

Assignment 1: Term Test I Solution

Submitted By:

Name: Kushal Prasad Joshi

Roll No.: 230345

Faculty: Science and Technology

Group: B

Submitted To:

Name: Nisha Thapa

Department: Information and Communication Technology

Assignment Given: 11th Poush 2081

Submission Date: 18th Poush 2081

1. What is DBMS. State advantages of DBMS over the file processing System.

A Database Management System (DBMS) is software designed to store, manage, and retrieve data efficiently while ensuring security, consistency, and integrity. It provides a systematic way of creating, managing, and interacting with databases. Examples of DBMS include MySQL, PostgreSQL, Oracle Database, and Microsoft SQL Server.

The advantages of DBMS over the file processing system are:

Advantage	File Processing System	DBMS
Data Redundancy and Inconsistency	Data redundancy occurs because data is stored in multiple files, leading to inconsistencies.	Reduces redundancy through normalization and centralizes data storage, ensuring data consistency.
Data Integrity and Accuracy	Maintaining data integrity is challenging as there is no centralized control over data validation.	Ensures data integrity using constraints like primary keys, foreign keys, and unique constraints.
Data Security	Offers minimal security; unauthorized users can easily access files.	Provides robust security through user authentication, roles, and permissions.
Concurrency Control	Difficult to manage simultaneous access by multiple users, leading to conflicts.	Supports multi-user environments with concurrency control mechanisms, ensuring data consistency.
Data Backup and Recovery	Limited or no automated backup and recovery options.	Provides automatic backup and recovery solutions, minimizing data loss in case of failures.
Query Processing	Requires manual coding to retrieve data.	Allows users to retrieve and manipulate data using powerful query languages like SQL.
Scalability and Flexibility	Not easily scalable as data grows.	Can handle large datasets and adapt to growing data requirements.

Data Sharing	Sharing data between different users is complex and inefficient.	Facilitate data sharing among users while maintaining security and consistency.
--------------	--	---

2.

a. Define the role of relational algebra and its associated basic operators with necessary notations.

Relational algebra is a procedural query language used to query and manipulate relational databases. It provides a foundation for SQL and acts as the theoretical basis for relational database operations. Relational algebra defines a set of operations on relations (tables) to retrieve or derive new relations from existing ones.

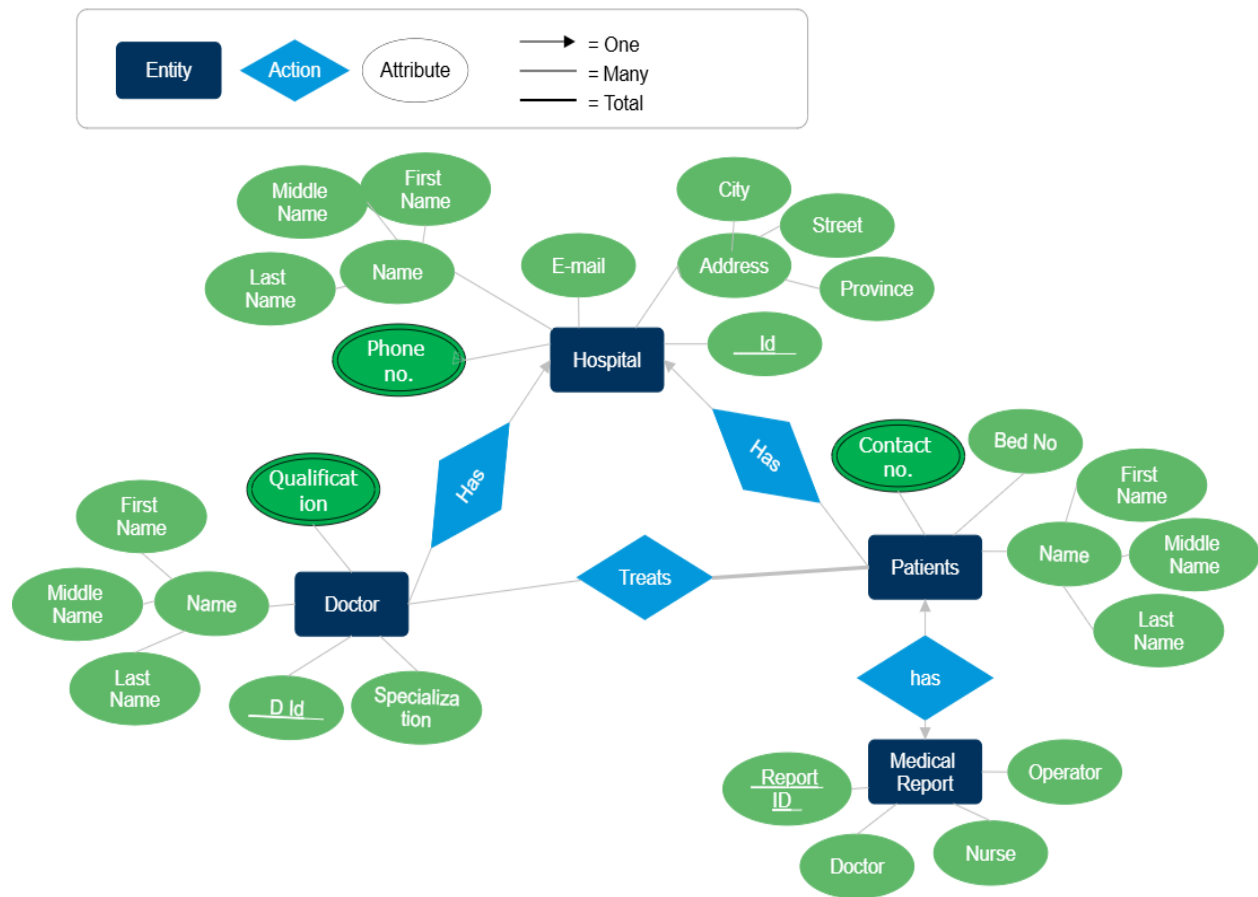
The basic operators associated with relational algebra are:

Operator	Symbol/Notation	Description
Selection (σ)	$\sigma_{\text{condition}}(R)$	Selects rows (tuples) from a relation (R) that satisfy a specified condition.
Projection (π)	$\pi_{\text{attributes}}(R)$	Selects specific columns (attributes) from a relation.
Union (U)	$R \cup S$	Combines tuples from two relations (R and S) into a single relation, removing duplicates.
Set Difference (-)	$R - S$	Finds tuples in relation R that are not in relation S.
Cartesian Product (\times)	$R \times S$	Combines each tuple of relation R with every tuple of relation S.
Rename (ρ)	$\rho_{\text{new_name}}(R)$	Renames a relation or its attributes to simplify or clarify queries.
Intersection (\cap)	$R \cap S$	Retrieves tuples that are common to both relations R and S.

Division (\div)	$R \div S$	Finds tuples in R that are related to all tuples in S.
Natural Join	$R \bowtie S$	Combines related tuples from two relations based on common attributes, removing duplicates.
Theta Join	$R \bowtie_{\theta} S$	Combines tuples from two relations based on a condition θ (e.g., $R.A = S.B$).
Left Outer Join	$R \Joinleft S$	Retrieves all tuples from the left relation and matches tuples from the right relation.
Right Outer Join	$R \Joinright S$	Retrieves all tuples from the right relation and matches tuples from the left relation.
Full Outer Join	$R \Joinfull S$	Combines tuples from both relations, including unmatched tuples from both sides.

b. Draw an ER diagram for hospital management system. (Use DOCTOR, PATIENT, HOSPITAL and MEDICAL_RECORD Entity)

Entity Relationship Diagram - Hospital Management System



3. Define normalization and state the advantages of normalization. Normalize database: Employee (emp_id, emp_name, phone, skill, salary, dept no, dept_name, jobno, job_title) upto 3NF.

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. This involves dividing a database into two or more tables and defining relationships between the tables to minimize data anomalies.

The advantages of normalization are: eliminates data redundancy, improves data integrity, facilitates maintenance, optimizes query performance, ensures data dependence, etc.

0NF: Given Table

Employee

Emp_id	Emp_name	Phone	Skill	Salary	Deptno	Dept_name	Jobno	Job_title

1NF: Ensure atomic values

Employee

Emp_id	Emp_name	Salary	Deptno	Dept_name	Jobno	Job_title

Phone

Emp_id	Phone

Skill

Emp_id	Skill

2NF: Remove partial dependencies

Employee

Emp_id	Emp_name	Salary

Phone

Emp_id	Phone

Skill

Emp_id	Skill

Department

Deptno	Dept_name

Job

Jobno	Job_title

3NF: Remove transitive dependencies

Employee

Emp_id	Emp_name	Salary

Phone

Emp_id	Phone

Skill

Emp_id	Skill

Department

<u>Deptno</u>	Dept_name

Job

<u>Jobno</u>	Job_title

4. Consider the following relations:

EMP (empno, deptno, ename, salary, city)

Write SQL query for the following cases:

- I. Display employee number and name in an increasing order of salary.**

```
SELECT empno, ename  
FROM EMP  
ORDER BY salary ASC;
```

- II. Display employee name starting with “S” and working in deptno 105.**

```
SELECT ename  
FROM EMP  
WHERE ename LIKE 's%'  
AND deptno=105;
```

- III. Delete all employees of department 100.**

```
DELETE  
FROM EMP  
WHERE deptno=100;
```

- IV. Display number of employees department wise.**

```
SELECT deptno, COUNT(*)  
FROM EMP  
GROUP BY deptno;
```

- V. Insert the new employee as 500, 102, rohit, 50000, kathmandu.**

```
INSERT INTO EMP  
VALUES  
(500, 100, 'rohit', 50000, 'kathmandu');
```

5. Write short notes on: (any two)

a. Keys

In databases, keys are used to identify rows in a table uniquely. They ensure the integrity of the data and help establish relationships between tables.

The different types of keys are:

Primary Key: A unique identifier for a record on a table. It cannot have NULL values and must be unique across all rows.

Foreign Key: A field in one table that refers to the primary key in another table, establishing a relationship between the two.

Unique Key: Like a primary key but allows a NULL value and can be used in fields where uniqueness is required but not necessarily as the table's primary key.

Composite Key: A key that consists of two or more columns to uniquely identify a record in a table.

Candidate Key: A set of columns that can be used as a primary key, where any of them could be chosen as the primary key.

b. Integrity Constraints

Integrity constraints are rules that ensure the accuracy and consistency of data in a database.

Different types of integrity constraints are:

Entity Integrity: Ensures that each table has a primary key and that the primary key value is unique and not NULL.

Referential Integrity: Ensures that a foreign key value in a child's table corresponds to a valid primary key value in a parent table.

Domain Integrity: Ensures that values in a column fall within a predefined range or set of permissible values (such as data type, format, or range).

User-Defined Integrity: Enforces custom rules specific to the business logic of the database, beyond standard entity, referential, and domain integrity.

c. Aggregate Functions

Aggregate functions are used to perform calculations on a set of values and return a single result.

Some aggregate functions are:

COUNT(): Returns the number of rows that match a specified condition.

SUM(): Calculates the total sum of a numeric column.

AVG(): Calculates the average value of a numeric column.

MIN(): Returns the smallest value in a column.

MAX(): Returns the largest value in a column.

GROUP_CONCAT(): Concatenates values from multiple rows into a single string (in some SQL databases).

FIRST() and LAST(): Return the first and last value in a column (depending on the database).
