Comprehensive Study Notes

Relational Database Management System

(Unit 4)

Class XI — Information Technology (Code 802)

Prepared by:
Ashish Kumar Bhoi

Contents

1 Relational Database Management System (RDBMS) Concepts **2**
- 1.1 Data, Information, and Big Data . . . 2
- 1.2 Database Management System (DBMS) & Benefits . . . 2
  - 1.2.1 Key Advantages of DBMS over File Systems . . . 3
- 1.3 The Relational Database Model . . . 3
- 1.4 RDBMS Terminology & Concepts . . . 3
- 1.5 Keys in Relational Databases . . . 3

2 Introduction to MySQL & Database Setup **5**
- 2.1 Overview of MySQL . . . 5
  - 2.1.1 Key Characteristics of MySQL . . . 5
- 2.2 Installing and Configuring MySQL on Windows . . . 5
- 2.3 SQL Command Classification . . . 5
- 2.4 MySQL Data Types . . . 6

3 Data Definition Language (DDL) Commands **7**
- 3.1 Database Level Commands . . . 7
- 3.2 Table Creation & Constraints . . . 7
  - 3.2.1 Common Table Constraints . . . 7
  - 3.2.2 Creating Tables with Constraints . . . 7
- 3.3 Inspecting Database Objects . . . 8
- 3.4 Altering Table Structure (`ALTER TABLE`) . . . 8
- 3.5 Deleting Tables (`DROP TABLE`) . . . 8

4 Data Manipulation Language (DML) Commands **9**
- 4.1 Inserting Data (`INSERT INTO`) . . . 9
- 4.2 Querying Data (`SELECT`) . . . 9
- 4.3 Modifying Records (`UPDATE`) . . . 10
- 4.4 Deleting Records (`DELETE`) . . . 10

5 Comprehensive Question Bank **11**
- 5.1 Multiple Choice Questions (MCQs) . . . 11
- 5.2 Assertion-Reasoning Questions . . . 13
- 5.3 One-Word / Brief Answer Questions . . . 14
- 5.4 Differentiate Between Questions . . . 15
- 5.5 Short Answer Questions . . . 16
- 5.6 Long Answer & SQL Query Questions . . . 17
- 5.7 Case-Study & Competency-Based Questions . . . 18

1 Relational Database Management System (RDBMS) Concepts

1.1 Data, Information, and Big Data

Data vs. Information:

- **Data:** A collection of raw, unorganized, uninterpreted facts, figures, numbers, and symbols. By itself, data lacks context and specific meaning (e.g., `101`, `"Rahul"`, `95.5`).
- **Information:** Data that has been processed, organized, structured, and presented within a specific context to convey clear meaning and utility (e.g., `"Student Rahul (Roll 101) scored 95.5% in IT"`).

Need for Computerized Database Management: In today's era of **Big Data**, vast volumes of structured and unstructured information are generated rapidly across organizations (airports, defense, police forces, hospitals, educational institutions, and financial institutions). Traditional paper-based file systems fail to scale, suffer from physical decay, introduce human calculation errors, and make data retrieval exceptionally slow.

Computerized databases resolve these challenges by providing high-speed storage, instant searchability, concurrent multi-user access, and automated data processing. For instance, a school library database records book stock details, active borrowers, issue dates, and automated due-date return alerts.

1.2 Database Management System (DBMS) & Benefits

Definition: A **Database Management System (DBMS)** is a specialized software program that enables users to create, maintain, query, control, and administer computerized databases. Popular DBMS implementations include MySQL, PostgreSQL, Oracle, Microsoft SQL Server, Microsoft Access, IBM DB2, and Sybase.

Key Core Capabilities of a DBMS

A DBMS provides mechanisms to:

- Store data in a structured, standard format.
- Query the database to retrieve specific subsets of data.
- Sort, filter, update, and manipulate stored records.
- Validate data input using predefined business logic and constraints.
- Generate customizable visual reports for decision-making.

1.2.1 Key Advantages of DBMS over File Systems

1. **Control of Data Redundancy:** Traditional file systems store duplicate copies of the same data across multiple departmental files. A DBMS centralizes data, minimizing unnecessary duplication.

2. **Data Consistency:** By eliminating redundancy, data inconsistency is automatically prevented. Updating an entry in a centralized DBMS instantly reflects across all linked views.
3. **Data Sharing:** Authorized multiple users and applications can access and query the same database concurrently.
4. **Enforcement of Security Restrictions:** Access controls, role-based authorization, and user passwords protect sensitive fields from unauthorized viewing or modification.
5. **Data Integrity Validation:** Enforces validation rules (constraints) to ensure data accuracy (e.g., age must be positive, marks must be between 0 and 100).
6. **Backup and Recovery Services:** Automated transaction logging prevents data loss during hardware failures or power outages.

1.3 The Relational Database Model

In a **Relational Database Management System (RDBMS)**, data is stored across separate two-dimensional tables (called **Relations**) that are linked together through shared common columns. Introduced by E.F. Codd in 1970, the relational model simplifies data representation while ensuring data integrity.

1.4 RDBMS Terminology & Concepts

- **Table / Relation:** A two-dimensional grid consisting of rows (tuples) and columns (attributes).
- **Record / Tuple / Row:** A horizontal row in a relation representing a single, complete entity record (e.g., details of one specific student or employee).
- **Field / Column / Attribute:** A vertical column in a relation containing values of a specific data attribute for all records. Each column has a unique name and uniform data type.
- **Domain:** The set or pool of all valid, permissible values from which an attribute derives its values (e.g., Domain for `Gender` column = `{'M', 'F', 'Other'}`).
- **Degree:** The total number of columns (attributes) in a relation.
- **Cardinality:** The total number of rows (tuples) in a relation.

1.5 Keys in Relational Databases

Keys are columns or combinations of columns used to uniquely identify rows and establish relationships between relations.

Table 1: Classification of Relational Keys

Key Type	Definition	Core Characteristics & Example
Candidate Key	An attribute or minimal set of attributes that uniquely identifies a row.	A table can have multiple candidate keys (e.g., `Admission_No` and `Aadhar_No` in a `Student` table).
Primary Key	The specific Candidate Key selected by the database designer to uniquely identify tuples.	**Must be unique and CANNOT contain NULL values**. A table has exactly one Primary Key.
Alternate Key	Any Candidate Key that was **not** selected as the Primary Key.	If `Admission_No` is chosen as Primary Key, then `Aadhar_No` becomes the Alternate Key.
Foreign Key	An attribute in a child relation that references the Primary Key of a parent relation.	Establishes referential integrity between tables (e.g., `DeptID` in `Employee` referring to `DeptID` in `Department`).

⚠ Foreign Key Data Type Rule

The primary key column in the parent table and the corresponding foreign key column in the child table **must share the exact same data type and size**.

2 Introduction to MySQL & Database Setup

2.1 Overview of MySQL

MySQL (pronounced *"My Sequel"*) is the world's most popular open-source Relational Database Management System. It was originally created in Sweden by David Axmark, Allan Larsson, and Michael Widenius.

2.1.1 Key Characteristics of MySQL

- **Open-Source & Free:** Distributed under the GNU General Public License; no licensing fees required.
- **High Performance & Speed:** Exceptional execution speed, reliability, and ease of optimization for large datasets.
- **ANSI-SQL Compliant:** Standardized SQL syntax compatibility.
- **Cross-Platform Flexibility:** Runs seamlessly on Windows, Linux, macOS, UNIX, and Solaris.
- **Programming Integration:** Native drivers for Java (JDBC), C++, Python, PHP, Perl, and C#.

2.2 Installing and Configuring MySQL on Windows

1. Download installer from official website: `https://www.mysql.com/downloads/`.
2. Select **MySQL Community Server** executable installer for Windows.
3. Run installer setup, accepting default destination directories.
4. During configuration, specify and record the **Root Password** (mandatory for authenticating every MySQL console session).

2.3 SQL Command Classification

SQL commands are broadly categorized into two major divisions:

Table 2: Classification of SQL Commands

Category	Full Form	Purpose & Included Commands
DDL	Data Definition Language	Defines, alters, and deletes database structures, tables, indexes, and constraints. Commands: `CREATE DATABASE`, `CREATE TABLE`, `ALTER TABLE`, `DROP TABLE`, `SHOW TABLES`, `DESCRIBE`.
DML	Data Manipulation Language	Queries, inserts, updates, and deletes data records stored within tables. Commands: `SELECT`, `INSERT INTO`, `UPDATE`, `DELETE`.

2.4 MySQL Data Types

Every column in a MySQL table must be assigned a data type specifying the nature of stored values.

Table 3: Common MySQL Data Types

Class	Data Type	Description	Literal Example
Text	`CHAR(size)`	Fixed-length string (1 to 255 chars). Right-padded with spaces to specified length.	`'Male'`, `'DELHI'`
Text	`VARCHAR(size)`	Variable-length string (1 to 255 chars). Allocates only required character bytes.	`'Abhishek'`, `'Computer Science'`
Numeric	`INT` / `INTEGER`	Whole integer numbers without decimals.	45, 1008
Numeric	`DECIMAL(size, d)`	Fixed-point decimal number. `size` = total digits, `d` = decimal places.	45000.50 (`DECIMAL(7,2)`)
Date	`DATE`	Date value in format `'YYYY-MM-DD'`.	`'2008-10-15'`
Time	`TIME`	Time value in format `'HH:MM:SS'`.	`'14:30:00'`

3 Data Definition Language (DDL) Commands

3.1 Database Level Commands

```sql
-- 1. Create a new database named 'School'
CREATE DATABASE School;

-- 2. Open / Activate database for execution
USE School;

-- 3. Display current active database name
SELECT DATABASE();
```

3.2 Table Creation & Constraints

The `CREATE TABLE` statement creates a new relation grid.

3.2.1 Common Table Constraints

- **`PRIMARY KEY`:** Uniquely identifies rows. Automatically enforces **NOT NULL** and **UNIQUE**.
- **`NOT NULL`:** Prevents null/empty entries in the column.
- **`UNIQUE`:** Guarantees all non-null values in the column are distinct.
- **`FOREIGN KEY`:** Links column to a primary key in another table, ensuring referential integrity.

3.2.2 Creating Tables with Constraints

```sql
-- Creating Student Table with Primary Key and NOT NULL constraints
CREATE TABLE Student (
    RollNo INT PRIMARY KEY,
    FName VARCHAR(25) NOT NULL,
    LName VARCHAR(25),
    Gender CHAR(1),
    DOB DATE,
    Marks DECIMAL(4,1)
);

-- Creating a Table with Composite Primary Key (Order_Num + Cust_Code)
CREATE TABLE Bills (
    Order_Num INT(4),
    Cust_Code VARCHAR(4),
    Bill_Date DATE,
    Bill_Amt DECIMAL(8,2) NOT NULL,
    PRIMARY KEY (Order_Num, Cust_Code)
);
```

3.3 Inspecting Database Objects

```
-- List all tables in current active database
SHOW TABLES;

-- Display column schema, datatypes, and constraints of a table
DESCRIBE Student;
-- OR
DESC Student;
```

3.4 Altering Table Structure (ALTER TABLE)

ALTER TABLE modifies an existing table's schema without deleting data.

```
-- 1. Add a new column 'Games' to Student table
ALTER TABLE Student ADD Games VARCHAR(20);

-- 2. Modify datatype of 'Games' column from VARCHAR to INTEGER
ALTER TABLE Student MODIFY Games INT;

-- 3. Drop (Remove) column 'Games' from Student table
ALTER TABLE Student DROP Games;

-- 4. Add Primary Key constraint to an existing table
ALTER TABLE Student ADD PRIMARY KEY (RollNo);

-- 5. Drop Primary Key constraint from a table
ALTER TABLE Student DROP PRIMARY KEY;

-- 6. Modify column to enforce NOT NULL constraint
ALTER TABLE Bills MODIFY Bill_Date DATE NOT NULL;
```

3.5 Deleting Tables (DROP TABLE)

```
-- Permanently delete table structure and all contained records
DROP TABLE Student;
```

4 Data Manipulation Language (DML) Commands

4.1 Inserting Data (INSERT INTO)

```
-- Method 1: Insert values for ALL columns in exact structural order
INSERT INTO Student VALUES (1, 'ABHISHEK', 'NARULA', 'M', '1998-10-05',
   98.0);

-- Method 2: Insert values into SPECIFIED columns only (unspecified
   columns receive NULL)
INSERT INTO Student (RollNo, Gender, LName, FName)
VALUES (8, 'F', 'SHANU', 'DEEPAKSHI');

-- Note: Strings/Dates must be enclosed in single quotes; NULL is entered
   without quotes.
```

4.2 Querying Data (SELECT)

The SELECT statement retrieves specific records from relations.

```
-- 1. Retrieve all columns for all records (* is wildcard for ALL)
SELECT * FROM Student;

-- 2. Retrieve specific columns
SELECT RollNo, FName, Marks FROM Student;

-- 3. Eliminate duplicate output values using DISTINCT
SELECT DISTINCT Marks FROM Student;

-- 4. Filter records using WHERE clause
SELECT FName, Marks FROM Student WHERE Marks > 90;

-- 5. Arithmetic operations on column values in result set
SELECT FName, Marks + 10 AS 'Bonus Marks' FROM Student WHERE Marks > 90;

-- 6. Relational Operators (=, >, <, >=, <=, <>)
SELECT FName, Marks FROM Student WHERE Gender = 'F';
SELECT FName, Marks FROM Student WHERE Marks <> 95;

-- 7. Logical Operators (AND, OR, NOT)
SELECT FName, Marks FROM Student WHERE Gender = 'F' AND Marks > 90;
SELECT FName, Marks FROM Student WHERE Marks < 50 OR Marks > 90;
SELECT FName, Marks FROM Student WHERE NOT (Marks < 50);

-- 8. Range selection using BETWEEN ... AND ... (Inclusive)
SELECT FName, LName, Marks FROM Student WHERE Marks BETWEEN 92 AND 95;

-- 9. Set membership matching using IN operator
SELECT FName, Marks FROM Student WHERE Marks IN (88, 92, 95);
```

```sql
-- 10. Pattern matching using LIKE operator
-- '%' matches 0 or more characters; '_' matches exactly 1 character
SELECT * FROM Student WHERE FName LIKE 'A%';      -- Starts with 'A'
SELECT * FROM Student WHERE FName LIKE '%K';      -- Ends with 'K'
SELECT * FROM Student WHERE FName LIKE '_____';   -- Exactly 5 characters
    long
SELECT * FROM Student WHERE FName LIKE '_H%';     -- 'H' as second
    character

-- 11. Handling NULL values (Use IS NULL / IS NOT NULL; NEVER = NULL)
SELECT FName, Marks FROM Student WHERE Marks IS NULL;
SELECT FName, Marks FROM Student WHERE Marks IS NOT NULL;

-- 12. Column Aliases using AS keyword
SELECT RollNo AS 'Roll Number', FName AS 'First Name', Marks FROM Student;

-- 13. Sorting output using ORDER BY clause (ASC is default, DESC for
    descending)
SELECT FName, LName, Marks FROM Student ORDER BY Marks DESC;
SELECT FName, LName, Marks FROM Student ORDER BY Marks ASC, FName DESC;
```

4.3 Modifying Records (UPDATE)

```sql
-- Modify specific column values for matching condition records
UPDATE Student SET Marks = Marks + 5 WHERE Gender = 'F';

-- Update multiple columns
UPDATE Student SET DOB = '1999-12-01', Marks = 95.0 WHERE RollNo = 8;
```

⚠ UPDATE Without WHERE Danger

Executing an UPDATE statement without a WHERE clause will modify **every single row** in the table!

4.4 Deleting Records (DELETE)

```sql
-- Delete specific records matching condition
DELETE FROM Student WHERE RollNo = 8;

-- Delete ALL records from table (Table structure remains intact)
DELETE FROM Student;
```

5 Comprehensive Question Bank

5.1 Multiple Choice Questions (MCQs)

1. Which of the following refers to raw, unorganized facts and figures?

   (A) Information
   (B) Data
   (C) Knowledge
   (D) Query

   Answer: (B) Data
   Explanation: Data consists of raw, uninterpreted facts. When data is processed and given context, it becomes Information.

2. What type of software is MySQL?

   (A) Operating System
   (B) Word Processor
   (C) Open-Source RDBMS
   (D) Programming Language

   Answer: (C) Open-Source RDBMS
   Explanation: MySQL is a free, open-source Relational Database Management System.

3. In a relational model, what is a horizontal row of a table called?

   (A) Attribute
   (B) Tuple / Record
   (C) Domain
   (D) Degree

   Answer: (B) Tuple / Record
   Explanation: Rows in a relation are termed Tuples or Records. Columns are termed Attributes or Fields.

4. What term describes the total number of attributes (columns) in a relation?

   (A) Cardinality
   (B) Degree
   (C) Domain
   (D) Primary Key

   Answer: (B) Degree
   Explanation: Degree refers to column count; Cardinality refers to row count.

5. Which key constraint ensures that a column uniquely identifies records and **cannot contain NULL**?

   (A) Foreign Key
   (B) Alternate Key
   (C) Primary Key
   (D) Foreign Key

   Answer: (C) Primary Key
   Explanation: A Primary Key enforces uniqueness and strictly disallows NULL values.

6. Candidate keys that are **not** chosen as the Primary Key are known as:

(A) Foreign Keys
(B) Alternate Keys
(C) Composite Keys
(D) Super Keys

Answer: (B) Alternate Keys
Explanation: All candidate keys not selected as the primary key are alternate keys.

7. Which SQL command classification includes `CREATE`, `ALTER`, and `DROP`?

(A) DML
(B) DDL
(C) DCL
(D) TCL

Answer: (B) DDL
Explanation: DDL (Data Definition Language) commands manage database and table structures.

8. Which SQL command category includes `SELECT`, `INSERT`, `UPDATE`, and `DELETE`?

(A) DDL
(B) DML
(C) DCL
(D) DQL

Answer: (B) DML
Explanation: DML (Data Manipulation Language) commands handle table data rows.

9. Which datatype is preferred for variable-length text strings to save storage space?

(A) CHAR
(B) VARCHAR
(C) INT
(D) DECIMAL

Answer: (B) VARCHAR
Explanation: `VARCHAR` dynamically allocates storage according to string length, unlike fixed-length `CHAR`.

10. Which SQL keyword is used to eliminate duplicate output rows in a `SELECT` query?

(A) UNIQUE
(B) DISTINCT
(C) DIFFERENT
(D) GROUP BY

Answer: (B) DISTINCT
Explanation: `SELECT DISTINCT` removes duplicate rows from result sets.

11. Which wildcard character in a `LIKE` pattern matches **exactly one** character?

(A) %
(B) _
(C) *
(D) ?

Answer: (B) _
Explanation: The underscore (_) matches a single character, whereas % matches 0 or more characters.

12. How must `NULL` values be tested in a SQL `WHERE` clause?

(A) = NULL
(B) <> NULL
(C) IS NULL
(D) == NULL

Answer: (C) IS NULL
Explanation: `NULL` represents unknown data; equality operators (=) fail with `NULL`. `IS NULL` must be used.

13. Which clause is used to sort the output result set of a `SELECT` query?

(A) SORT BY
(B) ALIGN BY
(C) ORDER BY
(D) GROUP BY

Answer: (C) ORDER BY
Explanation: `ORDER BY` sorts results in `ASC` (ascending) or `DESC` (descending) order.

14. Which SQL statement is used to modify existing data values in a table?

(A) ALTER TABLE
(B) CHANGE
(C) UPDATE
(D) MODIFY

Answer: (C) UPDATE
Explanation: `UPDATE` modifies stored data values. `ALTER TABLE` modifies table structures.

15. What happens if a `DELETE FROM Student;` command is executed without a `WHERE` clause?

(A) Syntax error occurs
(B) Only first row is deleted
(C) Table structure is deleted
(D) All rows are deleted

Answer: (D) All rows are deleted
Explanation: Without a `WHERE` filter, `DELETE` wipes all data records while preserving empty table structure.

5.2 Assertion-Reasoning Questions

1. **Assertion (A):** Primary key columns in a MySQL table cannot accept `NULL` values.
Reason (R): A Primary key is used to uniquely identify each tuple in a relation, and an unknown (`NULL`) identity violates entity integrity.
Answer: Both (A) and (R) are true and (R) is the correct explanation of (A).
Explanation: Entity integrity strictly prohibits NULLs in primary key attributes.

2. **Assertion (A):** The SQL command `SELECT * FROM Student WHERE Marks = NULL;` will display all students with missing marks.
Reason (R): The equality operator (=) cannot be used with `NULL`; `IS NULL` must be used.
Answer: (A) is false but (R) is true.
Explanation: Testing `= NULL` returns an empty set. The correct syntax is `WHERE Marks IS NULL`.

3. **Assertion (A):** `DROP TABLE` and `DELETE FROM` perform identical operations in SQL.
   Reason (R): `DROP TABLE` removes the table definition and data, whereas `DELETE` removes records while keeping the schema.
   Answer: (A) is false but (R) is true.
   Explanation: `DROP` is a DDL command deleting schema; `DELETE` is a DML command removing row records.

4. **Assertion (A):** The `VARCHAR` datatype is more storage-efficient than `CHAR` for storing variable-length names.
   Reason (R): `CHAR` pads short strings with spaces to fixed length, whereas `VARCHAR` stores only actual characters.
   Answer: Both (A) and (R) are true and (R) is the correct explanation of (A).
   Explanation: `VARCHAR` saves storage space by avoiding fixed space padding.

5. **Assertion (A):** A relation can have multiple candidate keys but only one primary key.
   Reason (R): All candidate keys are eligible to uniquely identify records, but only one is selected as the primary key.
   Answer: Both (A) and (R) are true and (R) is the correct explanation of (A).
   Explanation: Database design allows multiple candidate keys but strictly enforces one primary key selection.

5.3 One-Word / Brief Answer Questions

1. **Question:** Name the mathematical model on which modern tabular databases are structured.
   Answer: Relational Model.

2. **Question:** What term describes the number of rows (tuples) present in a relation?
   Answer: Cardinality.

3. **Question:** Which key constraint links a child table column to a parent table primary key?
   Answer: Foreign Key.

4. **Question:** What SQL statement displays the detailed structural schema of a table?
   Answer: `DESCRIBE` (or `DESC`).

5. **Question:** Which SQL DDL statement is used to add a new column to an existing table?
   Answer: `ALTER TABLE`.

6. **Question:** Name the wildcard character in `LIKE` clause that represents zero or more characters.
   Answer: Percent sign (%).

7. **Question:** Which SQL operator checks whether a column value falls within an inclusive numeric range?
   Answer: `BETWEEN ... AND ...`.

8. **Question:** What SQL statement is used to modify existing data values inside a table?
   Answer: `UPDATE`.

9. **Question:** Which operator is used to search for a set of multiple discrete values in a `WHERE` clause?
   Answer: `IN` operator.

10. **Question:** Name the creator/co-founder of MySQL after whose daughter "My" the software was named.
    Answer: Michael Widenius.

5.4 Differentiate Between Questions

1. **Differentiate between DBMS and RDBMS.**

Feature	DBMS	RDBMS
Data Storage Format	Stores data as navigation files, trees, or single flat files.	Stores data in 2D relational tables (rows and columns).
Table Relationships	Does not natively enforce relational links between data files.	Links tables using shared Primary and Foreign keys.
Data Redundancy	Higher redundancy potential.	Minimized redundancy through normalization rules.
Examples	MS Access (basic), XML Databases.	MySQL, PostgreSQL, Oracle, MS SQL Server.

2. **Differentiate between CHAR and VARCHAR Datatypes.**

Feature	CHAR(size)	VARCHAR(size)
Storage Allocation	Fixed length. Always consumes specified `size` bytes.	Variable length. Consumes only length of text + overhead.
Space Padding	Right-pads shorter strings with blank spaces.	Does not pad spaces. Stores exact character length.
Performance	Marginally faster static memory access.	Optimal storage efficiency for varying text lengths.
Best Use Case	Fixed-length codes (e.g., `Gender 'M'/'F'`, `State 'NY'`).	Names, email addresses, street descriptions.

3. **Differentiate between Primary Key and Foreign Key.**

Feature	Primary Key	Foreign Key
Primary Purpose	Uniquely identifies each tuple within its own relation.	References Primary Key of parent relation to link tables.
NULL Values	Strictly **prohibits NULL values**.	Accepts NULL values (unless explicitly set `NOT NULL`).
Uniqueness	Must contain 100% unique values per record.	Can contain duplicate values (referencing parent key).
Table Constraint	Exactly **one Primary Key** per table.	Multiple Foreign Keys permitted per table.

4. **Differentiate between DDL Commands and DML Commands.**

Feature	DDL (Data Definition Language)	DML (Data Manipulation Language)
Target Object	Operates on database **schema / table structure**.	Operates on internal **data rows / records**.
Key Commands	`CREATE`, `ALTER`, `DROP`, `DESCRIBE`.	`SELECT`, `INSERT`, `UPDATE`, `DELETE`.
Effect on Structure	Creates, alters, or removes tables and indexes.	Modifies or fetches data records inside existing tables.

5. **Differentiate between DELETE and DROP Commands.**

Feature	DELETE Statement	DROP TABLE Statement
Command Category	DML Command.	DDL Command.
Operation Scope	Removes specific or all data **rows** from a table.	Removes **entire table schema** and all stored data.
Table Availability	Table structure remains intact in database.	Table is permanently deleted from database catalog.
WHERE Clause	Supports `WHERE` clause filtering.	Does not use `WHERE` clause.

5.5 Short Answer Questions

1. **Question:** What is RDBMS? State two popular open-source RDBMS software.
Answer: A Relational Database Management System (RDBMS) is database software that stores data in structured 2D tables linked by primary and foreign keys. Examples include MySQL and PostgreSQL.

2. **Question:** Explain the difference between Candidate Key and Alternate Key with an example.
Answer: Candidate keys are all columns capable of uniquely identifying a record. The key selected as the main identifier is the Primary Key, while remaining non-selected candidate keys are Alternate Keys. E.g., if `RollNo` and `AdmissionNo` are candidate keys and `AdmissionNo` is made Primary Key, `RollNo` is the Alternate Key.

3. **Question:** Why is an `UPDATE` statement without a `WHERE` clause considered dangerous?
Answer: Without a `WHERE` clause, the `UPDATE` operation applies changes indiscriminately to every record in the table, overwriting existing column data globally.

4. **Question:** How does the `LIKE` operator differ from the equality operator (=) in SQL?
Answer: The = operator checks for exact string equality. The `LIKE` operator performs pattern matching using wildcard characters (% for multi-character sequences, _ for single character matches).

5. **Question:** Explain how `NULL` values are handled in SQL computations and comparison queries.
Answer: `NULL` signifies missing or unknown data. Arithmetic operations on `NULL` return `NULL`. Standard comparison operators (=, <>) fail with `NULL`; queries must use `IS NULL` or `IS NOT NULL`.

5.6 Long Answer & SQL Query Questions

1. **Question:** Explain the four major relational keys (Primary, Candidate, Alternate, Foreign) in detail. Describe their roles in ensuring data integrity across related tables.
   Answer:
   1. Candidate Key: Any attribute or set of attributes possessing unique values for every row. A table may have multiple candidate keys (e.g., `EmpID`, `PassportNo`, `SSN`).
   2. Primary Key: The specific candidate key chosen by the database architect to uniquely identify each tuple. Primary keys strictly disallow `NULL` values and enforce entity integrity.
   3. Alternate Key: All candidate keys not selected as the primary key. They remain unique secondary identifiers.
   4. Foreign Key: An attribute in a child table whose values must match the primary key of a parent table (or be NULL). Foreign keys establish relationships between relations and enforce referential integrity.

2. **Question:** Write complete SQL statements to accomplish the following database administrative tasks:

   (a) Create a database named `HospitalMaster` and activate it.

   (b) Create a table named `Doctor` with columns: `DocID` (CHAR 4, Primary Key), `DocName` (VARCHAR 30, NOT NULL), `Department` (VARCHAR 20), `Salary` (DECIMAL 8,2), and `JoinDate` (DATE).

   (c) Add a new column `Experience` (INTEGER) to `Doctor` table.

   (d) Modify `Department` column length to 35 characters.

   (e) Drop the `Experience` column from `Doctor` table.

   Answer:

```sql
-- (a) Create and activate database
CREATE DATABASE HospitalMaster;
USE HospitalMaster;

-- (b) Create Doctor table
CREATE TABLE Doctor (
    DocID CHAR(4) PRIMARY KEY,
    DocName VARCHAR(30) NOT NULL,
    Department VARCHAR(20),
    Salary DECIMAL(8,2),
    JoinDate DATE
);

-- (c) Add Experience column
ALTER TABLE Doctor ADD Experience INT;

-- (d) Modify Department column
ALTER TABLE Doctor MODIFY Department VARCHAR(35);

-- (e) Drop Experience column
ALTER TABLE Doctor DROP Experience;
```

5.7 Case-Study & Competency-Based Questions

1. **Case Study 1: Fitness Equipment Store Database Management**
   Scenario: The owner of "FlexFit India" manages inventory using a table named `GYM` with the following schema:

Table Schema: GYM

Column Name	Data Type	Description
PrCode	CHAR(4)	Product Code (Primary Key)
PrName	VARCHAR(30)	Product Name
UnitPrice	DECIMAL(8,2)	Price per unit in Rupees
Manufacturer	VARCHAR(25)	Manufacturing Brand Name

Required SQL Queries:

(a) Display the names and unit prices of all products with `UnitPrice` less than `Rs. 20,000.00`.

(b) Display details of products with `UnitPrice` in the range `20000.00` to `30000.00` (inclusive).

(c) Display all products manufactured by `"Fit Express"`, sorted in descending order of `UnitPrice`.

(d) Insert a new row for product: `PrCode` = `'P106'`, `PrName` = `'Vibro Exerciser'`, `UnitPrice` = `23000.00`, `Manufacturer` = `'Avon Fitness'`.

(e) Apply a 10% discount price reduction to all products in the `GYM` table.

(f) Display details of all products whose manufacturer name starts with the letter `'A'`.

Solutions:

```sql
-- Query (a)
SELECT PrName, UnitPrice FROM GYM WHERE UnitPrice < 20000.00;

-- Query (b)
SELECT * FROM GYM WHERE UnitPrice BETWEEN 20000.00 AND 30000.00;

-- Query (c)
SELECT * FROM GYM WHERE Manufacturer = 'Fit Express' ORDER BY UnitPrice
   DESC;

-- Query (d)
INSERT INTO GYM VALUES ('P106', 'Vibro Exerciser', 23000.00, 'Avon
   Fitness');

-- Query (e)
UPDATE GYM SET UnitPrice = UnitPrice - (UnitPrice * 0.10);

-- Query (f)
SELECT * FROM GYM WHERE Manufacturer LIKE 'A%';
```

2. **Case Study 2: School Academic & Marks Tracking System**
   Scenario: A school IT administrator manages student records in a table named `Student` (`RollNo`, `FName`, `LName`, `Gender`, `DOB`, `Marks`).
   Sub-Questions & SQL Solutions:

(a) Display first name, last name, and marks of students scoring between 92 and 95 marks.

```
SELECT FName, LName, Marks FROM Student WHERE Marks BETWEEN 92 AND
    95;
```

(b) Display all female students whose first name starts with letter 'S'.

```
SELECT * FROM Student WHERE Gender = 'F' AND FName LIKE 'S%';
```

(c) Display student records whose marks are missing (NULL).

```
SELECT * FROM Student WHERE Marks IS NULL;
```

(d) Delete student records where roll number is 13.

```
DELETE FROM Student WHERE RollNo = 13;
```