22/8/23

ASSIGNMENT 1

Unit-1&2 Introduction to DBMS and Relational Model & SQL

Q.1 List and explain the advantages of DBMS over file based system. Explain disadvantages of file based system.

→ Advantages of DBMS over File Based System:

i) Data sharing and Concurrency Control

DBMS enables multiple users to access and manipulate the data concurrently while managing potential conflicts.

ii) Data Query and Reporting

DBMS supports powerful querying capabilities through query languages like SQL. This allows users to retrieve specific subsets of data quickly and efficiently, enabling complex data analysis and reporting.

iii) Data Recovery and Backup

DBMS offers built-in mechanisms for data backup, recovery and rollback in case of system failures. This ensures that data can be restored to a consistent state even after unexpected events.

iv) Maintenance and Updates.

DBMS simplifies maintenance tasks such as adding, updating or deleting data by providing structured interfaces.

v) Scalability and Performance.
DBMS systems are designed to handle large amounts of data and can scale to ~~accomod~~ accommodate increasing data volumes. Additionally, they optimize data storage and retrieval for improved performance compared to file-based systems.

→ Disadvantages of File-Based Systems:

i) Data Redundancy and Inconsistency
Here, it often lead to data redundancy, where the same data is stored in multiple files, which results in inconsistencies when updates are made to one copy but not to other copies.

ii) Data isolation and Fragmentation
Data is isolated in separate files which makes it challenging to access and share data across different parts of an organization.

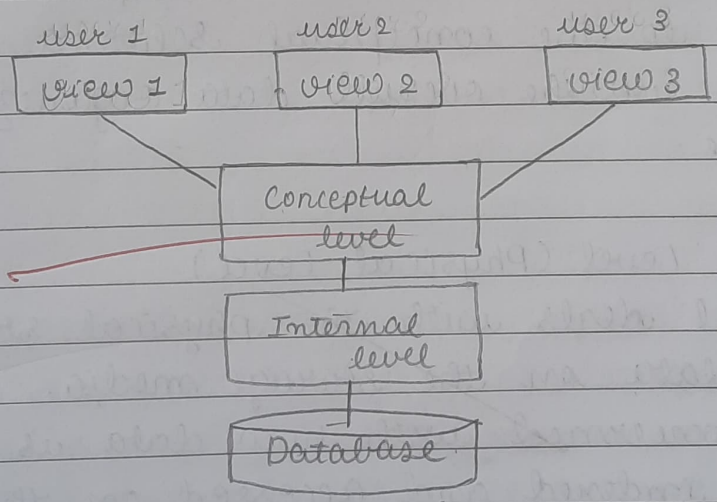
iii) Scalability Issues
As data grows, file-based systems can become less efficient and harder to manage. Adding new data and maintaining data consistency becomes more challenging

iv) Limited Data Relationships
Establishing relationships between data in

file-based systems which requires manual effort and maintaining these relationships can be error-prone.

v) Lack of Data Backup and Recovery.
File-based systems often lack automated backup and recovery mechanisms, making data restoration after failures, difficult and time-consuming.

Q.2 Draw and explain 3 level architecture of DBMS.
—> The Three level architecture is a concept for designing and organizing which divides the DBMS into three distinct levels.



i) External level (View level)
> This level represents the perspective of individual users or group of users
> It deals with the way data is presented to and accessed by different user groups.
> Here it provides tailored and customized

views of the data according to the needs of different user.

- Users interact with the database through queries and transactions based on their defined views.

ii) Conceptual level (Logical level)

- This level provides an abstract, high-level view of entire database.
- It focuses on defining the overall structure and organization of the data, independent of any spef specific user requirements or physical storage considerations.
- This is level is defined using high-level data model (ERM) and a data manipulation language.
- Changes to the conceptual schema will reflect changes in the overall data organization and structure.

iii) Internal level (Physical level)

- This level deals with the physical storage details of the data on the storage media.
- It is concerned with how data is actually stored, indexed and accessed on the physical storage devices.
- DBMS translates operations from conceptual level into instructions that the physical storage system can understand and execute.
- Changes at this level should not affect the above two levels, allowing for flexibility in adapting

Q.3 list and explain different categories / types of database users.

-4 i) Database Implementors
- They are responsible for designing and creating the database system.
- They decide the type of DBMS to use, design the overall database schema and configure hardware and software components.
- Their focus is on the initial setup and installation of the database.

ii) Database Designers
- They are responsible for designing the structure and organization of database schema.
- They analyse user requirements, defines entities, relationships, attributes and constraints and create the conceptual and logical database design.
- Their main focus is to ensure efficient data storage, data integrity.

iii) Database Administrators (DBAs)
- ~~They~~ are responsible for overall management, maintenance and performance of database system.
- They manage user access and security, optimize query performance and backup / recovery procedures.
- They identify and resolve and implement database related policies and procedures.

iv) Application Programmers
- They develop software applications that interact with the database to perform specific tasks
- They write code to implement business logic, data processing and data manipulation required by the application.
- ~~Pro~~ They use APIs and query language to communicate with the database system from within their applications.

v) End Users
- They are the individual ~~which~~ who indirectly interact with the database to retrieve, update and manipulate data.
- They include both users who perform queries and users who interact with system for their work.
- They use various tools, forms and reports to access and manage data without in-depth knowledge of the structure.

Q.4 list and explain different tasks / roles / functions / duties of DBA (Database Administrator).

→ DBA plays a crucial role in the management and maintenance of a database system.

i) Database Installation and Configuration.
DBA install and configure the database management system software on appropriate hardware and operating systems.

ii) Database Design and Planning
DBA participate in the initial design and planning of the system to create a well-structured and efficient database schema

iii) Security Management
DBA are responsible for implementing and maintaining security measures to protect sensitive data from unauthorized access.

iv) Data Backup and Recovery
DBA manage backup and recovery procedures to ensure data availability in case of system failures, human errors or disasters.

v) Performance Tuning and Optimization
DBA monitors the performance of the system and identify areas where improvements are needed to ensure the system operates efficiently

vi) Capacity Planning
DBA forecast future data growth and usage pattern to ensure that the system can handle the increasing load

vii) Database Upgrades and Patches
DBA manages the process of updating the DBMS to newer versions or applying patches and updates

Q.5 Explain Database System architecture with block diagram.

→ i) Query Processor : It interprets the queries received from end user via an application program into instructions.

① DML Compiler : Processes the DML statements into low level instruction (machine language)

② DML Interpreter : Processes the DDL statements into a set of table containing data about data.

③ Query Optimizer : Executes the instruction generated by DML Compiler.

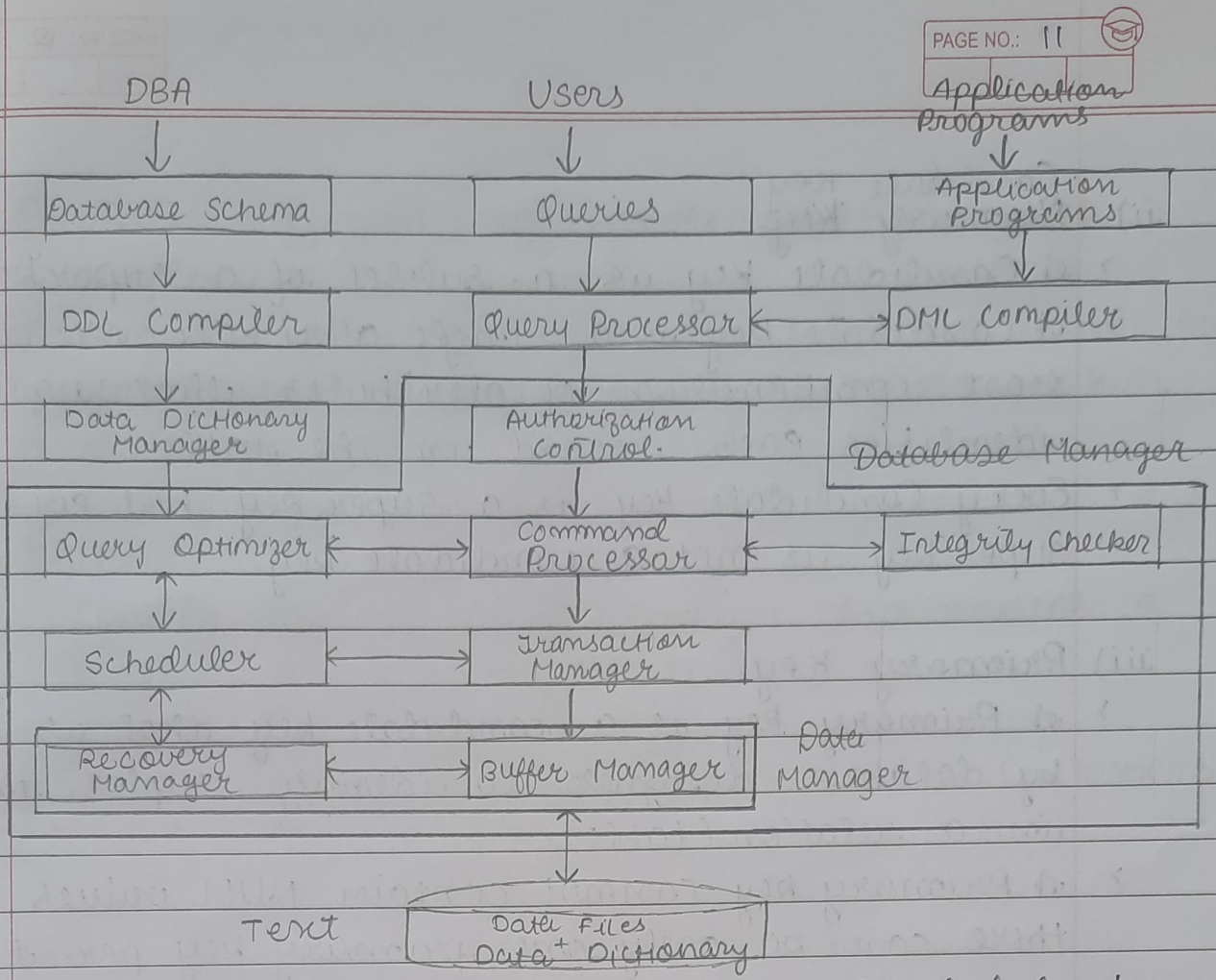
④ Embedded DML Pre-Compiler

ii) Storage Manager : A program that provides an interface between the data stored in the database and the queries received.

① Authorization Manager
② Integrity Manager
③ Transaction Manager
④ File Manager
⑤ Buffer Manager

iii) Disk Storage : Contains following components

① Data files : stores the data.
② Data Dictionary : contains information about the structure of any database object.
③ Indices : provides faster retrieval of data item.



Q.6 Define Super key, Primary key, Candidate key and alternate key in detail with example.

→ Keys are used to identify and represent unique records (rows) within the table.

i) Super Key

> A super key is a set of one or more attributes that can be used to uniquely identify tuples within a table.

> All those attributes in a table that is capable of identifying other attributes of the table in a unique manner are all super keys.

> A super key may contain extraneous attributes that aren't strictly necessary for uniqueness.

ii) Candidate Key

- A Candidate key is a subset of a superkey.
- A Candidate key is a single attribute or the least combination of attributes that uniquely identifies each record in the table.
- Every Candidate Key is a super key but every super key is not a candidate key.

iii) Primary Key.

- A Primary key is a candidate key that is chosen by database designer to identify tuples uniquely in a relation (table).
- A Primary key cannot contain NULL values and there can be only one primary key per table.

iv) Alternate Key.

- An Alternate key is a candidate key that is not chosen as the primary key but it still has the property of uniqueness.

Example: A "Employee" table with attribute:
employeeID, SSN, Name, email

- Super Keys: {employeeID}, {SSN}, {email}, {SSN, email}, {employeeID, Name}
- Candidate Keys: {employeeID}, {SSN}
- Primary key: {employeeID}
- Alternate Key: {SSN}

Q.7 Explain different aggregrate functions with example.

→ Aggregrate functions are SQL functions used to perform calculations on set of values within a database table.

i) COUNT()
Counts the number of rows that match a specific condition. g count (relation)

eg: > SELECT COUNT(*) AS TotalCustomers FROM Customers WHERE City = 'New York';

ii) SUM()
Calculates the sum of numeric values in a column. g sum (relation)

eg: > SELECT SUM(Amount) AS TotalSales FROM Sales WHERE ProductID = 123;

iii) AVG()
Computes the average of numeric values in a column g avg (relation)

eg: > SELECT AVG(Age) AS AverageAge FROM Employee;

iv) MIN()
Fetch the minimum value from a column.

eg: > SELECT MIN(Price) AS MinPrice FROM Products;

v) MAX()
Fetch the maximum value from a column

eg: > SELECT MAX(Price) AS MaxPrice FROM Products;

Q.8 Explain different types of operations in relational algebra.

→ Relational algebra is a formal query language used to manipulate and retrieve data from relational databases.

- Unary Operations :-

i) Selection (σ)
This operation retrieves rows that satisfy a specified condition and it is denoted by the Greek letter 'σ' and is similar to the SQL 'WHERE' clause

eg: σ(JobTitle = 'Manager')(Employee)

ii) Projection (π)
This operation retrieves specified columns from a table while eliminating duplicates & it is denoted by 'π'.

eg: π(Name, Salary)(Employee)

iii) Set Difference (-)
This operation returns the rows that exist in one table but not in another. It is used

to subtract the tuples of one relation from another.

eg: Employee - σ (JobTitle = 'Manager') (Employee)

- Binary Operations :-

i) Union (∪)
This operation combines rows from two tables, eliminating duplicates.

eg: Employee ∪ Customer

ii) Intersection (∩)
This operation retrieves common rows present in both input ~~fields~~ tables.

eg: Employee ∩ Customer

iii) Cartesian Product (×)
This operation combines all rows from one table with all rows from another table, creating a new table.

eg: Products × Suppliers

iv) Natural Join (⋈)
This operation combines rows based on matching attributes & eliminate duplicate columns.

eg: Orders ⋈ Customer

Q.9 Explain DDL and DML commands in detail.

—↓ DDL and DML commands are two categories of SQL commands used to manage and interact with the database.

i) DDL (Data Definition Language)

These commands are used to define, modify and manage the structure of the database objects. They focus on creating and altering database schema elements such as tables, indexes, constraints and views & are also not concerned with the actual data.

① CREATE TABLE : Used to create a new table, with specified columns, data types, constraints

eg:
```
CREATE TABLE SUPPLIER (
    ID INT PRIMARY KEY,
    NAME VARCHAR(25)
    );
```

② ALTER TABLE : Used to modify an existing table's structure by adding, modifying or dropping column and constraints.

eg:
```
ALTER TABLE SUPPLIER
ADD COLUMN EMAIL VARCHAR(30);
```

③ DROP TABLE : Used to delete an existing table and its associated data permanently from database

eg: DROP TABLE SUPPLIER;

④ TRUNCATE TABLE : Used to ~~create~~ remove all the records from a object while keeping the structure.

eg: TRUNCATE TABLE SUPPLIER;

⑤ RENAME TO : Used to rename the object in the database.

eg: ALTER TABLE SUPPLIER
RENAME TO EMPLOYEE;

iii) DML (Data Manipulation Language)
These commands are used to interact with the data stored in the database tables. They allow you to insert, update and delete data from the tables.

① SELECT : Used to fetch data from one or more tables based on specified conditions and returns the result set.

eg: SELECT NAME
FROM STUDENT
WHERE ID = 101;

② INSET INTO : Used to insert new rows into the table with specified values for columns.

eg:
```
INSERT INTO STUDENT (ID, NAME)
VALUES (101, 'JOHN');
```

③ UPDATE : used to modify existing data in a table by changing values of specific columns based on certain conditions.

eg:
```
UPDATE STUDENT
SET MARKS = 97
WHERE ID = 101;
```

④ DELETE : used to remove rows from a table based on specified conditions

eg:
```
DELETE FROM STUDENT
WHERE ID=101;
```

Q.10 Explain DCL and TCL commands in detail.

DCL and TCL commands are also another two categories of SQL commands that deal with controlling access to data and managing transactions.

i) DCL (Data Control Language)

These commands are used to control the access and permission of database users and manage the security aspects of database.

① GRANT : Grants specific permissions to users, allowing them to perform certain actions on

specified database objects.

eg: GRANT SELECT, INSERT ON EMPLOYEES TO Analyst;

② REVOKE : Removes specific permissions from users, revoking their ability to perform certain actions on database objects.

eg: REVOKE DELETE ON CUSTOMER FROM CLERK;

iii) TCL (Transaction Control Language)

These commands are used to manage transactions, ensuring the consistency and integrity of data changes made within a transaction.

① COMMIT : Finalizes and permanently applies all changes made within the current transaction to the database.

eg: COMMIT;

② ROLLBACK : Undoes all changes made within the current transactions, restoring the database to its state before the transaction started

eg: ROLLBACK;

③ SAVEPOINT : Creates a savepoint within a transaction allowing you to roll back to a specific point in the transaction without affecting entire transaction

eg: SAVEPOINT St1;

Q.11 What is data independence?

—> Data independence refers to the separation between the physical storage details of data and the logical view that users have of that data. It is typically categorized into two levels:

i) Physical Data Independence.

> It refers to the ability to modify the physical storage characteristics of the data without affecting the way it is accessed and manipulated by users.

> Changes to the storage devices, file structures, indexes and other physical storage components should not impact higher level data operations.

> Modification at the internal levels are occasionally necessary to improve performance.

ii) Logical Data Independence

> It refers to the ability to change the logical structure of the data without affecting the way users interact with the data.

> Changes to the table structures, relationships, attributes and constraints should not require changes to the queries that access the data.

> Modification at the logical levels is necessary whenever the logical structure of the database is change.

Q.12 Explain different types of joins with suitable example.

→ Joins are used to combine data from two or more tables based on related columns.

i) Natural Join (Inner Join)

It returns only the rows where there's a match between the specified columns in both tables.

eg:
```
SELECT Order.ID, Order.Total, Customer.Name, Customer.City
FROM Order
INNER JOIN Customer ON Order.Name = Customer.Name;
```

ii) Outer Join

A method of combining two or more tables so that the result includes unmatched rows of one of the tables or both tables. There are further three parts:

① Left Outer Join

Returns all the rows from the left table and the matching rows from the right table. If no match, null values are return for right table columns

② Right Outer Join

Returns all rows from the right table and the matching rows from left table. If no match, null values are return for left table columns.

③ Full Outer Join.
Returns all rows from both tables, including matching rows and rows without matches. Null values are returned for column without matches.

eg:
```
SELECT Order.ID, Order.Total, Customer.Name, Customer.City
FROM Order
FULL JOIN Customer ON Order.Name = Customer.Name;
```

Q.13 What is View and Role?

—↓ View

> A View is a virtual table that is derived from one or more existing tables in a database.
> Views allows you to abstract complex queries, provide a simplified interface to users and enforce security by restricting access to certain column or rows of data.
> Views combine data from multiple tables and present a unified result.
> Views do not store itself physical data; instead they retrieve data from their base tables.

+ Role eg: CREATE VIEW EmployeeInfo AS
SELECT EmployeeID, Name, JobTitle
FROM Employees;

—↓ Role.

> A Role is a named group of related privileges that can be assigned to users within a database.

> Roles are used to manage access control and permissions.
> By assigning roles to users, you can control what actions they can perform and what objects they can access in the database.
> Roles provide a higher level of abstraction for managing user permissions.

eg: CREATE ROLE Admin;
GRANT SELECT, INSERT, UPDATE, DELETE TO Admin;

CREATE ROLE Customer;
GRANT SELECT TO Customer;

Q.14 Consider the following schema for college library management system:
Student (Roll_No, Name, Branch)
BOOK (ISBN, Title, Author, Publisher)
Issue (Roll_No, ISBN, Date_of_Issue). Write SQL queries for the following statements

i) List Roll Number and Name of all students of the branch IT:

→ SELECT Roll_No, Name
FROM Student
WHERE Branch = 'IT';

ii) Find the name of students who have issued a book published by 'XYZ' publisher:

→ SELECT DISTINCT S.Name
FROM Student S
JOIN Issue i ON S.Roll_No = i.Roll_No

```
JOIN Book b ON i.ISBN = b.ISBN
WHERE b.Publisher = 'XYZ';
```

iii) list title of all books and their author issued by student 'Alice':

—↓

```
SELECT b.Title, b.Author
FROM Book b
JOIN Issue i ON b.ISBN = i.ISBN
JOIN Student s ON i.Roll_No = s.Roll_No
WHERE s.Name = 'Alice';
```

iv) list title of all books issued on or before 31st dec 2022:

—↓

```
SELECT b.Title
FROM Book b
JOIN Issue i ON b.ISBN = i.ISBN
WHERE i.Date_of_issue <= '31-12-2022';
```

Q.15 Consider the following relations and write a relational algebra:

EMP (empno, ename, jobtitle, managerno, hiredate, sal, commission, deptno)

DEPT (deptno, dname, location):

i) Find the Employees working in the department number 10, 20, 30 only:

—↓ $\Pi_{ename}(\sigma_{deptno=10 \vee deptno=20 \vee deptno=30}(EMP))$

ii) Find Employees whose name start with letter A or a:

—↓ $\Pi_{ename}(\sigma_{ename\ LIKE\ 'A\%' \vee ename\ LIKE\ 'a\%'}(EMP))$

iii) Find Employees along with their department name:
→ EMP ⋈ (DEPT ⋈ π deptno, dname (DEPT))

iv) Find the Employees who are working in Smith's department
→ π empno, ename (σ ename = 'Smith' (EMP ⋈ DEPT))

v) Find the Employees who get a salary more than Allen's salary:
→ π empno, ename (σ sal > (π sal (σ ename = 'Allen' (EMP))))

vi) Display employees who are getting the maximum salary in each department:
→ EMP ⋈ (π deptno, MAX(sal) (EMP))

vii) Find a list of employees whose hire date is on or before 1-April-18:
→ π empno, ename (σ hiredate <= '01-04-2018' (EMP))

Q.16 Consider the following relations and write an SQL query
~~EMPLOYEE~~ Employee (Eid, Ename, Ph-no, joining-date, Salary, dept-id)
Department (dept-id, dept-name, location, manager-id)

i) Find all those employees who worked in the Sales department.
→
```
SELECT Ename
FROM Employee
WHERE dept-id IN (SELECT dept-id FROM Department WHERE
                  dept-name = 'Sales');
```

ii) Find the employee name whose salary is the lowest salary of any department:

```
SELECT Ename
FROM Employee
WHERE salary = (SELECT MIN(Salary) FROM Employee);
```

iii) Find out the employees who get a higher salary.

```
SELECT Ename
FROM Employee
WHERE salary > (SELECT MAX(Salary) FROM Employee);
```

iv) Find employees whose salary is more than the average salary and work in the same department as an employee whose first name contains the letter 'v':

```
SELECT E.Ename
FROM Employee E
WHERE Salary > (SELECT AVG(salary) FROM Employee)
  AND dept_id = (SELECT dept_id FROM Employee
                 WHERE Ename LIKE '%v%' LIMIT 1);
```