and can manually manage data without complex queries

Metadata

- Metadata in DBMS is characterized as **data about data.**
- It describes the **context and information** about data, the **way that data** is stored and the **various relations among data.**
- Metadata in a relational DBMS stores **data about Constraints, Table Relationships, Data Types, Columns, Tables,** and so on.



Types of metadata in DBMS

1. Technical Metadata

- It stores information about the way in which data is stored in the Database, i.e. Data types, file locations, indexing, query performance. table name, ...

2. Business Metadata

- It contains information about a particular business and the way that the business may handle the data in the database.
- Information like ownership, Policies of handling data, business rules and regulations, etc

3.Operational metadata

- It refers to information that describes the processes and operations performed on data, such as data processing workflows, system performance metrics, and transformation rules.
- It helps track data usage, monitor system efficiency, and ensure operational transparency within a system.

4. Descriptive metadata

It refers to data that provides detailed information about the database's contents.

List some problem with file system

1.Data Redundancy: Multiple copies of the same data can exist, leading to inconsistency and unnecessary storage usage.

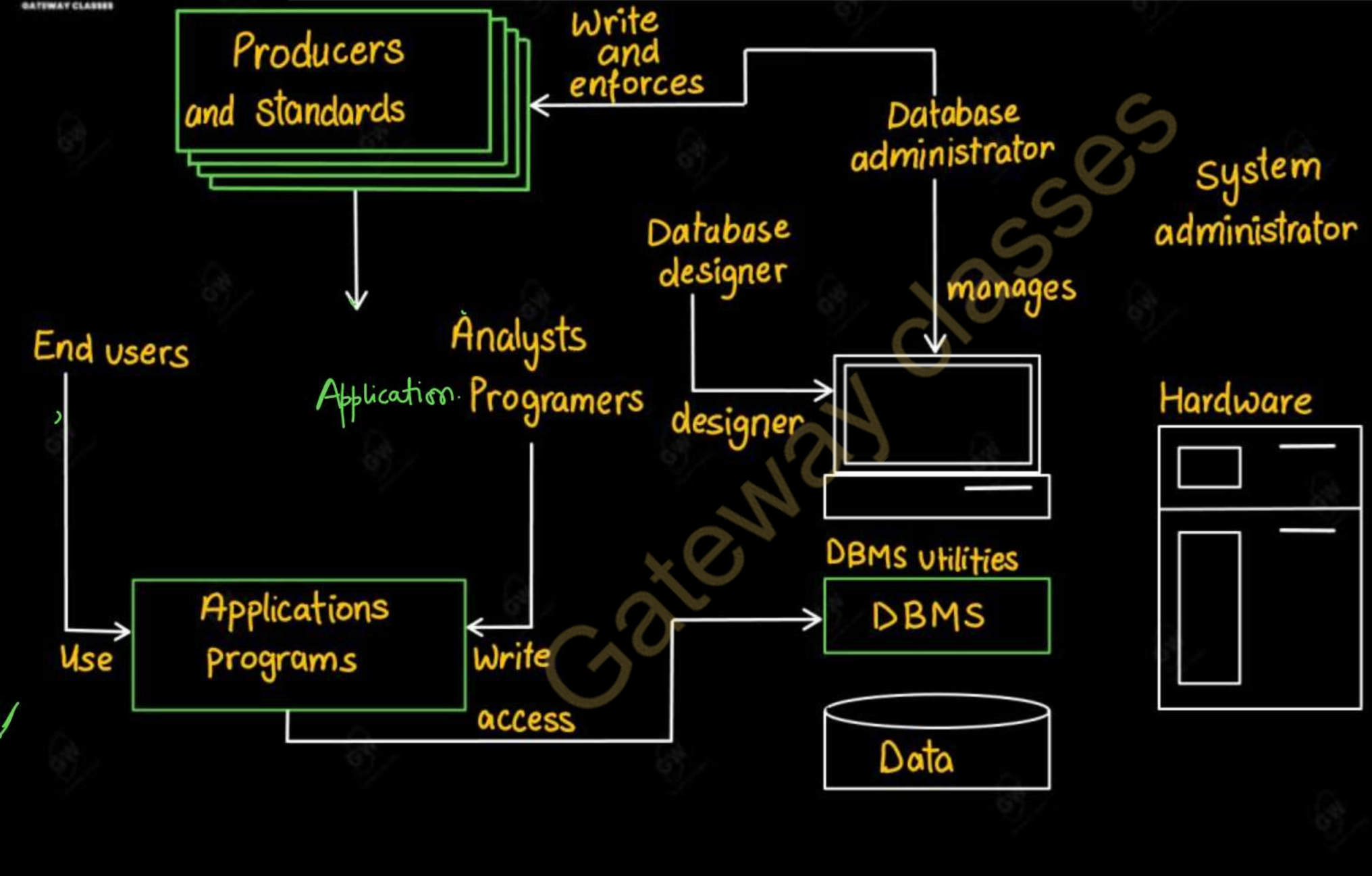
2.Data Inconsistency: Updates to data may not be reflected across all files, causing mismatched or outdated information.

3.Limited Access Control: File systems lack robust access control mechanisms, making it harder to manage user permissions securely.

4.No Multi-User Access: File systems do not efficiently support concurrent access, leading to conflicts and data corruption when multiple users access files simultaneously.

5.Lack of Backup and Recovery: File systems often lack automated backup and recovery mechanisms, increasing the risk of data loss

DATABASE USER



1. Database Administrator (DBA)

- A Database Administrator (DBA) is a person/team who defines the schema and also controls the all levels of the database.
- The DBA will then create a new account ID and password for the user if he/she needs to access the database.
- DBA is also responsible for providing security to the database and he allows only authorized users to access/modify the database.
- DBA is responsible for problems such as security breaches and poor system response time.
- DBA also monitors the recovery and backup and provides technical support.
- The DBA has a DBA account in the DBMS which is called a system or super user account.
- DBA repairs damage caused due to hardware and/or software failures.
- DBA is the one having privileges to perform DCL operations such as GRANT and REVOKE ,to allow/restrict a particular user from accessing the database.

2. Application Programmers

- Application Programmers also referred as System Analysts or simply Software Engineers, are the back-end programmers who writes the code for the application programs.
- They are the computer professionals.
- These programs could be written in Programming languages such as Visual Basic, Developer, C, FORTRAN, COBOL etc.
- Application programmers design, debug, test, and maintain set of programs called "canned transactions" for the Naive (parametric) users in order to interact with database.

3.System analyst

- System analysts are users who analyze and design database systems.
- They focus on understanding business requirements and translating them into technical database solutions.
- They ensure that the database structure meets the needs of the organization and optimize performance, security, and data flow.

4.Naive user/parametric end user

- Naive users are users who interact with the **database through simple interfaces** without knowing how the database works.
- Example: Withdrawing money from an ATM or booking tickets online. (Regular basis use database)

5. Sophisticated Users

- Sophisticated users can be engineers, scientists, business analyst, who are familiar with the database.
- They can develop their own **database applications according to their requirement**.
- They interact the database by writing SQL queries directly through the query processor.

6.Database designer

- A database designer is responsible for creating the **structure of a database.**
- They define how **data will be stored, organized,** and related within the system by designing tables, relationships, constraints.

7.Specialized users

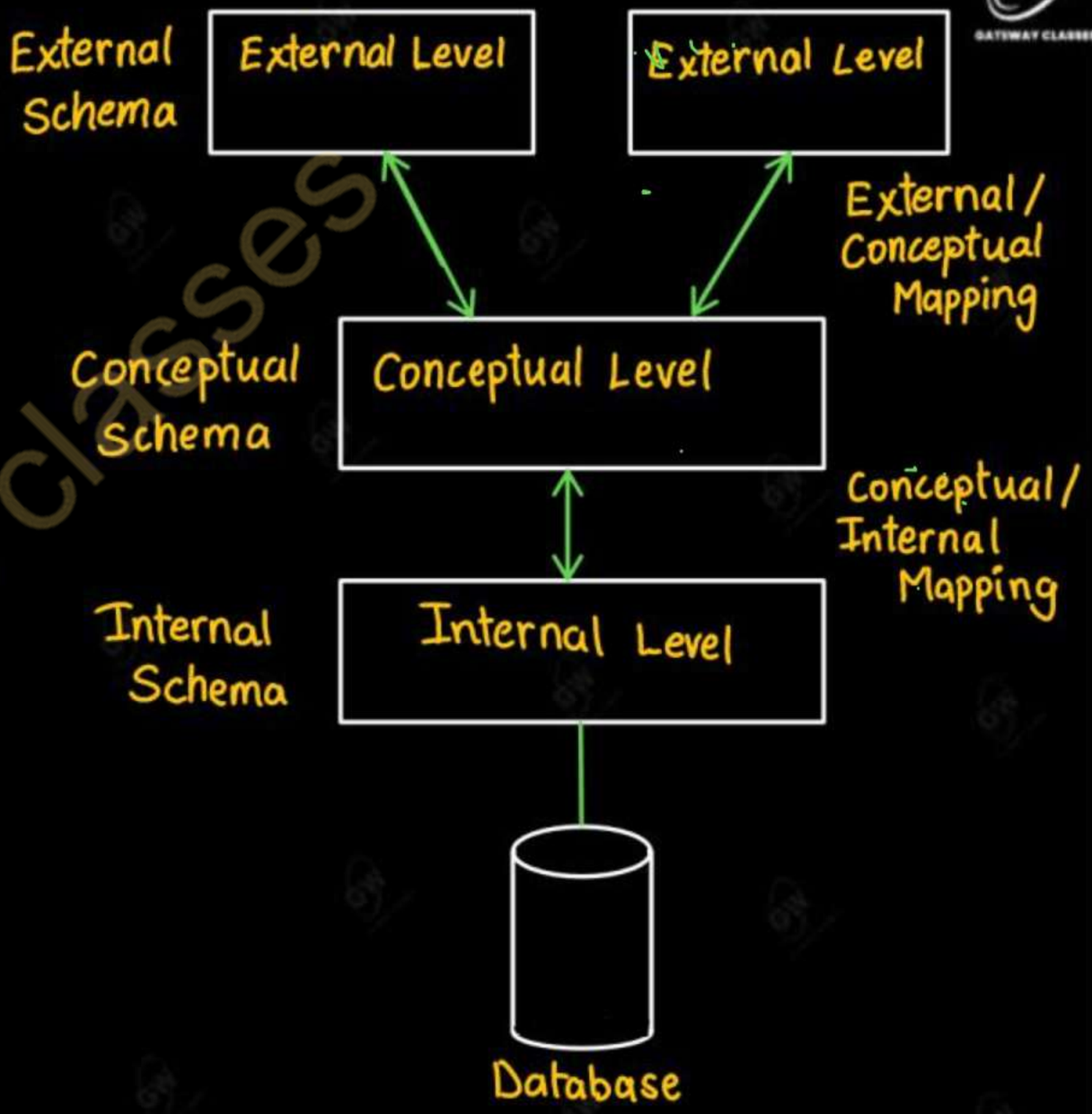
They are the user who write custom database applications or perform complex operations not supported by standard query languages.

8.Casual users, or temporary users

They are the individuals who access the database occasionally without needing detailed knowledge of its internal structure.

Three schema Architecture

- The three schema architecture is also used to separate the user applications and physical database.
- The three schema architecture contains three-levels.
- It breaks the database down into three different categories.



- Mapping is used to transform the request and response between various database levels of architecture.
- Mapping is not good for small DBMS because it takes more time.
- In External / Conceptual mapping, it is necessary to transform the request from external level to conceptual schema.
- In Conceptual / Internal mapping, DBMS transform the request from the conceptual to internal level.

Objectives of Three schema Architecture

- The main objective of three level architecture is to enable multiple users to access the same data with a personalized view while storing the underlying data only once.
- Thus it separates the user's view from the physical structure of the database.

This separation is desirable for the following reasons:

1. Different users need different views of the same data.

The users of the database should not worry about the physical implementation and internal workings of the database such as data compression and encryption techniques, hashing, optimization of the internal structures etc.

- All users should be able to access the same data according to their requirements.
- DBA should be able to change the conceptual structure of the database without affecting the user's
- Internal structure of the database should be unaffected by changes to physical aspects of the storage.

1. Internal Level

- The internal level has an internal schema which describes the physical storage structure of the database.
- The internal schema is also known as a physical schema.
- It uses the physical data model. It is used to define that how the data will be stored.
- The physical level is used to describe complex low-level data structures in detail.

The internal level is generally is concerned with the following activities:

- Storage space allocations.

  For Example: B-Trees, Hashing etc.

- Access paths.

  For Example: Specification of primary and secondary keys, indexes, pointers and sequencing.

- Data compression and encryption techniques.
- Optimization of internal structures.
- Representation of stored fields.

STORED_EMPLOYEE record length 60
Empno : 4 Decimal offset 0 unique
Ename : String length 15 offset 4
Salary : 8.2 decimal offset 19
Deptno : 4 decimal offset 27
Post : string length 15 offset 31

2. Conceptual Level

EMPLOYEE
Empno : Integer (4) key
Ename : String (15)
Salary : String (8)
Deptno : Integer (4)
Post : String (15)

The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level.

- The conceptual schema describes the structure of the whole database.
- The conceptual level describes what data are to be stored in the database and also describes what relationship exists among those data.
- In the conceptual level, internal details such as an implementation of the data structure are hidden.
- Programmers and database administrators work at this level.

3. External Level

- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- An external schema is also known as view schema.
- Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
- The view schema describes the end user interaction with database system.

Mapping between Views

The three levels of DBMS architecture don't exist independently of each other.

There must be correspondence between the three levels i.e. how they actually correspond with each other.

DBMS is responsible for correspondence between the three types of schema. This correspondence is called Mapping.

There are basically two types of mapping in the database architecture:

- Conceptual/ Internal Mapping
- External / Conceptual Mapping

Conceptual mapping in DBMS

- It refers to the process of translating or linking the conceptual schema (logical view of the entire database) to the internal schema (physical storage structure).

Key Functions:

Logical-Physical Mapping: It connects logical data structures (tables, attributes) to their physical storage formats (files, indexes).

Data Independence: Conceptual mapping provides physical data independence, allowing changes in physical storage (e.g., indexing, file organization) without altering the conceptual schema.

External mapping in DBMS

- It refers to the process of linking the external schema (user-specific views) to the conceptual schema (overall logical structure).
- This mapping ensures that different users can access customized views of the data without affecting the underlying database structure.

Key Functions:

- **View-Logical Mapping:** It connects user views (external schema) to the logical structure (conceptual schema) by filtering and presenting relevant data.
- **Data Independence:** External mapping provides logical data independence, allowing changes to the conceptual schema without affecting user views.

AKTU QUESTION

Q.1	State the procedural DML and non-procedural DML With their difference	AKTU 2020-21
Q.2	What are the four advantages of file system Over database	AKTU 2022-23
Q.3	What is database, DBMS, Metadata.	AKTU 2022-23
Q.4	List some problem with file system.	AKTU 2022-23
Q.5	Explain all the user of database in detail.	AKTU 201-22
Q.6	Describe the three schema architecture .Why we need mapping between schema level	AKTU 2022-23

Database Management System

UNIT 1
INTRODUCTION

Today's Target

- ER diagram Questions
- AKTU PYQs

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

Q.1. In a university, a student enrolled in course and a student be assigned at least one or more course .Each course is taught by a single professor .To maintain instruction quality a professor can delivery only one course.

① Entity Set

→ Student
→ Course
→ professor

② Attribute

Student — Student_id (P.k)

Student_id, Stud_name,
Address

Course

Course_id, Coursename, Duration — Courseid (P.k.)

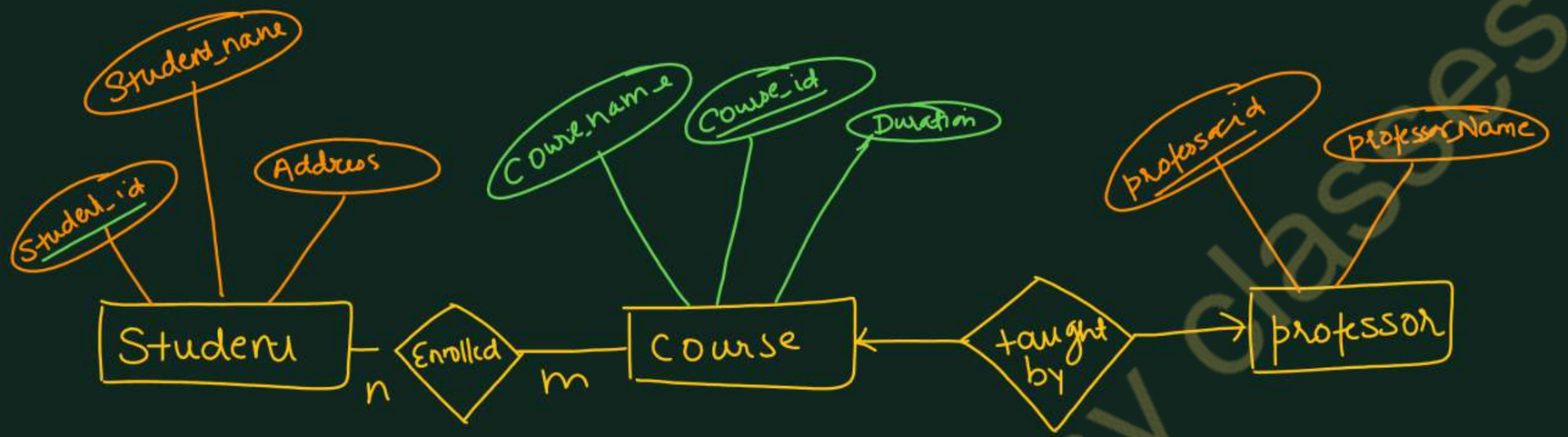
Professor

professor_name, professor_id — professor_id (P.k)

①

②

(Realtionship Enrolled
taught by
(Cardinality Constraints)



ERDAIGRAM

Q.2. Draw an ER Model for an University database application where

a) A University has many departments.

b) Each department has multiple instructors; one among them is the head of the department

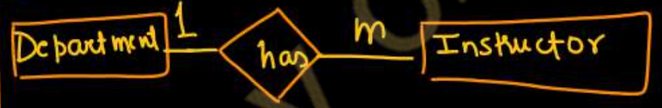
c) An instructor belongs to only one department

d) Each department offers multiple courses, each of which is taught by a single instructor.

e) A student may enroll for many courses offered by different departments.

Entity set	
Department	Department Department_id, Deptname, Location
Instructor	Instructor Instname, Inst_id, Roomno
Course	Course Course_id, Coursename, duration
Student	Student Student_id, Student_name, Address, DOB

1.

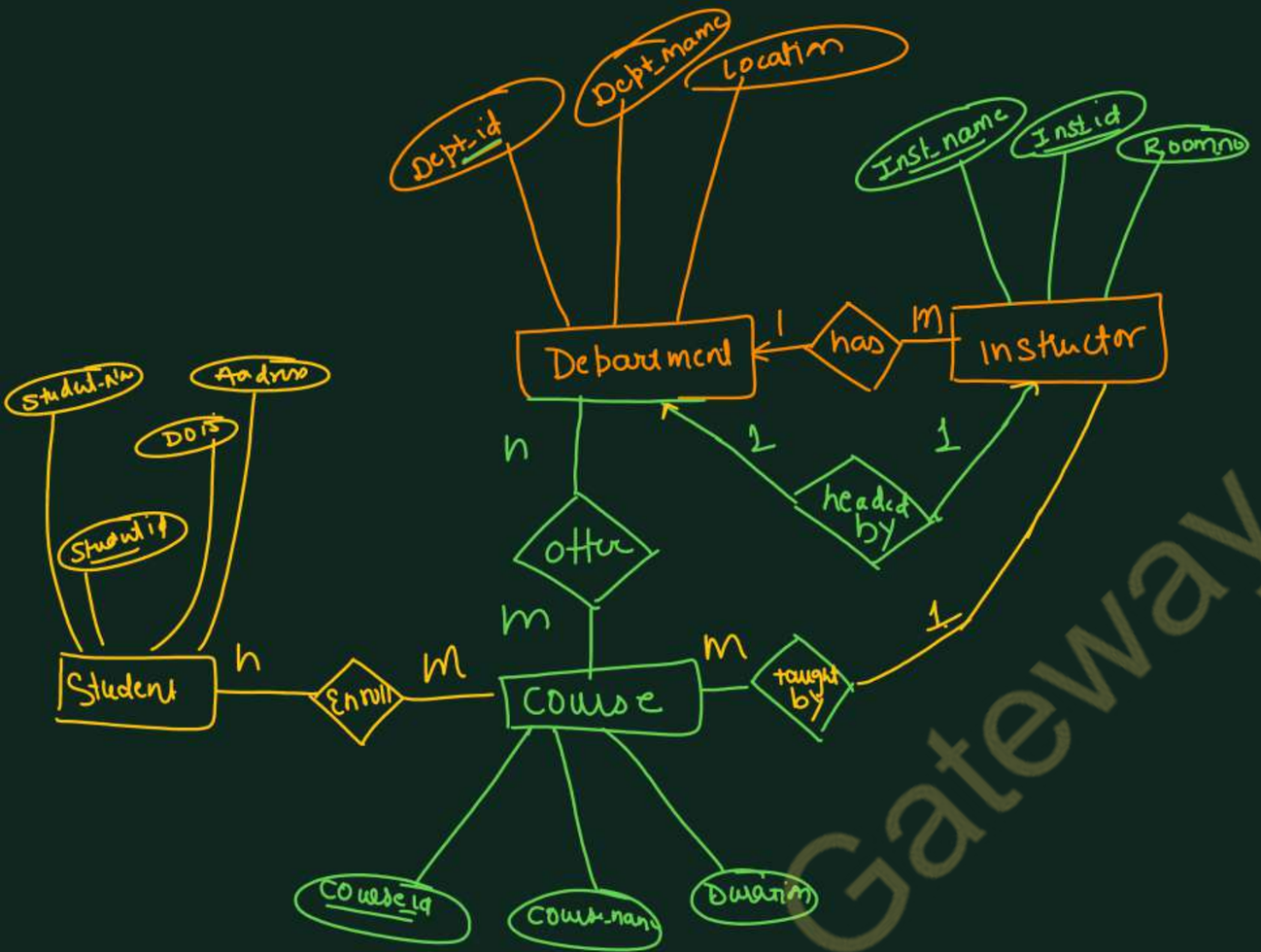


2.

3.

4.

5.



Q.3. Construct an E-R diagram for a car insurance company whose customers own one or more cars each .Each car associated with it zero to any number of recorded accidents.

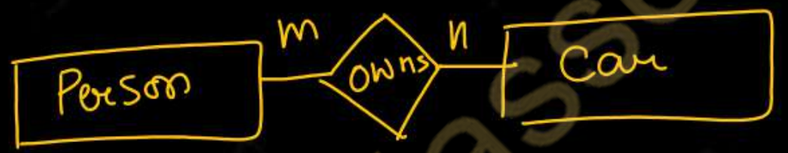
Entity set

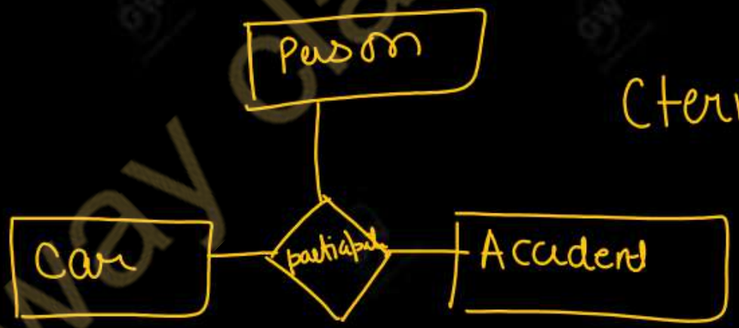
Person
Car
Accident

Attribute

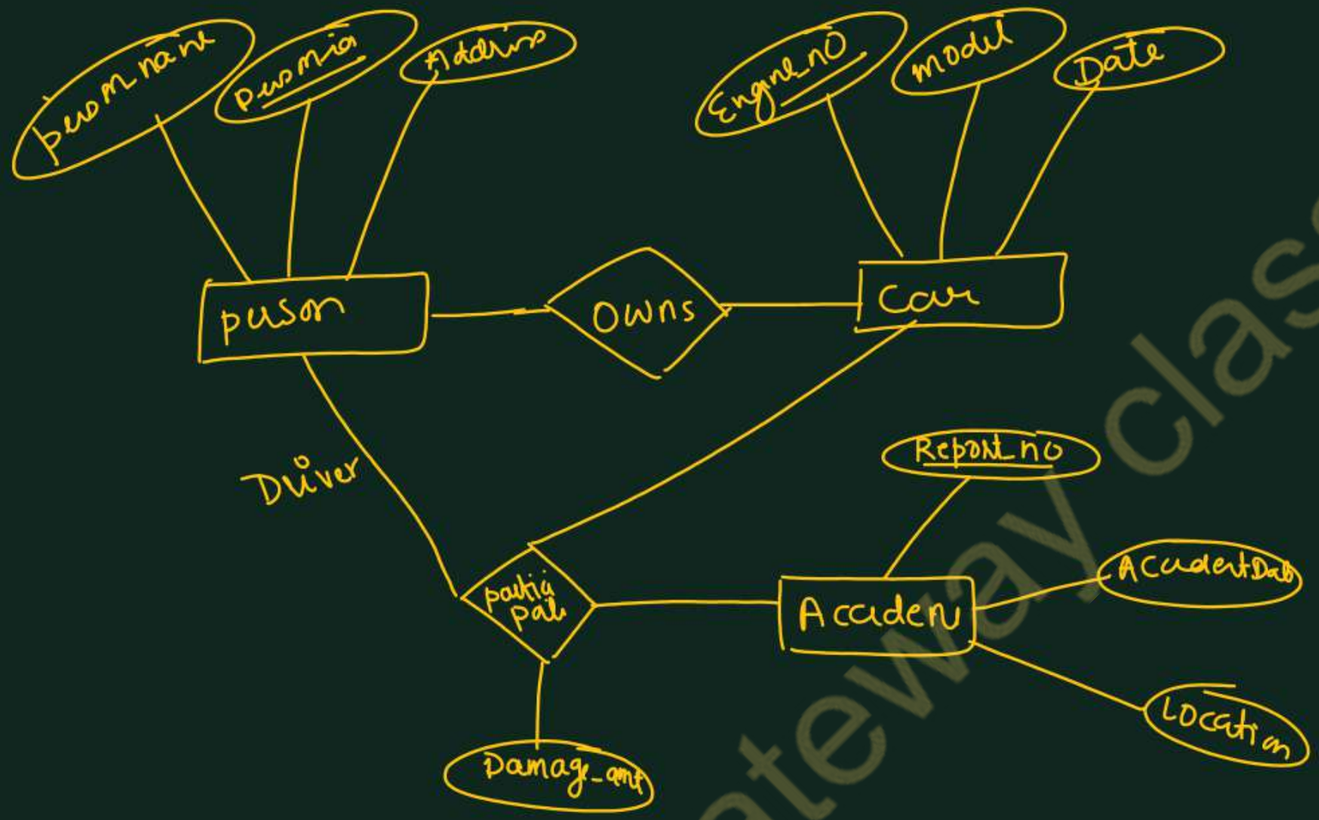
1 Person
person_id, person_name
Address
2. Car
Engineno, model, Date

3. Accident
Report_no
Accident_date
location





(ternary Relationship)



How does different definition language support three schema architecture.

2 Marks

Different data definition languages (DDL) support the Three-Schema Architecture by allowing operations at different schema levels:

- **Internal Schema:** Low-level DDL commands define storage structures (e.g., tablespaces in SQL).
- **Conceptual Schema:** DDL defines the logical structure, such as tables, relationships, and constraints (e.g., CREATE TABLE in SQL).
- **External Schema:** DDL helps create user views (e.g., CREATE VIEW in SQL) that present customized representations of the data.

AKTU QUESTIONS

Q.1	Describe the three schema architecture .Why do we need mapping between schema level .How does different definition language support this architecture.	AKTU 2022-23

Database Management System

UNIT 1
INTRODUCTION

Today's Target

- ER diagram Questions

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

Q.1. Design an E-R diagram for keeping track of the exploits of your favorite sports team. You should store the matches played, the scores in each match, the players in each match and individual player statistics for each match. Summary statistics should be modeled as derived attributes. + table (Relational Schema)

Entity set	Attribute
1. match	match (strong entity set)
2. player	match_id, Stadium, Date
	Own_score, Opponent_Score
	Oponent, No_of_player
	Result
	(match_id (Primary Key)

Player

player_id (Strong entity Set)

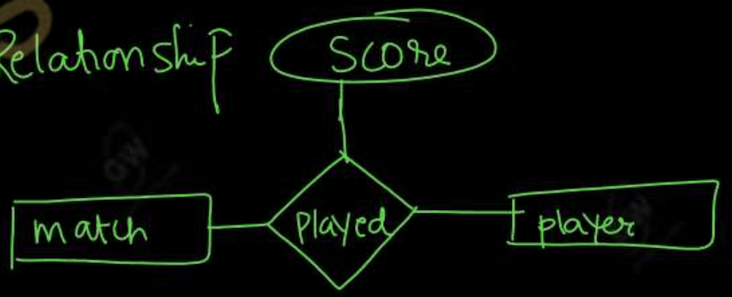
player_Name

Summary Statistics (derived attribute)

(player id)

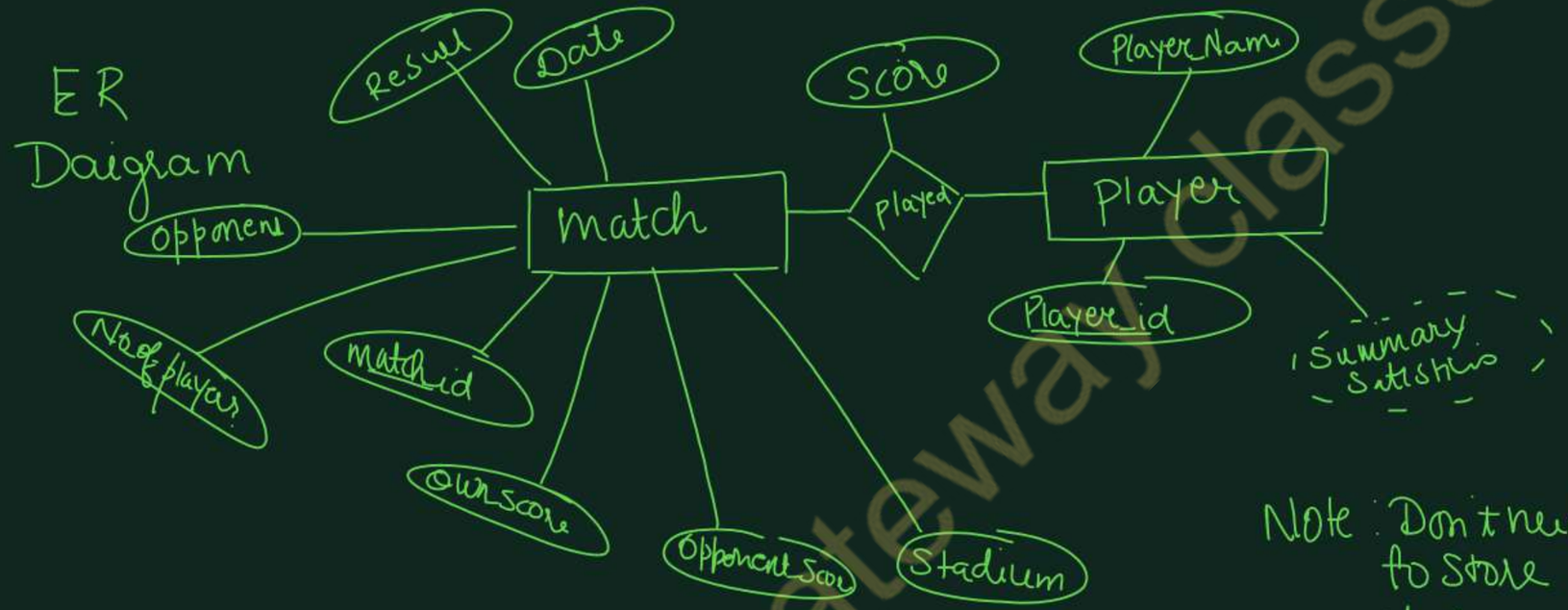
Primary Key

Relationship



Cardinality Constraints m:n

ER Daigram



Note: Don't need to store derive attribute

table

Relational Schema

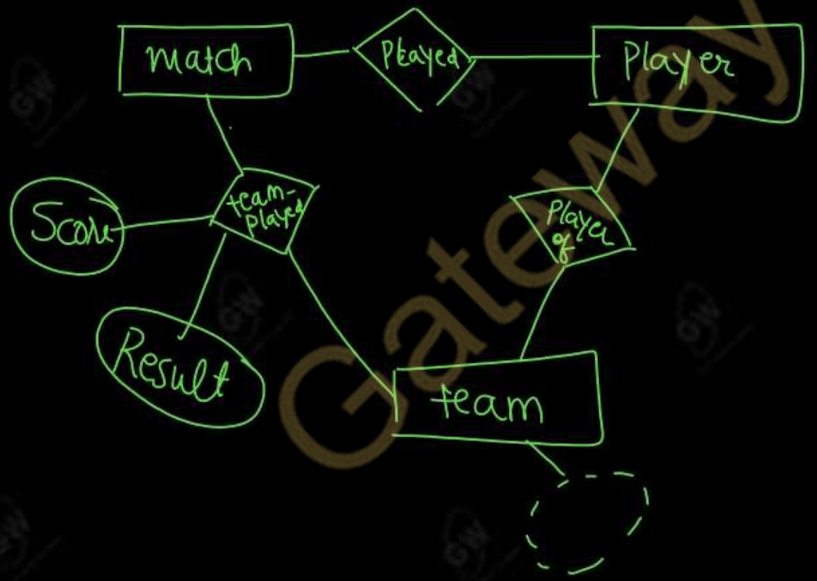
match (match_id , Result, Date, opponent , No of player , match_id, own score , Opponent Score , Stadium)

player (Player_Name , Player_id)

Played (match_id Player_id , Score)

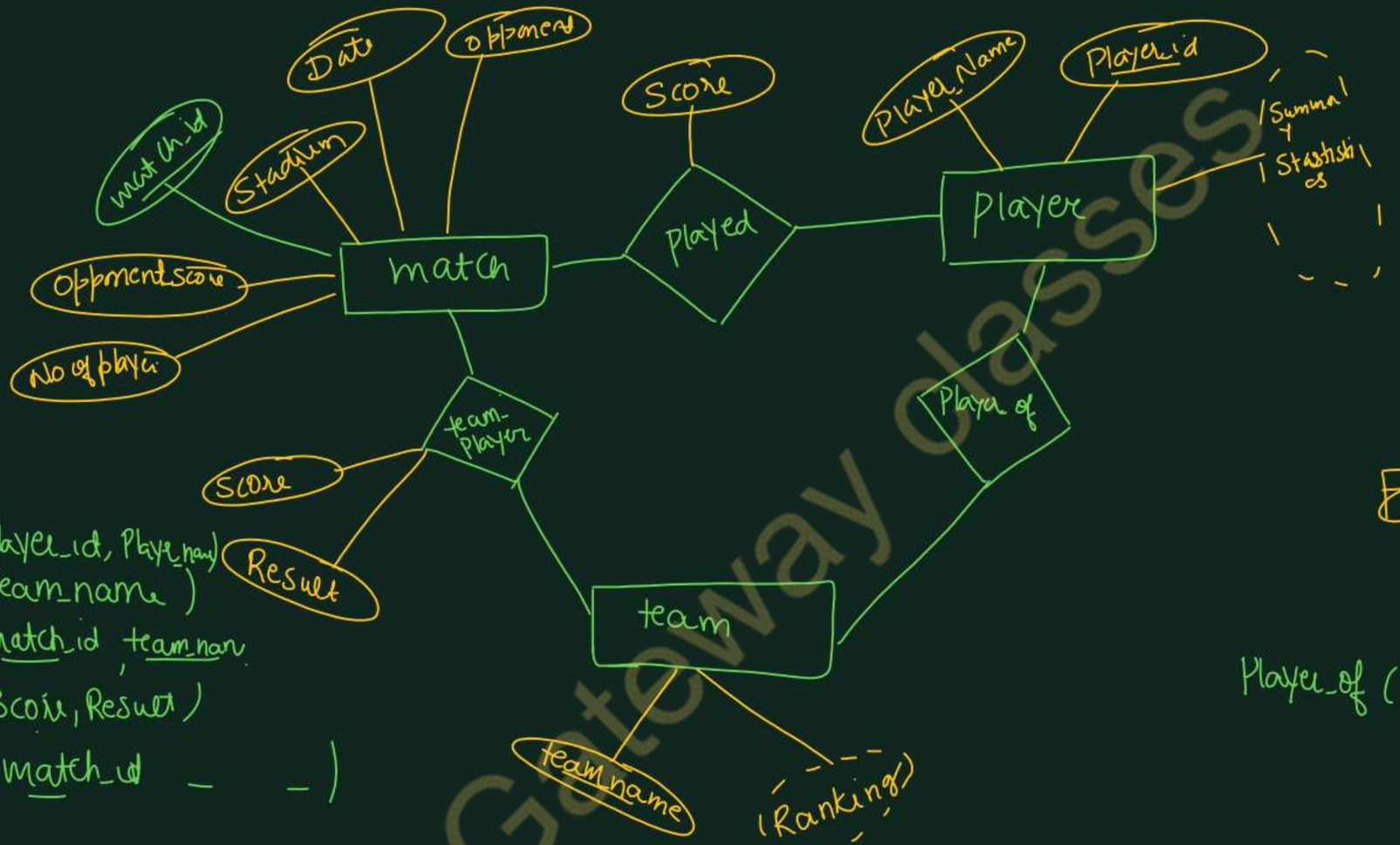
Q.2. Design an E-R diagram for keeping track of the exploits of your favorite sports team. You should store the matches played, the scores in each match, the players in each match and individual player statistics for each match. Summary statistics should be modeled as derived attributes.

Extend the E R diagram of the pervious question to track the same information for all teams in a league



Entity Set
1. match
2. player
3. team

team
→ team_name
→ Ranking (derived attribute)



E-R Daiglam

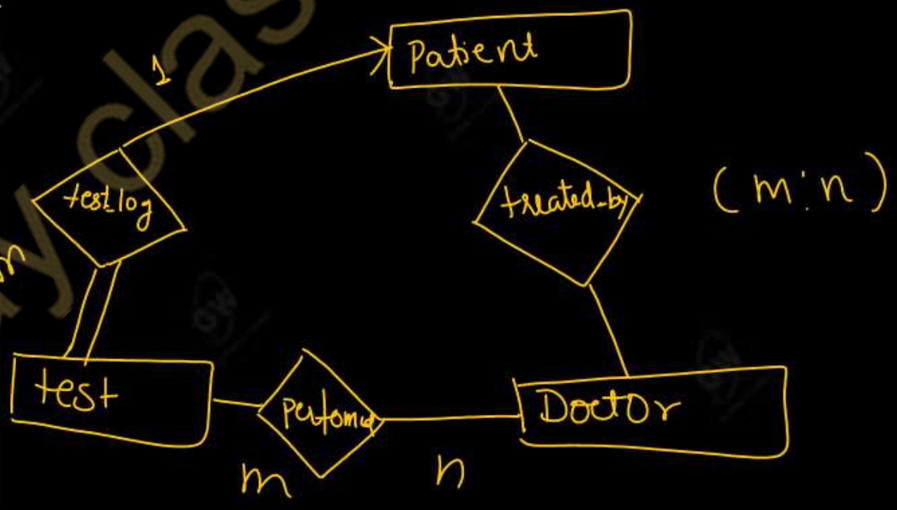
Player (Player_id, Player_name)
team (team_name)
team_player (match_id, team_name, Score, Result)
Match (match_id _ _)

Player_of (player_id, team_name)

Q.3. Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.

Entity Set
1. Patient
2. Doctor
3. test

Attribute	(Strong attribute)
Patient (Strong Entity Set)	Doctor
P_id, Insurance, admitted_date, Check_out_date name	D_id name Specialisation
test	
t_id, Date, Result, name, time	(Strong Entity Set)



(table)

Relational Schema

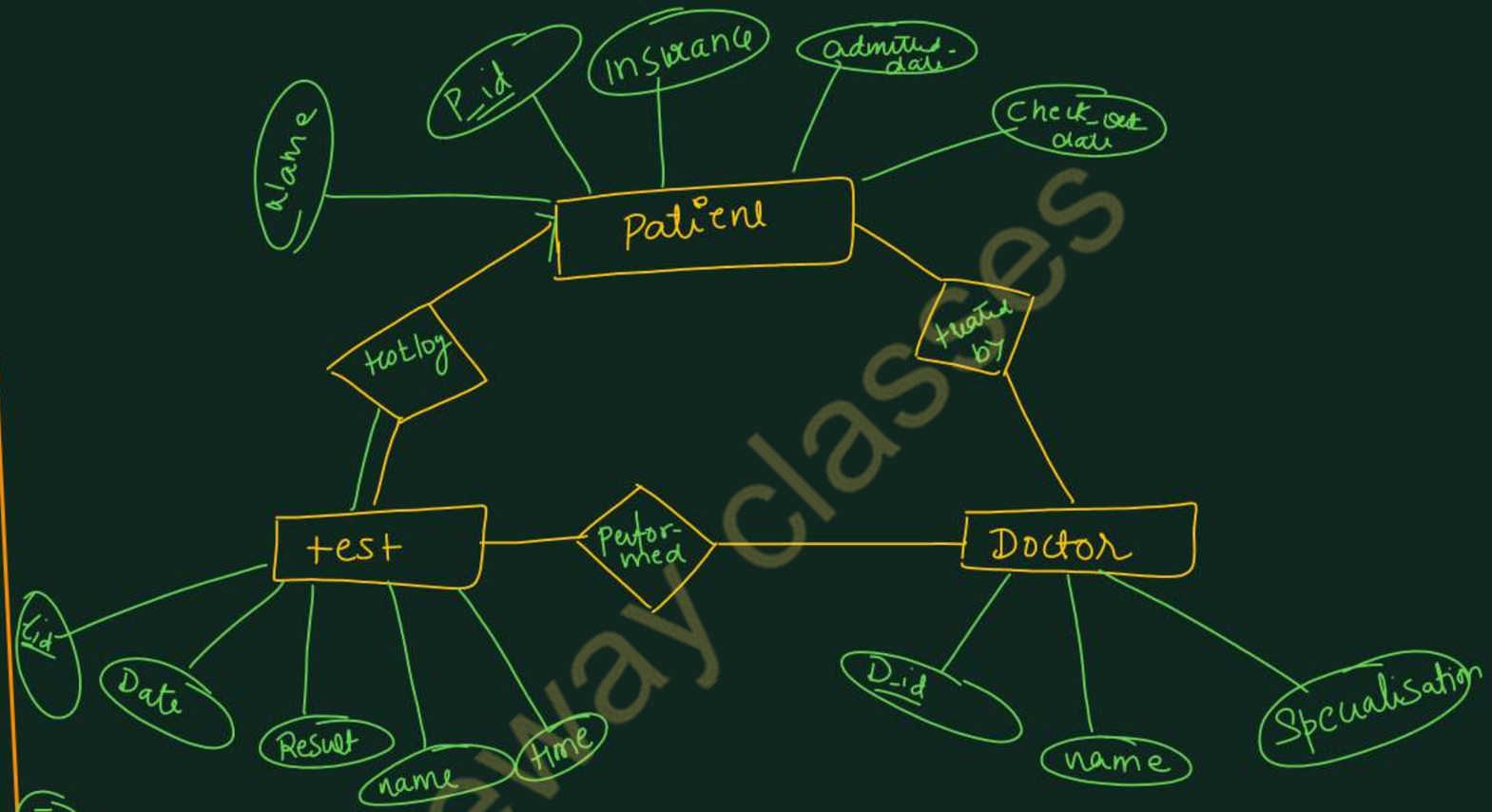
① Patient (Name, P_id, Insurance, admitted_date, Check_out_date)

② Doctor (D_id, name, Insurance)

③ treated by (D_id, P_id)

test (t_id, Date, Result, ④ name, time, P_id)

⑤ Performed by (t_id, D_id)



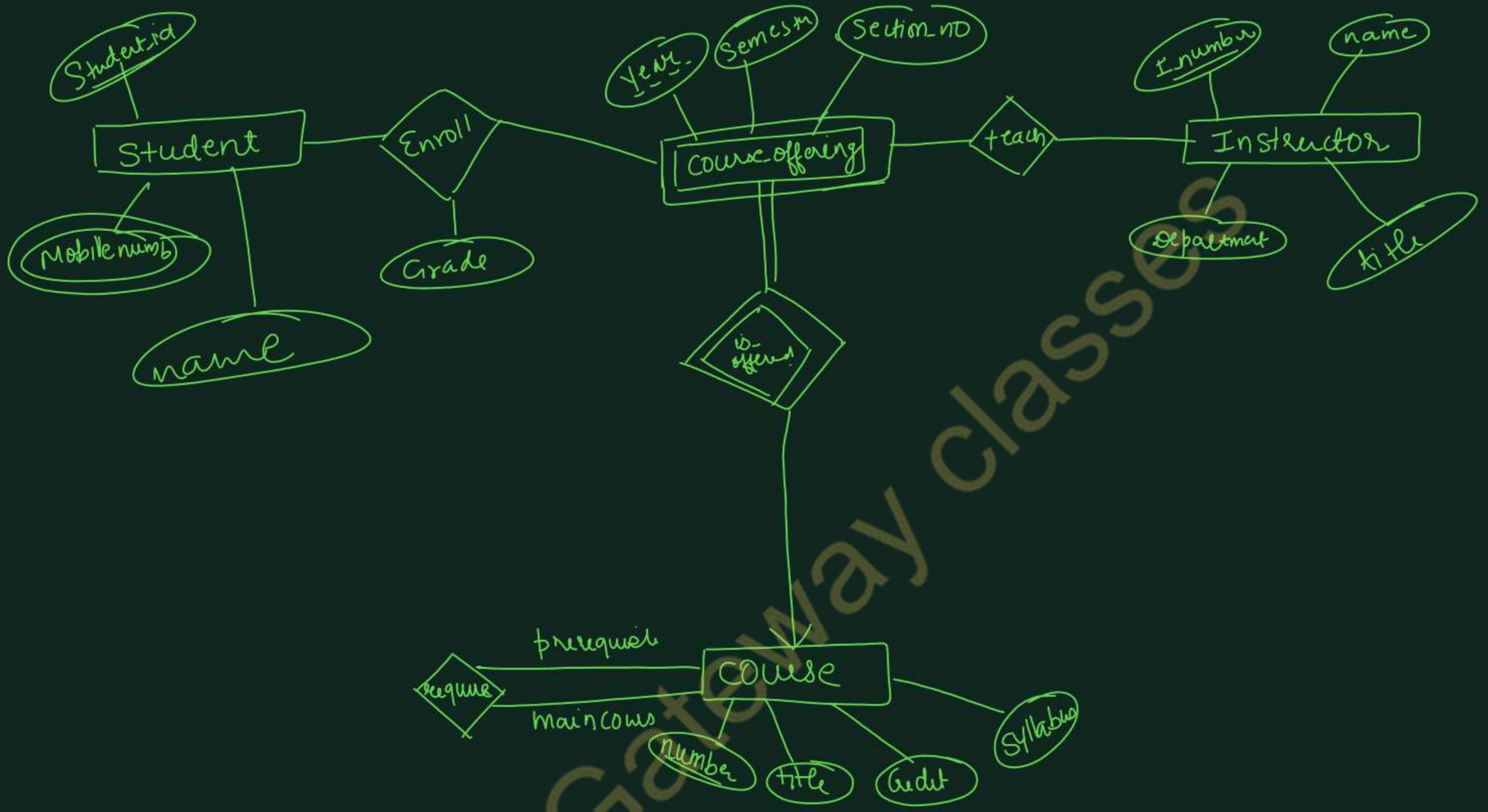
Q.4 A university registrar's office maintains data about the following entities:

(a) courses, including number, title, credits, syllabus, and prerequisites;

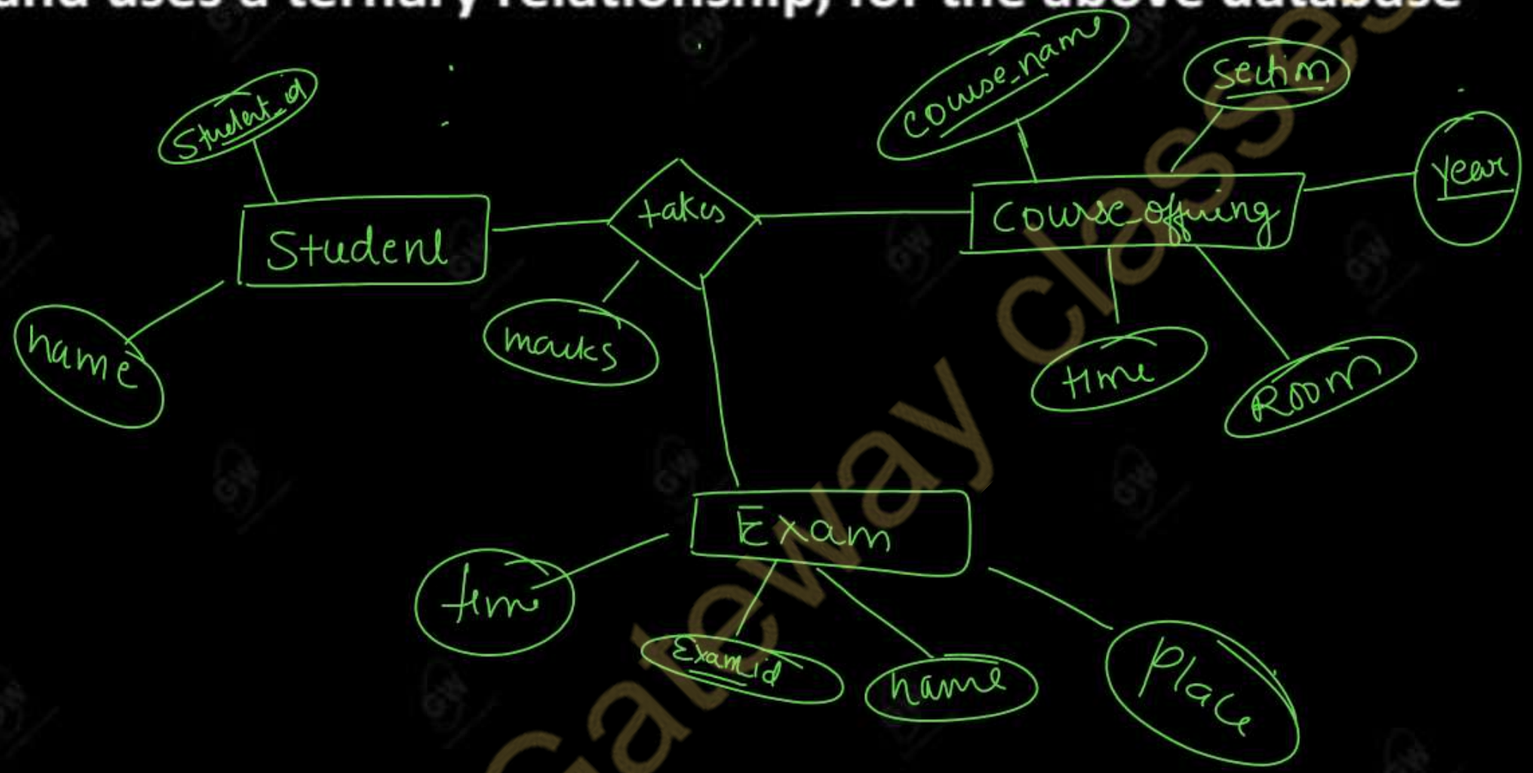
(b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom;

(c) students, including student-id, name, and program; and

(d) instructors, including identification number, name, department, and title. Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled.Construct an E-R diagram for the registrar's office. Document all assumptions that you make about the mapping constraints.



Q.5 Consider a database used to record the marks that students get in different exams of different course offerings. a) Construct an E-R diagram that models exams as entities, and uses a ternary relationship, for the above database



Q.6: Suppose you are given the following requirements for a simple database for the National Hockey League (NHL):

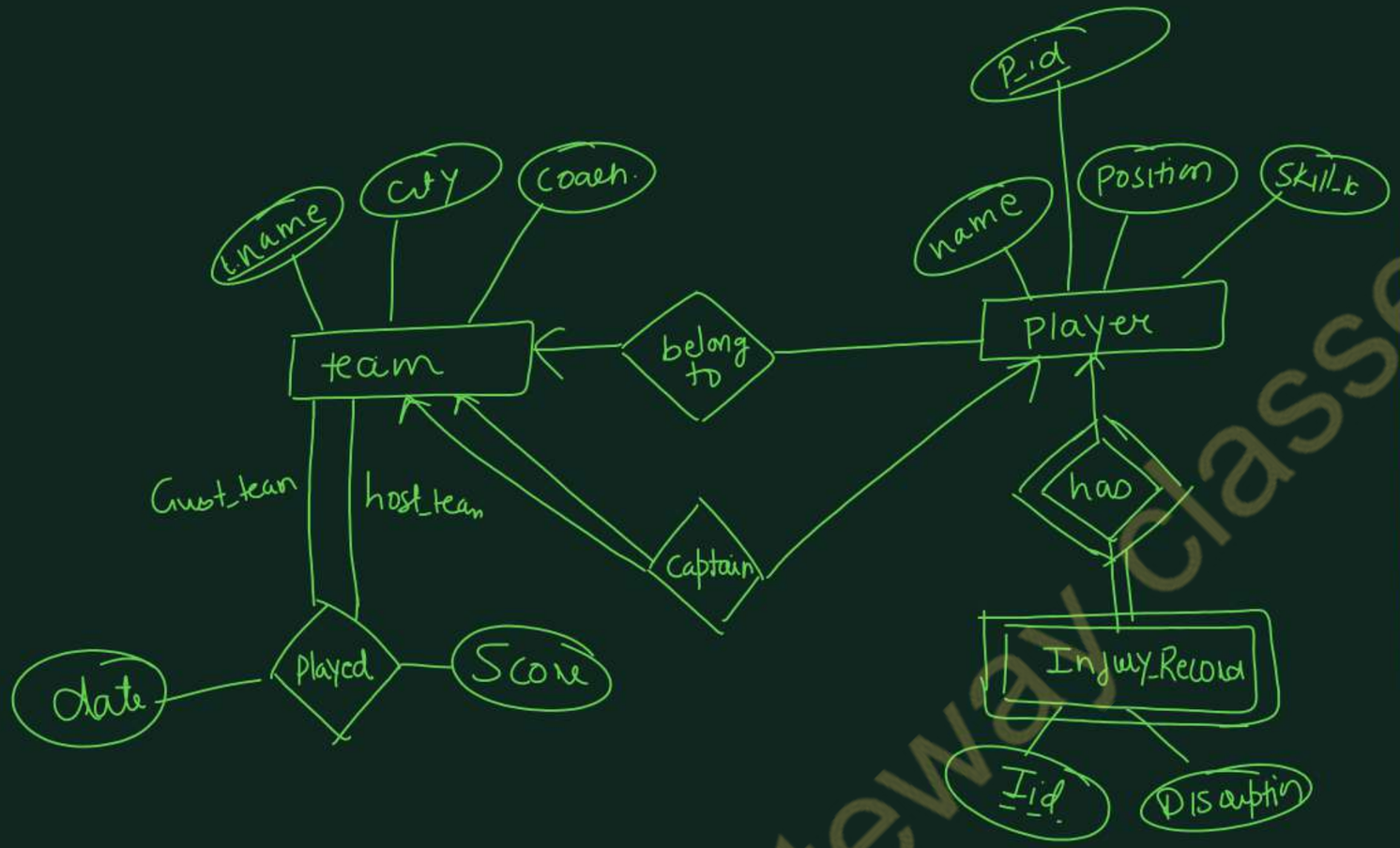
1. the NHL has many teams, each team has a name, a city, a coach, a captain, and a set of players, each player belongs to only one team,

2 each player has a name, a position (such as left wing or goalie), a skill level, and a set of injury records,

3 a team captain is also a player

, 4. 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.

(Relational Schema)



team (t_name, City, Coach, Captain_id)

Player (Pid, name, position, Skill_level, t_name

Injury_Record (Pid, Iid, Disaption)

Played (Guest_team, host_team, t_name, date, Score)

Database Management System

UNIT 1
INTRODUCTION

Today's Target

- ER diagram Questions
- EER diagram
- AKTU PYQs

By PRAGYA RAJVANSHI

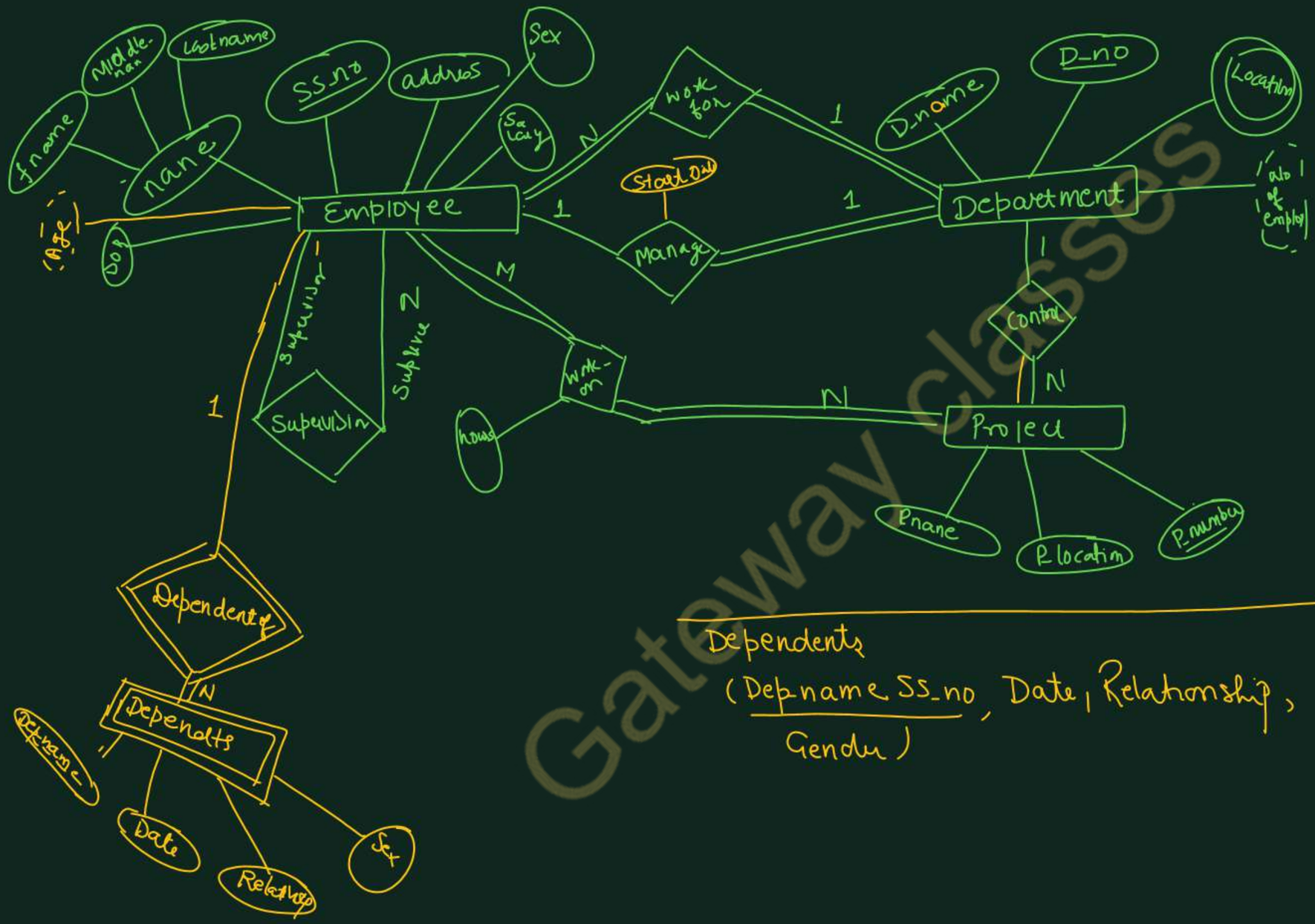
B.Tech, M.Tech(C.S.E.)

Q.1 Construct an ER Diagram for Company having following details :

1. Company organized into DEPARTMENT. Each department has unique name and a particular employee who manages the department. Start date for the manager is recorded. Department may have several locations.
2. A department controls a number of PROJECT. Projects have a unique name, number and a single location.
3. Company's EMPLOYEE name, ssno, address, salary, sex and birth date are recorded. An employee is assigned to one department, but may work for several projects (not necessarily controlled by her dept). Number hours/week an employee works on each project is recorded; The immediate supervisor for the employee. ? Employee's DEPENDENT are tracked for health insurance purposes (dependent name, birthdate, relationship to employee.

OR

ER Diagram for Employee Project Management System



Dependents
(Dep_name SS_no, Date, Relationship, Gender)

Relational schema

Employee

(fname, Middlename, Lastname, SSno address, Sex, Salary DoB Dno, Supa_ssn)

Department
(Dno, Dname, Mgr_ssn, Start_Date

Department location.
(D_no, location)

Project -
(Pnumber, Pname, P_location Dname)

WORK_on
(Ssn, Pnumber, hours)

ENHANCED ER DAIGRAM

- Enhanced ERDs are high-level models that represent the requirements and complexities of complex databases.
- The EER model includes all modeling concepts of the ER model. In addition, EER includes the following concepts.

1. Subclasses and Super classes
2. Specialization and Generalization
3. Category or Union type
4. Attribute and relationship inheritance

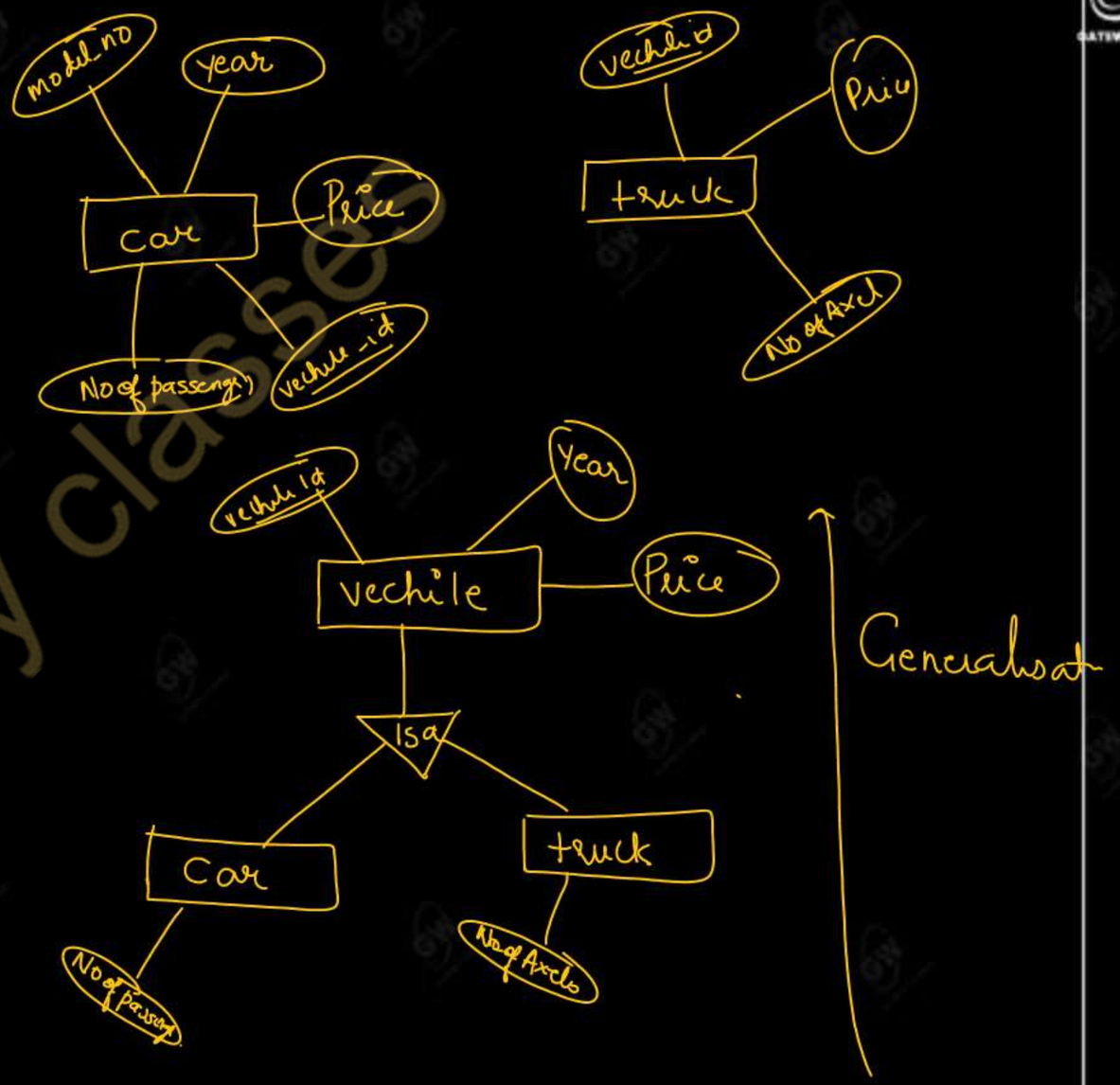
Superclass and Subclass

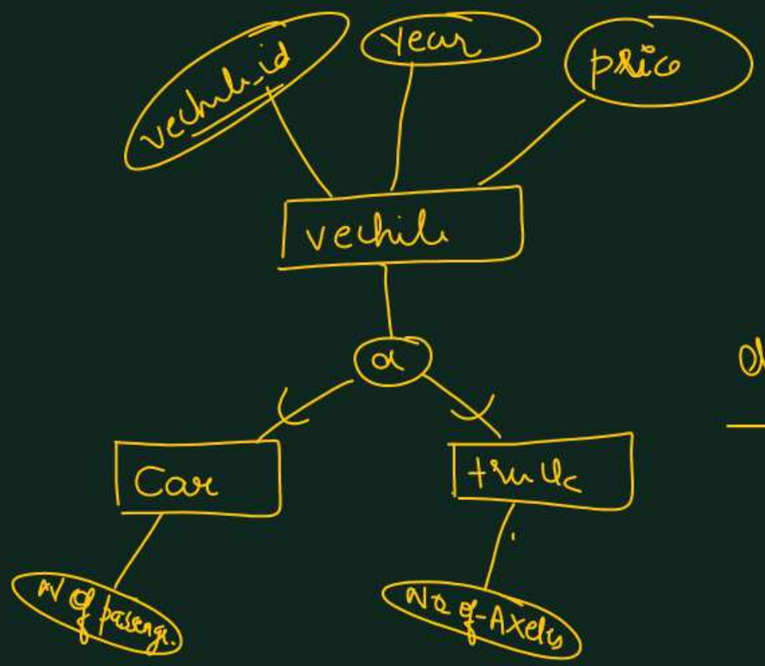
- A superclass is a high-level entity that can be further segmented into subclasses or subsets,.
- It is also referred to as a Parent class.
- A subclass can be referred to as a child or derived class.

Example: Science is a Super class which has subclasses like Physics, Chemistry, Biology.

Generalization

- Generalization is the process of extracting common properties from a set of entities and creating a generalized entity set from it.
- It is a bottom-up approach in which two or more entities set can be generalized to a higher-level entity set if they have some attributes in common.





d → disjoint

—— partial participation

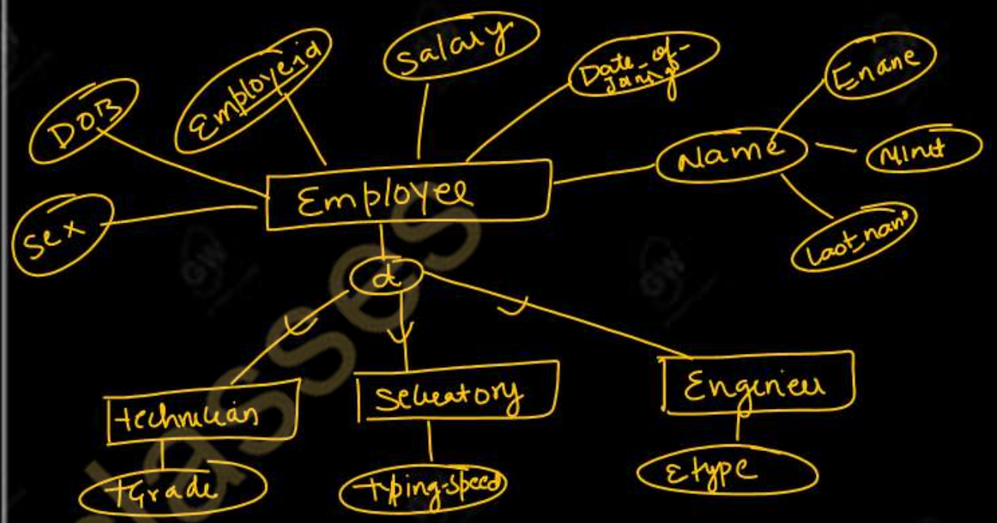
Generalisation
Generalization
(Bottom up approach)

Specialisation
(top-down Approach)

Specialization

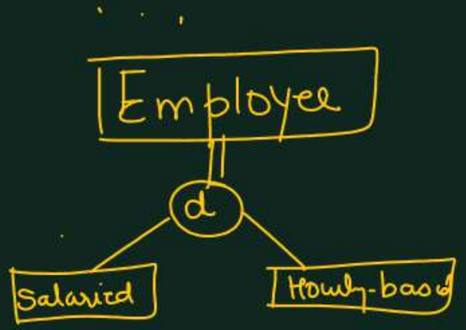
- In specialization, an entity set is divided into sub-entities set based on its characteristics.
- It is a top-down approach where the higher-level entity set is specialized into two or more lower-level entities set

Constraints
1. disjoint, partial participation



d → disjoint
(Employee either be a technician, Secretary, Employee)
partial partial —

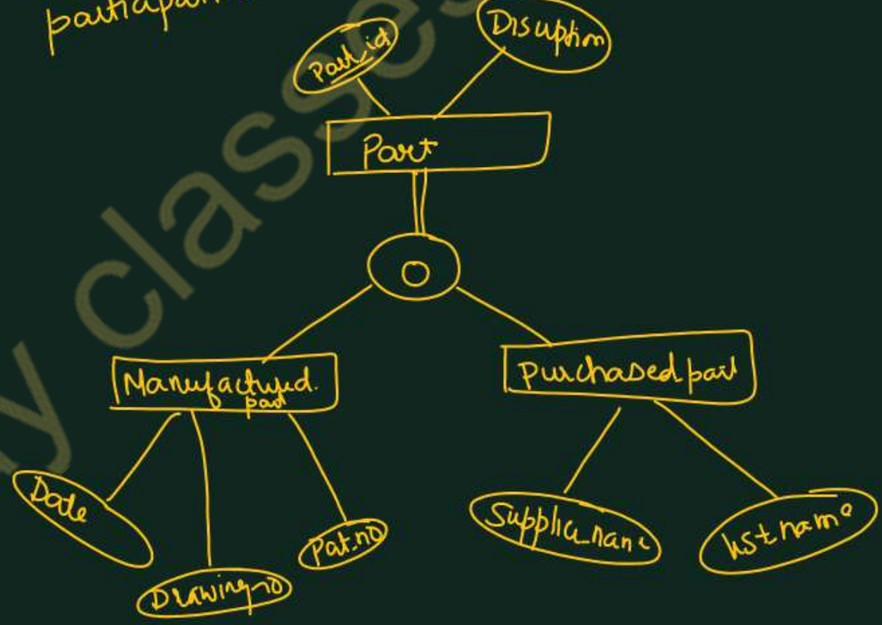
Attribute Constraints
technician → tgrade
Sex, DOB, Employee_id etc. Inherits Employee entity set.



disjoint, total participation

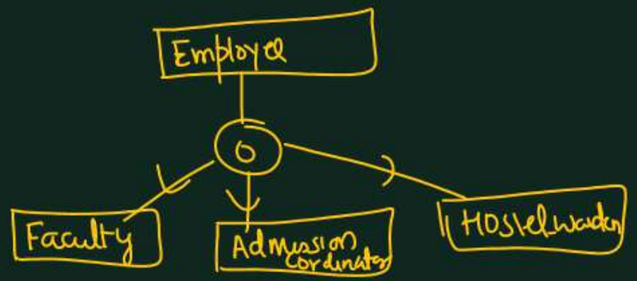
Every employee either be Salaried or Hourly-based

Overlap, total participation



(Inheritance)

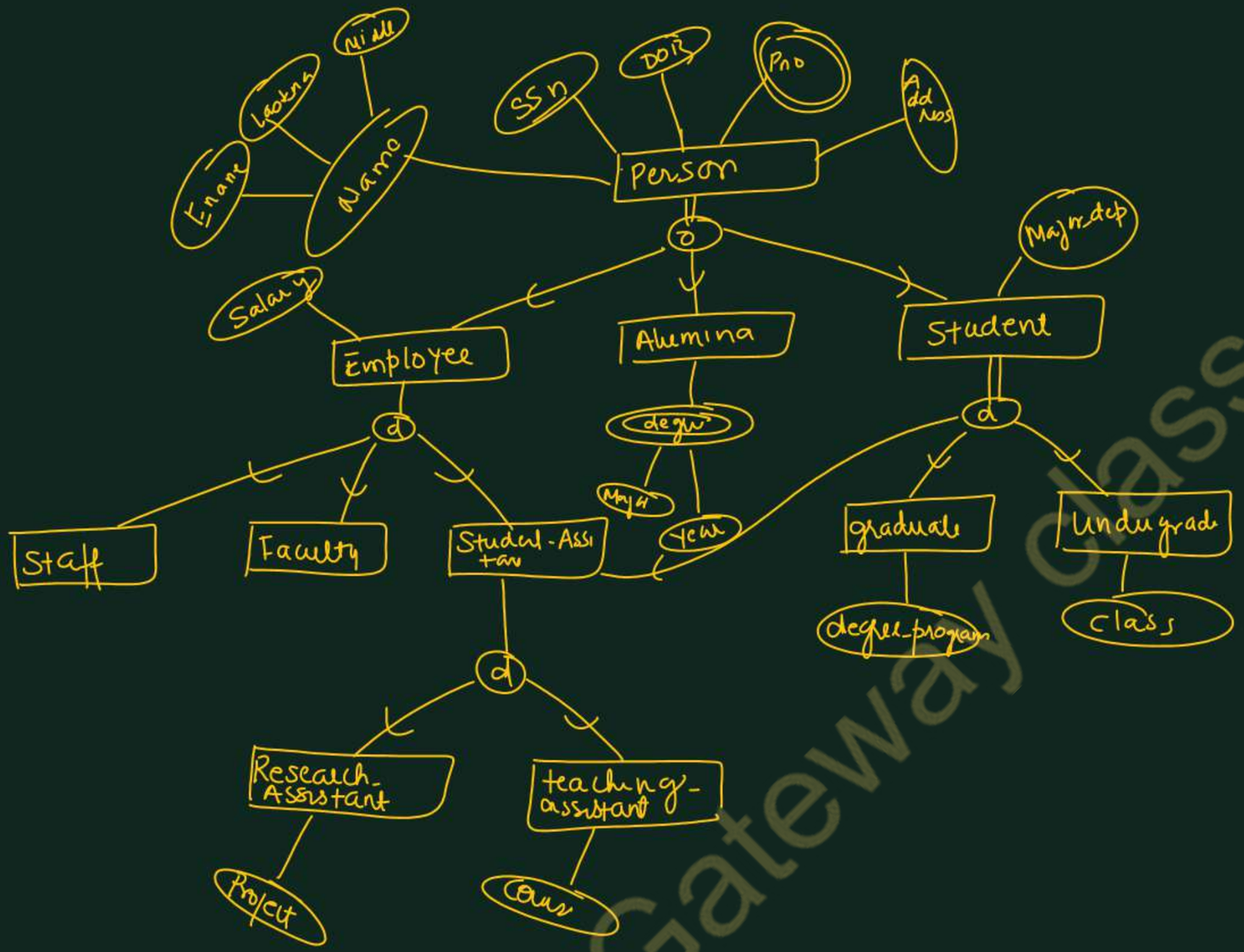
Overlap partial participation



Inheritance: It is an important feature of generalization and specialization

1.Attribute inheritance : allows lower level entities to inherit the attributes of higher level entities and vice versa.

2.Participation inheritance : In participation inheritance, relationships involving higher level entity set also inherited by lower level entity and vice versa.

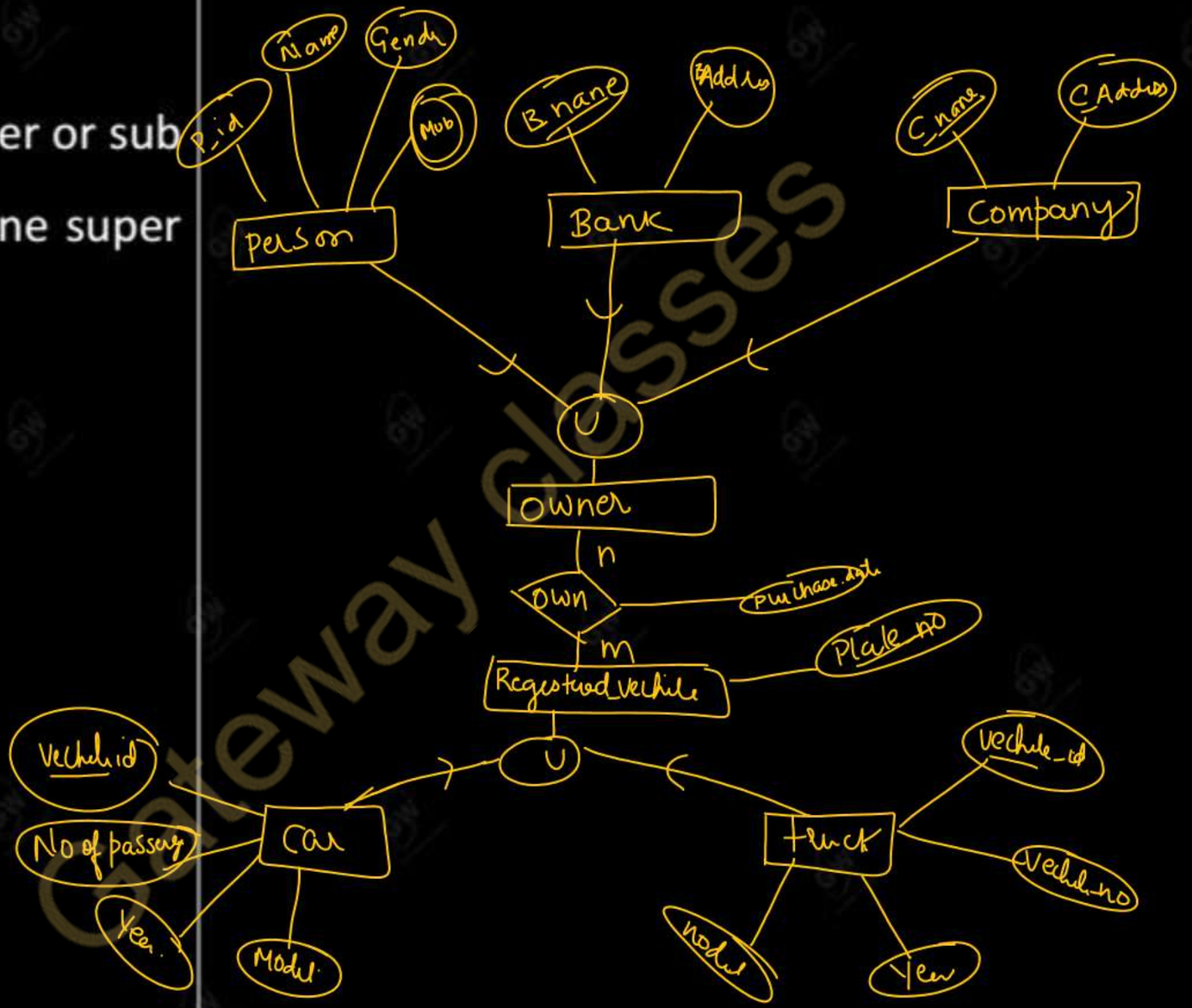


(University database With Multiple Inheritance)

1. Generalisation
2. Specialisation
3. Attribute Inheritance
4. Disjoint, Overlap
   total, partial

category or union

- Relationship of one super or sub class with more than one super class.



Constraints

In Enhanced ER (EER) diagrams, the concepts of total and partial constraints refer to the relationship between a superclass and its subclasses in a generalization/specialization hierarchy.

Total Participation (Total constraint):

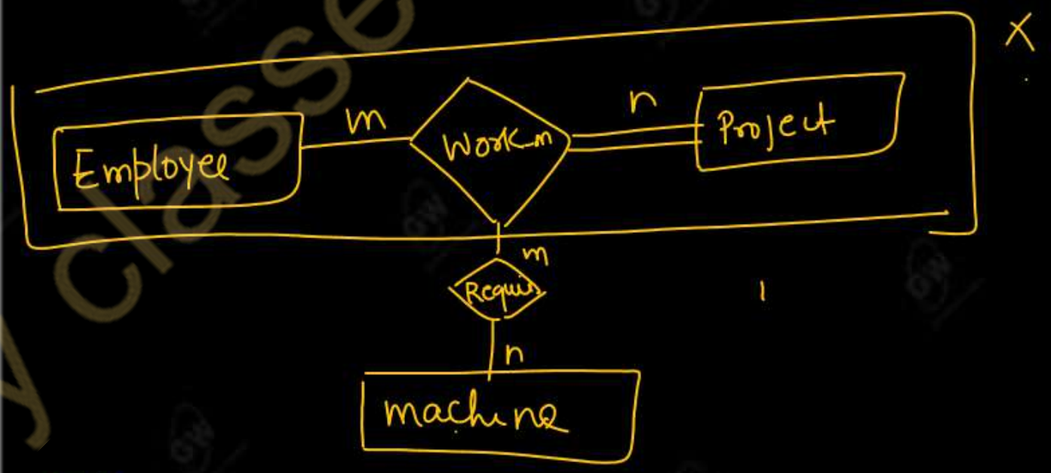
- Every entity in the superclass must be a member of at least one subclass.
- In EER diagrams, this is represented by a double line connecting the superclass

Partial Participation (Partial constraint):

- Some entities in the superclass may not belong to any subclass.
- This is shown using a single line connecting the superclass

Aggregation

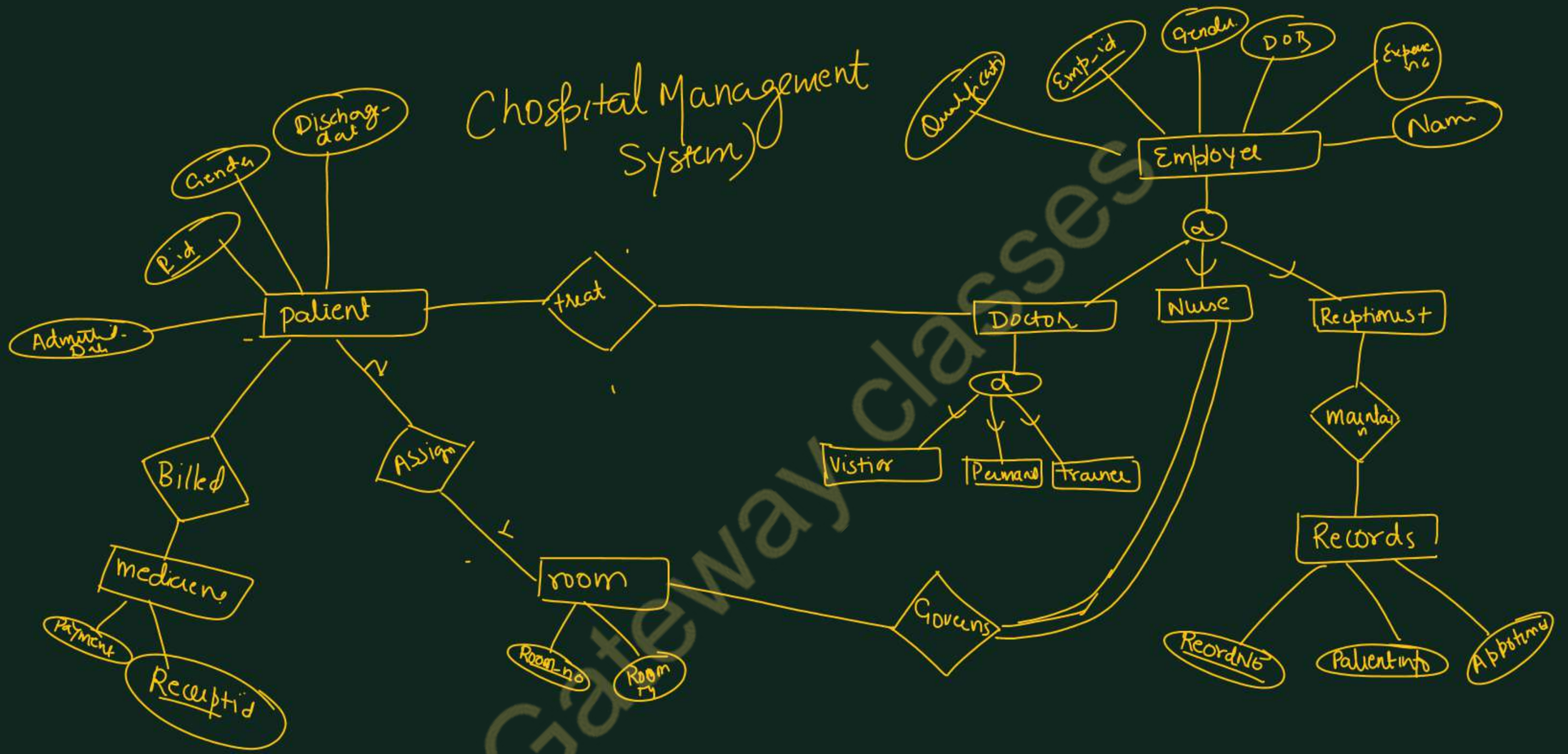
- An ER diagram is not capable of representing the relationship between an entity and a relationship which may be required in some scenarios.
- In those cases, a relationship with its corresponding entities is aggregated into a higher-level entity.
- Aggregation is an abstraction through which we can represent relationships as higher-level entity sets.

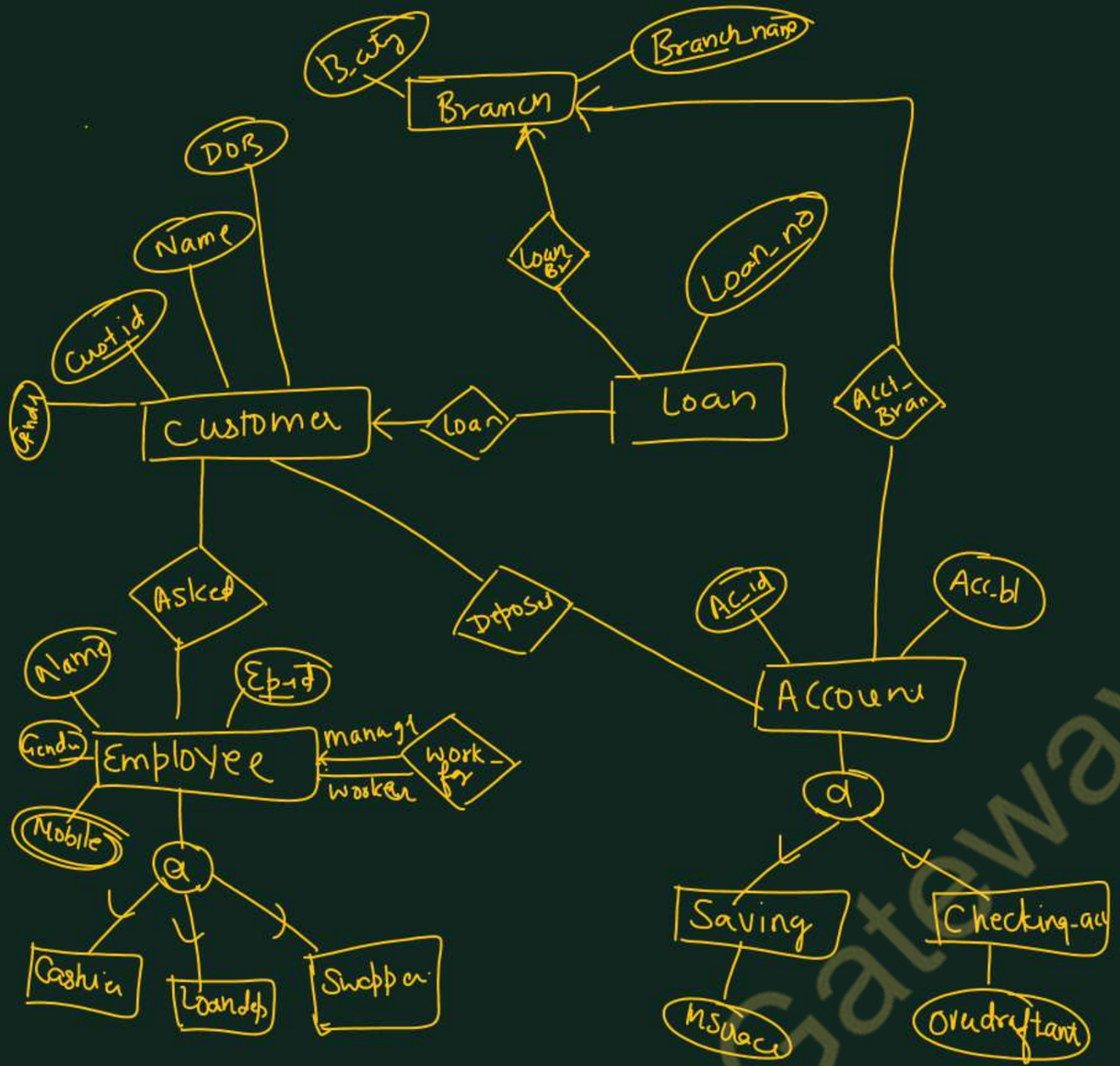


Eg P1 - M1 M2
Eg P2 - M1

AKTU QUESTIONS

Q.1	A database is being constructed to keep track of the teams and games of a sport league. A team has a number of players, not all of whom participate in each game. It is desired to keep track of players participating in each game for each team, the positions they play in that game and the result of the game. (i) Design an E-R schema diagram for this application **(ii) Map the ER diagram into relational model**	AKTU 2022-23
Q.2	ER Diagram for Employee Project Management System.	AKTU 2021-20





Banking System

AKTU QUESTIONS

Q.3	Construct an E-R diagram for a car insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents? Also convert the E-R diagram into tables?	AKTU 2018-19
Q.4	Draw an ER diagram of Hospital or Bank with showing the Specialization, Aggregation, Generalization.	AKTU 2017-18
Q.5	**ER diagram for any marketing company**	AKTU 2016-17

AKTU QUESTIONS

Q.6	A university registrar's office maintains data about the following entities: (a) courses, including number, title, credits, syllabus, and prerequisites; (b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom; (c) students, including student-id, name, and program; and (d) instructors, including identification number, name, department, and title. Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled. Construct an E-R diagram for the registrar's office. Document all assumptions that you make about the mapping constraints.	AKTU 2015-16