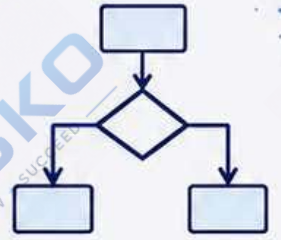


1010101010  
0101010101  
1010101010  
0101010101



# Notes





# Unit - I

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

S.A.

1) What is DBMS?

→ A DBMS is a software system that allows users and applications to create, retrieve, update, and manage data efficiently.

2) What is Relation?

→ A relation is a collection of tuples (rows/records) having the same set of attributes (columns/fields).

OR

A relation in DBMS is simply a table that stores data in the form of rows (tuples) and columns (attributes).

3) What is tuple and attribute?

→

- Attribute - An attribute is a column of a relation (table) that represents a property or characteristic of the data.

- Tuple - A tuple is a row of a relation (table) that represents a single record of data.

4) What is Domain?

→

A domain in DBMS is the set of valid values that an attribute (column) can take.

RDBMS

## Unit - I -

any two for  
classmate  
3 marks  
each

Date \_\_\_\_\_  
Page \_\_\_\_\_

Q1.\* Explain the characteristics of a database. Why is a DBMS preferred over file system?

→

Database -

A database is an organized collection of related data that is stored in a structured way so it can be easily accessed, managed and updated. It allows user and application to store large amount of ~~data~~ data and information, retrieve it quickly and maintain accuracy, security and consistency.

Characteristics of a database.

1. Shared data -

A database allows multiple users and applications to access the same data at the same time without conflict and system crashes.

2. Reduced data redundancy / Duplication.

In DBMS data is stored in a structured way so that the duplication of data is minimized.

### 3. Data Integrity -

Rules and constraints ensures that the data stored is accurate, consistent and reliable.

### 4. Data Security -

Only authorized users can access or modify data, which protects info from misuse.

### 5. Data Independency -

Changes in a data structure do not affect application programs, which makes maintenance easier.

### 6. Centralized control -

All data is managed from a central location, improving back up recovery and overall management.

Why a DBMS is preferred over a File System.

1. Better data consistency -

DBMS removes redundancy and keeps data consistence across the system while file systems often create Duplication and mismatched data.

2. Strong Security -

DBMS allows authentication, authorization and privileges. File systems cannot provide this level of control.

3. Backup and Recovery -

DBMS provides automatic backup and recovery mechanisms. In file systems, data recovery is difficult and manual.

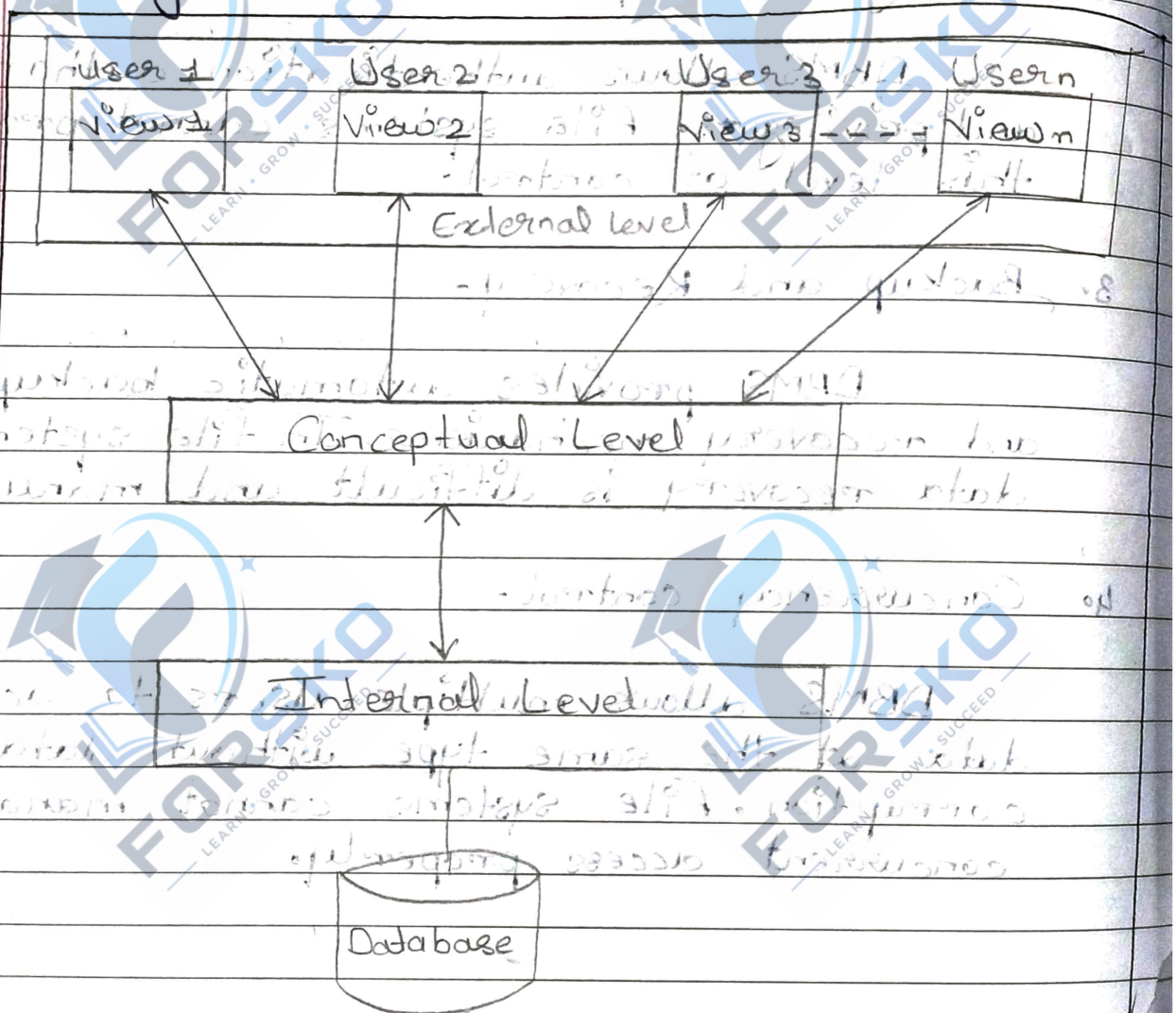
4. Concurrency control -

DBMS allows multiple users to access data at the same time without data corruption. File systems cannot manage concurrent access properly.

Q2.\* Describe the architecture of DBMS with neat explanation of each level (internal, conceptual, external).

→ The three level architecture defines how data is viewed, stored and managed inside a DBMS.

It separates the database into External Level, Conceptual Level and Internal Level. This separation provide data independence, security and easier management.



## ⇒ External Level - (View Level) -

- This is the topmost level of the architecture.
- It contains multiple user views and each view shows only the necessary part of the database for that user.
- It hides the rest of the database providing security and privacy.
- Different users have different views without affecting each other.

### Example -

⑤ A student may see only his roll number, marks and attendance and the teacher see all of this.

- It

## 2. Conceptual Level (Logical Level)

- This level provides a complete and ~~unified~~ logical view of the entire database.
- It describes what data is stored, the relationships among the data, and the constraints applied.
- It hides physical storage details and focuses on structure.
- Elements defined here include -
  - 1) Tables
  - 2) Attributes
  - 3) Primary & Foreign key
  - 4) Relationship
  - 5) Constraints (NOT NULL, UNIQUE)
- It ensures data consistency, integrity and centralized control.
- All external views derived from this level.
- It is the middle level of the architecture.
- It is the middle level of the architecture.

SHAHID, ABDUL

### 3. Internal Level (Physical Level) -

- This is the lowest level of the architecture
- It describes how data is actually stored in the system.
- It deals with -
  - File organization (sequential, index, hashed).
  - Storage block, pages and records
  - Indexes, Access paths, Compression & encryption.
  - Focus is on efficient storage, fast data access, and optimization
- User never interact this level directly it is controlled by the DBMS.

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

Q3. What is SQL? Discuss the components of SQL (DDL, DML, DCL, TCL) with examples for 6 marks.

→

SQL -

SQL (Structure Query Language) is a standard language used to store, manage, retrieve and manipulate data in a relational database.

It allows users to create tables, insert data, update records, control access and manage transactions in database.

Components of SQL -

⇒ DDL (Data Definition Language) -

DDL is used to define and modify the structure of database objects like tables, views and indexes.

These commands automatically commit changes.

DDL Commands include -

⇒ CREATE -

CREATE Command is used to create a new table and database objects.

Syntax -

CREATE TABLE table\_name (column-name data-type);

2) ALTER -

ALTER Command is used to change an existing table or database objects.

ALTER TABLE Employee ADD dept; varchar

3) DROP -

DROP command is used to delete table permanently.

DROP TABLE Employee;

4) TRUNCATE -

TRUNCATE removes all records but keeps the structure of the table.

TRUNCATE TABLE Employee;

## 2. DML (Data Manipulation Language)

DML commands are used to insert, update, delete, and retrieve data stored in the database.

These commands affect the actual data inside the table.

Important DML Commands —

### i) INSERT —

INSERT command is used to insert new records in a table.

INSERT INTO Emp VALUES (1, 'Ravi', 4500, 'HR');

table\_name VALUES (V1, V2, V3, V4, ..., Vn);

### ii) UPDATE —

UPDATE is used to modify existing records in a relation/table.

UPDATE Employee SET SALARY = 5000 WHERE EmpID = 1;

### iii) DELETE —

DELETE is used to remove records satisfied to the given condition permanently.

DELETE FROM Emp WHERE EmpID = 1;

DQL — (Data Query Language)

### iv) SELECT —

SELECT is used to retrieve data

SELECT Name, Salary FROM Emp;

### 3. DCL (Data Control Language) -

DCL commands control access, privileges & permissions on database objects. They help in managing security.

#### DCL Commands & Examples -

##### i) GRANT -

Give access rights to a user.

GRANT SELECT, INSERT ON Emp TO user1;

##### ii) REVOKE -

Removes previously granted rights

REVOKE INSERT ON Emp FROM user1;

classmate

Date \_\_\_\_\_  
Page \_\_\_\_\_

#### 4. TCL (Transaction Control Language)

TCL commands manage transactions which are groups of SQL operations performed as a single unit.

TCL ensures data consistency & help recover errors.

i) COMMIT - Saves all the changes made in the transaction.

COMMIT;

ii) ROLLBACK - SAVEPOINT - Creates a checkpoint for the back partially.

SAVEPOINT sp1; -- any 1 savepoint  
SAVEPOINT sp2; -- second savepoint

iii) ROLLBACK - Undo changes if a mistake happens.

ROLLBACK; - Remove mistakes

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

ASC / DESC

2. ORDER BY Clause - Ascending / Descending

ORDER BY is used to sort the result in ascending (ASC) or descending (DESC) order.

If we not specified, SQL sort in ascending order by default.

Eg-

```
SELECT Name, Salary / SELECT column, column
FROM Employee FROM table name
ORDER BY Salary DESC; ORDER BY ASC;
OR
DESC;
```

This arranges the employees from highest salary to the lowest salary.

3. GROUP BY Clause -

GROUP BY Clause is used to group rows which have the same value in one or more columns.

It is mainly used with aggregate function like COUNT, SUM, AVG, MAX, MIN.

Eg-

```
SELECT Department, AVG(Salary) AS
```

FROM Employee

GROUP BY Department;

This groups employees by Dept and shows the avg salary

#### 4. HAVING Clause.

WHERE filters individual rows, but HAVING filters groups created by GROUP BY.

HAVING is used when we want to apply a condition on aggregated results.

Eg-  

```
SELECT Department, COUNT(*) AS
TotalEmployees
FROM Employee
GROUP BY Department
HAVING COUNT(*) > 5;
```

This shows only those departments where more than 5 employees work.

#### Combined Example-

```
SELECT Department, AVG(Salary)
FROM Employee
WHERE Salary > 30000
GROUP BY Department
HAVING AVG(Salary) > 40000
ORDER BY AVG(Salary) DESC;
```

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

Having Clause is used to filter groups based on the condition after GROUP BY clause is applied.

It is similar to WHERE but it operates on group result rather than individual rows.

Syntax -

```
SELECT column1, agg-fun(column2)
FROM table-name
GROUP BY column1
HAVING condition;
```

```
SELECT department, COUNT(*) AS total employee
FROM Employee
GROUP BY department
HAVING COUNT(*) > 5;
```

## Unit - II

any one  
6 marks  
from two.

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

Q4.\* What are constraints? Explain Primary key, Foreign key, Unique, Check, Not NULL with example.

→ Constraints -

Constraints are rules applied on table columns to maintain accuracy and consistency of data in a database. They prevent invalid data from being inserted, updated, or deleted.

Constraints help enforce business rules at the database level.

Common constraints are - Primary key, Foreign key, Unique, Check and Not NULL.

1. Primary Key Constraint -

A primary key uniquely identifies each row in a table.

→ It cannot contain NULL values.

→ It must contain unique values.

→ A table can have only one primary key.

Eg- CREATE TABLE Students (  
RollNo INT PRIMARY KEY,  
Name VARCHAR (30),  
Age INT  
);

Here, RollNo is the primary key.

No two students can have the same RollNo and it can not be left empty.