

## MYSQL

### Key concepts

SN	Name of Concept	Explanation with example
1	<b>Domain</b>	A domain is a set of values that can be stored in a column of a database table. They are the allowed values for a column. Example: A "gender" column might have a domain of "Male" or "Female." in attendance register
2	<b>Table (Relation)</b>	A collection of related data in rows and columns.
3	<b>Tuple/ Record</b>	A row in a table. Example: One student's information in a student table
4	<b>Attribute / Field</b>	A column in a table. Example: "Name," "grade," or "Roll Number" in a student table.
5	<b>Degree</b>	The number of columns in a table.
6	<b>Cardinality</b>	The number of rows in a table.
7	<b>Candidate Key</b>	A column (or set of columns) that can uniquely identify a row. Example: Student ID or Roll Number
8	<b>Primary Key</b>	A chosen candidate key or a column which will uniquely identify a row in a table. A table can only have one primary key. Example: Student Roll Number
9	<b>Alternate Key</b>	A candidate key that is not chosen as the primary key. Example If Student Roll Number is the primary key, then maybe Email Address could be an alternate key.
10	<b>Data Definition Language (DDL)</b>	DDL commands are used to define the database structure. They handle creating, altering, and deleting tables and databases.
11	<b>Data Manipulation Language (DML)</b>	DML commands are used to manage the data within the database. This includes inserting, selecting, updating, and deleting data. Examples: `INSERT`, `SELECT`, `UPDATE`, `DELETE`.
12	<b>Introduction to MySQL</b>	MySQL is a popular, free, and open-source database management system. It's used by many websites and applications. It is reliable, fast, and easy to use.
13	<b>Creating a Database</b>	The `CREATE DATABASE` command makes a new database. Syntax: CREATE DATABASE <database_name>; Example: CREATE DATABASE SchoolDB;
14	<b>Data types</b>	Data Types specify the kind of data a column can hold (e.g., text, numbers, dates). Examples: 12. `CHAR(size)`: Fixed-length text string. 13. `VARCHAR(size)`: Variable-length text string. 14. `INT(size)`: Integer number. 15. `DATE`: Date value (YYYY-MM-DD).
15	<b>Creating a Table</b>	The `CREATE TABLE` command makes a new table in a database. Syntax: CREATE TABLE <table_name> ( column1 datatype,

		column2 datatype, ... ); Example: CREATE TABLE Students ( StudentID INT, Name VARCHAR(255), Address VARCHAR(255));
16	<b>Drop : Deleting a Table</b>	The `DROP TABLE` command removes a table from the database. Syntax: DROP TABLE <table_name>; Example: DROP TABLE Students;
17	<b>Alter Command</b>	The `ALTER` command modifies the structure of a table. ALTER TABLE command allows us to perform the following operations: <ul style="list-style-type: none"> <li>▪ Adding new column in existing table            Syntax: ALTER TABLE &lt;table-name&gt; ADD COLUMN &lt;column-name&gt; datatype(length);            Example: ALTER TABLE employee ADD COLUMN grade char(2);</li> <li>▪ Dropping existing column from table            Syntax: ALTER TABLE &lt;table-name&gt; DROP COLUMN &lt;column-name&gt;;            Example: ALTER TABLE employee DROP COLUMN grade;</li> <li>▪ Modifying column definition in table            Syntax: ALTER TABLE &lt;table-name&gt; MODIFY column-name datatype(length);            Example: ALTER TABLE employee MODIFY            ename varchar(25);</li> <li>▪ Changing the name of column            Syntax: ALTER TABLE &lt;table-name&gt; CHANGE [&lt;old column-name&gt;] [&lt;new column-name&gt;] &lt;datatype&gt;;            Example: ALTER TABLE students CHANGE grade score char(2);</li> <li>▪ Dropping constraint after table creation.            Syntax: ALTER TABLE [&lt;table_name&gt;] DROP CONSTRAINT [&lt;constraint_name&gt;];            Example: ALTER TABLE emp drop PRIMARY KEY ALTER TABLE emp ALTER TABLE &lt;table_name&gt;</li> <li>I. Adding constraint after table creation.            Syntax: ALTER TABLE [&lt;table_name&gt;] ADD CONSTRAINT [&lt;constraint_name&gt;];            Example: ALTER TABLE emp ADD UNIQUE emp_name;</li> </ul>
18	<b>Data Query</b>	Querying involves retrieving specific data from the database

19	<b>Select Query</b>	<p>SELECT Command: With the SELECT command we can retrieve previously inserted rows. The general form of SELECT is:</p> <p>Syntax: SELECT column names FROM table_name; Example: SELECT * FROM Student;</p> <p>SELECT name, stream FROM Student;</p>
20	<b>Where Clause</b>	<p>The WHERE clause is used to retrieve data that satisfy the specified conditions. With WHERE clause the following operators can be used:</p> <ul style="list-style-type: none"> <li>• Comparison/ Relational Operators are: &lt;, &lt;=, =, !=, &lt;&gt;, &gt;=, &gt;</li> <li>• Logical Operators are: AND &lt; &lt;= OR NOT =</li> <li>• Comparison operator for special value – NULL, IS NULL</li> </ul>
21	<b>Relational Operators</b>	<p>Example: SELECT * FROM Student WHERE Age &lt; 18 ;</p> <p>SELECT name FROM Student WHERE stream = "S"</p> <p>SELECT name, age, stream FROM Student WHERE Stream='C';</p>
22	<b>BETWEEN Operator</b>	<p>The 'BETWEEN' operator selects values within a given range.</p> <p>Example: `SELECT * FROM Products WHERE Price BETWEEN 10 AND 20;`</p>
23	<b>Logical Operators</b>	<p>Logical operators ('AND', 'OR', 'NOT') combine conditions.</p> <p>OR: True if either condition is true</p> <p>Example: Age &gt; 18 or City = 'Mumbai'</p> <p>Means either age &gt;18 or city is Mumbai</p> <p>NOT: Reverses the condition Example: NOT City = 'Delhi'</p> <p>Means exclude Delhi</p> <p>AND: True if both conditions are true</p> <p>Example: SELECT * FROM Students WHERE Age &gt;18 AND City = 'Delhi';</p>
24	<b>NULL Value</b>	<p>NULL` represents a missing or unknown value.</p>
25	<b>NOT NULL Constraint</b>	<p>The 'NOT NULL' constraint ensures that a column cannot have a null value.</p>
26	<b>IS NULL and IS NOT NULL</b>	<p>'IS NULL' and 'IS NOT NULL' are used to check for null values.</p> <p>Example: SELECT * FROM Students WHERE Phone IS NULL;</p>

27	<b>Inserting Data</b>	The `INSERT INTO` command adds new rows to a table. Syntax:        INSERT        INTO table_name VALUES (value1, value2, ...); Example: INSERT INTO Students VALUES (1, 'John Doe', 'Delhi');
28	<b>Deleting Data</b>	The `DELETE` command removes rows from a table. Syntax: DELETE FROM table_name WHERE condition; Example: DELETE FROM Students WHERE StudentID = 1;
29	<b>Updating Data</b>	The `UPDATE` command modifies existing rows in a table. Syntax: UPDATE table_name SET column1 = value1, ... WHERE condition; Example: UPDATE Students SET City = 'Mumbai' WHERE StudentID = 1;

### Multiple Choice Questions (MCQ)

QN	Question	Answer
1	What is the primary purpose of a database? A) To store data in an unorganized manner B) To manage and organize data efficiently for retrieval and manipulation C) To create graphical user interfaces D) To perform hardware troubleshooting	B) To manage and organize data efficiently for retrieval and manipulation <b>Explanation:</b> A database is used to store, manage, and retrieve data in an organized manner to support efficient operations.
2	Which of the following is a function of a DBMS? A) Creating web applications B) Ensuring data integrity and security C) Managing computer hardware D) Writing operating system code	B) Ensuring data integrity and security <b>Explanation:</b> A DBMS ensures data integrity, security, and efficient data management.
3	In the relational model, a tuple is: A) A column in a table    B) A row in a table C) A table in a database   D) A database schema	B) A row in a table <b>Explanation:</b> A tuple represents a single row in a relation (table) in the relational model.
4	Which key uniquely identifies a tuple in a relation and is chosen by the database designer? A) Candidate Key   B) Alternate Key C) Primary Key   D) Foreign Key	C) Primary Key <b>Explanation:</b> The primary key is chosen by the designer to uniquely identify tuples in a table.
5	_____ is an example of a Data Manipulation Language (DML) command. A) CREATE   B) DROP   C) INSERT   D) ALTER	C) INSERT <b>Explanation :</b> INSERT is a DML command which is used to add new record in a table.
6	_____ is used to create a new database in MySQL. A) NEW DATABASE   B) CREATE DATABASE C) MAKE DATABASE   D) ADD DATABASE	B) CREATE DATABASE <b>Explanation:</b> The `CREATE DATABASE` command is used to create a new database in MySQL.

7	The data type which is most suitable for storing the "Marks" column in the Students table (assuming whole numbers)? A) VARCHAR B) INT C) FLOAT D) DATE	B) INT <b>Explanation:</b> INT is used for whole numbers, suitable for storing marks.
8	Identify the query to add a new column "Phone" to the Students table? A) ALTER TABLE Students ADD Phone VARCHAR(10); B) MODIFY TABLE Students Phone VARCHAR(10); C) ADD COLUMN Students Phone VARCHAR(10);	A) ALTER TABLE Students ADD Phone VARCHAR(10); -: The `ALTER TABLE ... ADD` command is used to add a new column.

	D) UPDATE TABLE Students Phone VARCHAR(10);	
9	Identify the query to retrieve students with marks between 60 and 90 (inclusive)? A) SELECT * FROM Students WHERE Marks BETWEEN 60 AND 90; B) SELECT * FROM Students WHERE Marks > 60 AND Marks < 90; C) SELECT * FROM Students WHERE Marks IN (60, 90); D) SELECT * FROM Students WHERE Marks = 60 OR Marks = 90;	A) SELECT * FROM Students WHERE Marks BETWEEN 60 AND 90; <b>Explanation:</b> The `BETWEEN` operator includes the range 60 to 90 (inclusive).
10	Among the following find out the correct query to retrieve students from "Delhi" with marks greater than 75? A) SELECT * FROM Students WHERE City = 'Delhi' OR Marks > 75; B) SELECT * FROM Students WHERE City = 'Delhi' AND Marks > 75; C) SELECT * FROM Students WHERE City = 'Delhi' XOR Marks > 75; D) SELECT * FROM Students WHERE City != 'Delhi' AND Marks > 75;	B) SELECT * FROM Students WHERE City = 'Delhi' AND Marks > 75; <b>Explanation:</b> The `AND` operator ensures that both the conditions (City = 'Delhi' and Marks > 75) must be true.

### Assertion and Reasoning-Based Questions

Mark the correct choice as

- (a) Both A and R are true and R is the correct explanation for A
- (b) Both A and R are true and R is not the correct explanation for A
- (c) A is True but R is False
- (d) A is false but R is True
- (e) Both A and R are False

QN	Question	Answer
----	----------	--------

1	<b>Assertion (A):</b> A primary key can never have NULL values. <b>Reason (R):</b> A primary key uniquely identifies each record in a table.	(a) Both A and R are true and R is the correct explanation for A
2	<b>Assertion (A):</b> DROP TABLE command permanently removes the table and its data. <b>Reason (R):</b> DROP is used to delete selected rows from a table.	(c) A is True but R is False
3	<b>Assertion (A):</b> A tuple refers to a row in a relation. <b>Reason (R):</b> A domain refers to the number of rows in a table.	(c) A is True but R is False
4	<b>Assertion (A):</b> SQL is used to create, modify, and retrieve data from databases.	(c) A is True but R is False

	<b>Reason (R):</b> SQL is a programming language used to create operating systems.	
5	<b>Assertion (A):</b> The BETWEEN operator includes both boundary values. <b>Reason (R):</b> BETWEEN 10 AND 20 includes 10 and 20 in the result.	(a) Both A and R are true and R is the correct explanation for A
6	<b>Assertion (A):</b> A candidate key may have duplicate values. <b>Reason (R):</b> Candidate keys are used to establish relationships between tables.	(e) Both A and R are False
7	<b>Assertion (A):</b> IS NULL is used to find rows with undefined (missing) values in a column. <b>Reason (R):</b> NULL means a value of 0.	(c) A is True but R is False
8	<b>Assertion (A):</b> ALTER TABLE command can be used to add a new column to an existing table. <b>Reason (R):</b> ALTER is only used to remove columns from a table.	(c) A is True but R is False
9	<b>Assertion (A):</b> MySQL is an example of a relational database management system. <b>Reason (R):</b> MySQL allows storage of unstructured media files such as images	(b) Both A and R are true and R is not the correct explanation for A
10	<b>Assertion (A):</b> SELECT is used to retrieve data from one or more tables. <b>Reason (R):</b> SELECT statement modifies existing data in a table.	(c) A is True but R is False
11	<b>Assertion (A):</b> A relation in a relational database is also known as a table. <b>Reason (R):</b> Relations contain rows and columns to store data.	(a) Both A and R are true and R is the correct explanation for A
12	<b>Assertion (A):</b> INSERT command adds new records into a table. <b>Reason (R):</b> DELETE command is also used to insert records into a table.	(c) A is True but R is False

### Very Short Questions

QN	Question	Answer
1	Differentiate between a database and a DBMS.	A database is a collection of organized data, while a DBMS is a software to manage that data. Example: Student records are a database; Oracle is a DBMS to handle them.
2	State two advantages of using a DBMS.	1. Data Consistency: Reduces redundancy by storing data centrally. 2. Security: Provides access control to protect data.
3	How does a DBMS reduce data redundancy?	Answer: A DBMS stores data in a centralized manner, eliminating duplicate copies using normalized tables, ensuring efficient storage.

		Example: Storing customer details once in a single table.
4	Differentiate between a candidate key and a primary key.	A candidate key is a set of attributes that uniquely identifies tuples, while a primary key is one chosen candidate key. Example: RollNo and Email can be candidate keys; RollNo is selected as the primary key.
5	Differentiate between DDL and DML.	DDL (Data Definition Language): Defines database structure, e.g., CREATE, DROP. DML (Data Manipulation Language): Manipulates data, e.g., INSERT, UPDATE.
6	How can you verify a database creation in MySQL?	Use the command <code>`SHOW DATABASES;`</code> to list all databases, checking if the created database (e.g., Library) appears.
7	Differentiate between CHAR and VARCHAR data types.	CHAR: Fixed-length string, e.g., CHAR(10) always uses 10 spaces. VARCHAR: Variable-length string, e.g., VARCHAR(10) uses only required spaces.
8	Write the command to create a table "Employee" where EmpID column as a primary key and Name column should not be left blank.	<code>`CREATE TABLE Employee (EmpID INT primary key, Name VARCHAR(50) not null);`</code>
9	Manoj wants to delete a record from "Employee" whose EmpID is 5. Help him to write a query in MySQL.	<code>`DELETE FROM Employee WHERE EmpID = 5;`</code>

10	Smita wants to modify the value in the marks column to 95 whose roll no is 101 in the student table.	`UPDATE Student SET Marks = 95 WHERE RollNo = 101;`
11	Write the query to add a column "Salary" to the "Employee" table.	`ALTER TABLE Employee ADD Salary DECIMAL(10,2);`
12	Categorize the following commands as DDL or DML: INSERT, UPDATE, ALTER, DROP	INSERT, UPDATE: DML ALTER, DROP: DDL

### Case- Based Questions

QN	Question	Answer
1	Ritika, a computer science teacher, is guiding her students in maintaining a record of exam results using MySQL. She	i) SELECT Name FROM Student WHERE Marks > 85;
	creates a table named Student to store the following details: Help her to write query for: i) Ritika wants to find out which students have scored <b>more than 85 marks</b> . ii) One student's marks was entered incorrectly. Ritika wants to change David's marks to 72.65. Write the SQL command to do this. iii) A new student named Frank with RollNo 106, Marks 80.5 and Grade B joins the class. Help Ritika to insert the record into the table.	ii) UPDATE Student SET Marks = 72.65 WHERE Name = 'David'; iii) INSERT INTO Student VALUES (106, 'Frank', 80.5, 'B');

2

RELATION : LIBRARY

Rohit, a school librarian, wants to digitize the library’s records to manage the availability and issue of books efficiently. He designs a table named Library in MySQL with the following fields:

i) Write an SQL query to display books with price greater than 300 and less than or equal to 450.

ii) Rohit mistakenly entered a wrong category for Book\_ID 104. Write the command to change its category to 'Literature'.

iii) Rohit wants to delete the record of the book titled 'The Alchemist'. Write the SQL command.

Answers:

i) SELECT \* FROM Library WHERE Price > 300 AND Price <= 450;

ii) UPDATE Library SET Category = 'Literature' WHERE Book\_ID = 104;

iii) DELETE FROM Library WHERE Title = 'The Alchemist';

Book_ID	Title	Author	Category	Price	Issued
101	Python Programming	Nitin Sinha	Computer	350.00	Yes
102	Wings of Fire	A.P.J. Abdul Kalam	Biography	275.50	No
103	Learning MySQL	Vikram Dutt	Computer	425.00	Yes
104	The Alchemist	Paulo Coelho	Fiction	200.00	No
105	Introduction to AI	Anjali Mehta	Computer	500.00	Yes

Dr. Mehta, the administrator of Sunrise Hospital, is developing a database to manage patient records. She designs a **table Patient** to store patient details. Help her to write the query. The structure and data of the table is as follows:

RELATION : PATIENT

3	P102	Rahul Verma	45	M	Diabetes	Dr. Sinha	2024-02-15	No
	P103	Meena Iyer	28	F	Migraine	Dr. Kapoor	2024-01-20	Yes
	P104	Aman Joshi	55	M	Heart Disease	Dr. Bhalla	2024-03-02	No
	P105	Riya Malhotra	40	F	Arthritis	Dr. Sinha	2024-02-25	Yes
	Patient_ID	Name	Age	Gender	Disease	Doctor	Admission_Date	Discharged
	P101	Ananya Sharma	32	F	Pneumonia	Dr. Kapoor	2024-01-12	Yes

<p>i) Write the SQL query to display the name and disease of all patients who are still admitted.</p> <p>ii) Write an SQL command to insert a new record: Patient_ID: P106, Name: Veer Singh, Age: 38, Gender: M, Disease: Asthma, Doctor: Dr. Kapoor, Admission_Date: '2024-03-10', Discharged: No</p> <p>iii) Find the names of all patients who were admitted in February 2024.</p> <p><b>Answers :</b></p> <p>i) SELECT Name, Disease FROM Patient WHERE Discharged = 'No';</p> <p>ii) INSERT INTO Patient VALUES ('P106', 'Veer Singh', 38, 'M', 'Asthma', 'Dr. Kapoor', '2024-03-10', 'No');</p> <p>iii) SELECT Name FROM Patient WHERE Admission_Date BETWEEN '2024-02-01' AND '2024-02-29';</p>
---

### Long Answer Questions

Refer table Student

QN	Question	Answer
1	<p>i) Identify the <b>Primary key</b> and any <b>Alternate key</b> in the <b>Student</b> table.</p> <p>ii) Write the command to create the database SchoolDB and use it.</p> <p>iii) Write the SQL command to create the Student table as per the structure above.</p>	<p><b>i) Primary Key:</b> RollNo (unique for every student)</p> <p><b>Alternate Key:</b> Name (assuming no two students have the same name)</p> <p><b>ii) CREATE DATABASE</b> SchoolDB; <b>USE</b> SchoolDB;</p> <p><b>iii) CREATE TABLE</b> Student ( RollNo INT PRIMARY KEY, Name VARCHAR(50), Marks decimal(5,2), Grade CHAR(1));</p>
2	<p>i) Insert the following record into the Student table: RollNo: 106, Name: Farah, Marks: 79.45, Grade: B</p> <p>ii) Write a query to display names of students who scored marks between 80 and 90.</p> <p>iii) Write the SQL command to delete the record of the student David.</p>	<p>i) INSERT INTO Student VALUES (106, 'Farah', 79.45, 'B');</p> <p>ii) SELECT Name FROM Student WHERE Marks BETWEEN 80 AND 90;</p> <p>iii) DELETE FROM Student WHERE Name = 'David';</p>
3	<p>i) Increase the marks of all students with grade 'B' by 5 marks.</p> <p><b>ii) What is the degree and cardinality of the Student table?</b></p>	<p>i) UPDATE Student SET Marks = Marks + 5 WHERE Grade = 'B';</p> <p>ii) Degree (Number of columns): 4 Cardinality (Number of rows): 5</p>
4	<p>i) Write the SQL query to display RollNo and Name of students whose Grade is NOT NULL.</p> <p>ii) Change the data type of column Marks to DECIMAL(5,2) using SQL.</p> <p>iii) Write the command to remove the entire table Student from the database.</p>	<p>i) SELECT RollNo, Name FROM Student WHERE Grade IS NOT NULL;</p> <p>ii) ALTER TABLE Student MODIFY Marks DECIMAL(5,2);</p> <p>iii) DROP TABLE Student;</p>

## Math Functions():

### Mod (), Power () & Round()

Three commonly used numeric functions are POWER(), ROUND() and MOD().

Function	Description	Example
POWER(P,Q) or POW(P,Q)	Calculates P to the power Q.	SELECT POWER(2,3); Output: 8 (because $2 \times 2 \times 2 = 8$ )
ROUND(N,D)	Rounds off number N to D number of decimal places. Note: If D=0, then it rounds off the number to the nearest integer.	SELECT ROUND(3.14159, 2); Output:3.14  SELECT ROUND(165.2); Output:165
MOD(X, Y)	Returns the remainder after dividing number X by number Y.	mysql> SELECT MOD(10, 3); Output:1 (because $10 \div 3 = 3$ remainder 1)

## String/Text functions:

UCASE ()/UPPER (), LCASE ()/LOWER (), MID ()/SUBSTRING ()/SUBSTR (), LENGTH (), LEFT (), RIGHT (), INSTR(), LTRIM (), RTRIM(), TRIM ().

String functions in SQL are powerful tools for manipulating text data. They allow us to modify, extract, and format strings in various ways.

Function Name	Description	Example	Output
UCASE(string) OR UPPER(string)	Converts a string to uppercase	SELECT UPPER('MySQL');	MYSQL
LOWER(string) OR LCASE(string)	Converts a string to lowercase	SELECT LOWER('MYSQL');	mysql
SUBSTRING(string, pos, n) OR MID(string, pos, n) OR SUBSTR(string, pos, n)	Extracts a substring from a string. <b>Index starts at 1</b>	SELECT SUBSTRING('DataBaseS ystems', 5, 5);	BaseS (Starts at 5th character, takes 5 characters)
LENGTH(string) OR LEN(string)	Returns the length of a string	SELECT LENGTH(' SQL Rocks! ');	12 (Counts all characters including spaces)

LEFT(string, N)	Returns the left part of a string with the specified number of characters	SELECT LEFT('Environment', 6);	Enviro
RIGHT(string, N)	Returns the right part of a string with the specified number of characters	SELECT RIGHT('Environment', 4);	ment
INSTR(string, substring)	Returns the position of the first occurrence of a substring	SELECT INSTR('I love learning SQL and Python', 'SQL');	17 (Starts at position 17)
LTRIM(string)	Removes leading spaces from a string	SELECT LTRIM(' Hello World');	Hello World
RTRIM(string)	Removes trailing spaces from a string	SELECT RTRIM('Hello World ');	Hello World
TRIM(string)	Removes leading and trailing spaces from a string	SELECT TRIM(' Hello World ');	Hello World

### Multiple Choice Questions

QN	Question	Answer
1	<b>What does TRIM(' AI Tools ') return?</b> A) ' AI Tools '      B) 'AITools' C) 'AI Tools'      D) ' AI Tools '	C) 'AI Tools' Explanation:TRIM() removes spaces from both ends.
2	<b>What is the output of LTRIM(' Coding')?</b> A) ' Coding'      B) ' Coding ' C) 'Coding'      D) 'Coding '	C) 'Coding' Explanation:LTRIM() removes spaces from left side only
3	<b>What is the position of "SQL" in INSTR('Learn SQL Programming', 'SQL')?</b>	C)7 Explanation:"SQL" starts at 7th position in the string.

	A) 7    B) 8    C) 7    D) 6	
--	------------------------------	--

4	<b>What does</b> RIGHT('EXAMINATION', 6) <b>return?</b>  A) ATION    B) TION    C) NATION    D) INATION	C)NATION Explanation:Last 6 letters "NATION"
5	<b>What is the result of</b> SUBSTRING('ComputerScience', 9, 7)?  A) puterSc            B) mputer C) Science            D) terScie	C) Science Explanation:Starts at position 9 → "S", takes 7 letters → "Science"
6	<b>What does</b> LENGTH('Python 3') <b>return?</b>  A) 6    B) 8    C) 7    D) 9	B) 8 Explanation: Characters including space: "P y t h o n _ 3" → 8 characters
7	<b>What is the output of</b> UCASE('cbse board')?  A) cbse board    B) CBSE BOARD C) Cbse board    D) CBSe BOARD	B) CBSE BOARD Explanation: UCASE() converts text to uppercase → "CBSE BOARD"

## Assertion and Reasoning-Based Questions

### Mark the correct choice as

- (a) Both A and R are true and R is the correct explanation for A
- (b) Both A and R are true and R is not the correct explanation for A
- (c) A is True but R is False
- (d) A is false but R is True

QN	Question	Answer
1	<b>Assertion (A):</b> TRIM(' Hello ') returns 'Hello'. <b>Reason (R):</b> The TRIM() function removes all internal spaces from a string.	(c) A is true but R is false Explanation: TRIM() removes only leading and trailing spaces, not the spaces between words.
2	<b>Assertion (A):</b> INSTR('Banana', 'n') returns 3. <b>Reason (R):</b> INSTR() returns the position of the first occurrence of a substring.	Answer: (a) Both A and R are true and R is the correct explanation for A Explanation: 'n' first appears at the 3rd position in 'Banana'.
3	<b>Assertion (A):</b> LEFT('Notebook', 4) returns 'Note'. <b>Reason (R):</b> The LEFT() function returns the last n characters of a string.	Answer: (c) A is true but R is false Explanation: LEFT() returns the first n characters, not the last
4	<b>Assertion (A):</b> The LENGTH('apple') function returns 6. <b>Reason (R):</b> The LENGTH() function returns the number of characters in a string.	Answer: (d) A is false but R is true Explanation: LENGTH('apple') returns 5, not 6.

5	<b>Assertion (A):</b> MID('Computer', 2, 3) returns 'omp'. <b>Reason (R):</b> The MID() or SUBSTRING() function extracts a portion of a string from a given position for a specified length.	Answer: (a) Both A and R are true and R is the correct explanation for A Explanation: Starting from position 2, extract 3 characters → "omp".
6	<b>Assertion (A):</b> The function UCASE('hello') returns HELLO. <b>Reason (R):</b> The UCASE() or UPPER() function is used to convert text to uppercase.	Answer: (a) Both A and R are true and R is the correct explanation for A Explanation: UCASE() or UPPER() converts lowercase to uppercase letters.
7	<b>Assertion (A):</b> The MOD(17, 5) function returns 2. <b>Reason (R):</b> The MOD() function returns the product of the two numbers.	Answer: (c) A is true but R is false Explanation: MOD() returns the remainder of division, not the product. $17 \div 5 = 3$ remainder 2.
8	<b>Assertion (A):</b> The POWER(2, 3) function returns 8. <b>Reason (R):</b> The POWER() function is used to raise one number to the power of another.	Answer: (a) Both A and R are true and R is the correct explanation for A Explanation: $2^3 = 8$ ; POWER(2, 3) means 2 raised to the power of 3

### Very Short Questions

Consider a following table and answer the following questions

Students (RollNo, Name, Marks, Grade)

RollNo	Name	Marks	Grade
101	Alice	84.75	B
102	Bob	91.45	A
103	Charlie	76.20	B
104	David	68.99	C
105	Eva	88.33	B

QN	Question	Answer
1	Write a query to display the names of all students in uppercase.	SELECT UPPER(Name) FROM STUDENT;
2	Write a query to show the first 3 characters of each student's name.	SELECT LEFT(Name, 3) FROM STUDENT;
3	Write a query to display each student's name and the result of Marks MOD 10 (i.e., the	SELECT Name, MOD(Marks, 10) AS Remainder FROM STUDENT;

	remainder when marks are divided by 10).	
--	--	--

4	Write a query to display Name and the position of letter 'a' in the name using the INSTR) function.	SELECT Name, INSTR(Name, 'a') AS Position FROM STUDENT;
5	Write a query to display the Name, Marks, and rounded marks (to nearest integer) of each student.	SELECT Name, Marks, ROUND(Marks) AS RoundedMarks FROM STUDENT;

### Case- Based Questions

QN	Question																								
1	<p>The HR department of <b>TechNova Ltd.</b> maintains a table named Employee that stores information about employees' names, salaries, and departments. Here's a sample of the table:</p> <table><tr><th>Emp_ID</th><th>Name</th><th>Salary</th><th>Department</th></tr><tr><td>E101</td><td>rohit sharma</td><td>45200.75</td><td>Sales</td></tr><tr><td>E102</td><td>Ananya Das</td><td>37800.50</td><td>HR</td></tr><tr><td>E103</td><td>MEENA JOSHI</td><td>62000.00</td><td>Finance</td></tr><tr><td>E104</td><td>sushil mehra</td><td>50000.25</td><td>IT</td></tr><tr><td>E105</td><td>Divya Nair</td><td>47000.00</td><td>HR</td></tr></table> <p><b>Help her to write query for</b></p> <p>i) Write an SQL query to display all employee names in uppercase. ii) Display the last 4 characters of the employee names iii) Write a query to round the salaries to the nearest 100. iv) Display employee names without leading and trailing spaces.</p> <p><b>Answers :</b></p> <p>i) SELECT UPPER(Name) FROM Employee; ii) SELECT RIGHT(Name, 4) FROM Employee; iii) SELECT Name, ROUND(Salary, -2) FROM Employee; iv) SELECT TRIM(Name) FROM Employee;</p>	Emp_ID	Name	Salary	Department	E101	rohit sharma	45200.75	Sales	E102	Ananya Das	37800.50	HR	E103	MEENA JOSHI	62000.00	Finance	E104	sushil mehra	50000.25	IT	E105	Divya Nair	47000.00	HR
Emp_ID	Name	Salary	Department																						
E101	rohit sharma	45200.75	Sales																						
E102	Ananya Das	37800.50	HR																						
E103	MEENA JOSHI	62000.00	Finance																						
E104	sushil mehra	50000.25	IT																						
E105	Divya Nair	47000.00	HR																						
2	<p>BookBazaar maintains a table named Books to store details of the books available in their store. Here's a sample of the table:</p> <ul style="list-style-type: none"><li>Write a query to round off the prices to the nearest whole number.</li><li>Show the square of the price of each book.</li><li>Extract 3 characters starting from position 6 from each author's name.</li></ul> <p><b>RELATION : Book_ID</b></p> <table><tr><th>Book_ID</th><th>Title</th><th>Author</th><th>Price</th></tr><tr><td>B101</td><td>the alchemist</td><td>paulo coelho</td><td>299.75</td></tr><tr><td>B102</td><td>Wings of Fire</td><td>A.P.J Abdul Kalam</td><td>350.40</td></tr><tr><td>B103</td><td>the theory of everything</td><td>Stephen Hawking</td><td>499.99</td></tr><tr><td>B104</td><td>deep work</td><td>Cal Newport</td><td>280.00</td></tr><tr><td>B105</td><td>zero to one</td><td>Peter Thiel</td><td>399.50</td></tr></table>	Book_ID	Title	Author	Price	B101	the alchemist	paulo coelho	299.75	B102	Wings of Fire	A.P.J Abdul Kalam	350.40	B103	the theory of everything	Stephen Hawking	499.99	B104	deep work	Cal Newport	280.00	B105	zero to one	Peter Thiel	399.50
Book_ID	Title	Author	Price																						
B101	the alchemist	paulo coelho	299.75																						
B102	Wings of Fire	A.P.J Abdul Kalam	350.40																						
B103	the theory of everything	Stephen Hawking	499.99																						
B104	deep work	Cal Newport	280.00																						
B105	zero to one	Peter Thiel	399.50																						

**Answer :**

- i) `SELECT Title, ROUND(Price) FROM Books;`
- ii) `SELECT Title, POWER(Price, 2) FROM Books;`
- iii) `SELECT SUBSTRING(Author, 6, 3) FROM Books;`

## **DATE FUNCTIONS**

### **CONCEPT MAP**



1. **NOW( )** : Returns the current date and time as a datetime value.

**Example:**

`SELECT NOW();`

```
+-----+
| NOW( ) |
+-----+
| 2025-04-25 15:05:56 |
+-----+
```

2. **DATE( )** : Extracts the date part from a datetime expression.

**Example:**

`SELECT DATE(NOW());`

```
+-----+
| DATE(NOW( )) |
+-----+
| 2025-04-25 |
+-----+
```

3. **MONTH( )** : Returns the month number (1-12) from a date.

**Example:**

`SELECT MONTH('2025-04-25');`

```
+-----+
| MONTH( '2025-04-25' ) |
+-----+
| 4 |
+-----+
```

4. **MONTHNAME( )** : Returns the full month name of a date.

**Example:**

`SELECT MONTHNAME('2025-04-25');`

MONTHNAME('2025-04-25')
April

5. **YEAR()** : Extracts the year from a date value.

Example:

**SELECT YEAR('2025-04-25');**

YEAR('2025-04-25')
2025

6. **DAY()** : Returns the day of the month (1-31) from a date.

Example:

**SELECT DAY('2025-04-25');**

DAY('2025-04-25')
25

7. **DAYNAME()** : Returns the name of the weekday for a date.

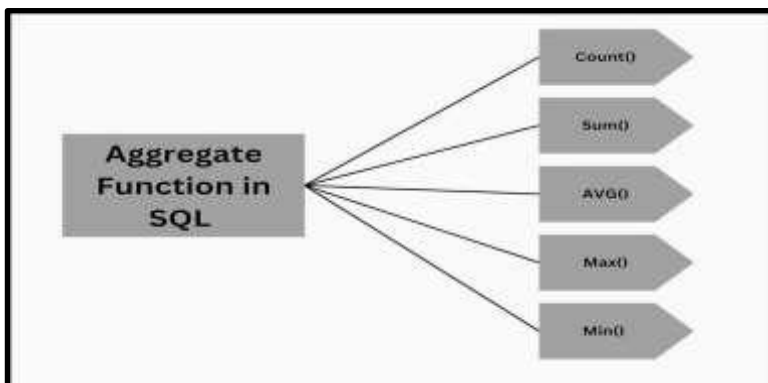
Example:

**SELECT DAYNAME('2025-04-25');**

DAYNAME('2025-04-25')
Friday

## AGGREGATE FUNCTIONS

### CONCEPT MAP



Aggregate functions are special SQL functions that perform calculations on multiple rows and return a single value.

Table below shows the difference between Single row functions and multiple row functions. Consider the below Table **Employees** in order to understand the concept of aggregate function with examples.

## Employees Table

employee_id	first_name	last_name	department	salary	hire_date	birth_date
1	John	Smith	Engineering	75000.00	2020-05-15	1985-03-25
2	Sarah	Johnson	Marketing	68000.00	2019-10-10	1990-07-12
3	Michael	Davis	Engineering	82000.00	2018-02-28	1982-11-30
4	Emily	Wilson	HR	62000.00	2021-01-05	1992-04-18
5	David	Brown	Engineering	79000.00	2017-08-01	1988-09-03
6	Lisa	Miller	Marketing	70000.00	2019-03-15	1991-05-22
7	Robert	Taylor	Finance	90000.00	2016-11-12	1980-12-15
8	Jennifer	Anderson	HR	64000.00	2022-06-20	1994-02-28
9	William	Thomas	Engineering	85000.00	2018-07-09	1986-08-17
10	Jessica	Moore	Finance	88000.00	2017-04-25	1983-10-10

### BASIC AGGREGATE FUNCTIONS

**1. COUNT ( ) :** The COUNT function counts rows or values. It's like asking "How many?"  
**Examples:**

**a. Count all rows in the employees table**

**SELECT COUNT(\*) FROM employees;**

COUNT(*)
10

**b. Count employees in the Engineering department**

**SELECT COUNT(\*) FROM employees WHERE department = 'Engineering';**

COUNT(*)
4

**c. Count how many different departments exist**

**SELECT COUNT(DISTINCT department) FROM employees;**

COUNT(DISTINCT department)
4

The DISTINCT keyword makes COUNT only count unique values, ignoring duplicates.

2. **SUM ()** : The SUM function adds up numeric values. It's like asking "What's the total?"

**Examples:**

- a. **Calculate total salary budget**

**SELECT SUM(salary) FROM employees;**

```
+-----+
| SUM(salary) |
+-----+
|  763000.00  |
+-----+
```

3. **AVG()**: The AVG function calculates the average (mean) of numeric values. It's like asking "What's the typical value?"

- a. **Find average employee salary**

**SELECT AVG(salary) FROM employees;**

```
+-----+
| AVG(salary) |
+-----+
| 76300.000000 |
+-----+
```

4. **MIN()** : This function finds the smallest (MIN) value. They're like asking "What's the lowest?"

**Examples:**

- a. **Find lowest salary**

**SELECT MIN(salary) FROM employees;**

```
+-----+
| MIN(salary) |
+-----+
|   62000.00  |
+-----+
```

5. **MAX()** : This function finds the smallest (MIN) or largest (MAX) value. They're like asking "What's the lowest/highest?"

**Examples:**

- a. **Find highest salary**

**SELECT MAX(salary) FROM employees;**

```
+-----+
| MAX(salary) |
+-----+
|   90000.00  |
+-----+
```

## **COUNT() vs COUNT(\*) in SQL**

The main difference between COUNT() and COUNT(\*) is what they count:

- COUNT(\*) counts all rows in the result set, including those with NULL values
- COUNT(column) counts only non-NULL values in the specified column

id	name	department	manager_id
1	Alice	Sales	5
2	Bob	Marketing	NULL
3	Charlie	Sales	5
4	David	IT	NULL
5	Eve	Management	NULL

**SELECT COUNT(\*) FROM employees;**

**v/s**

**SELECT COUNT(manager\_id) FROM employees;**

The results would be:

- COUNT(\*): 5 (counts all rows)
- COUNT(manager\_id): 2 (counts only non-NULL values in manager\_id column)

This distinction is important when working with data that might contain NULL values, as the two counting methods can give significantly different results.

## QUERYING AND MANIPULATING DATA

### GROUP BY and HAVING Clauses

#### GROUP BY Clause

The GROUP BY clause lets you group rows that have the same values into summary rows. It's like sorting information into piles or categories. When combined with aggregate functions, it helps you analyse data by groups.

#### Basic GROUP BY Syntax

**SELECT column1, aggregate\_function(column2) FROM  
table**

**GROUP BY column1; For  
example:**

**Count employees in each department**

**SELECT department, COUNT(\*) AS employee\_count FROM  
employees**

**GROUP BY department;**

department	employee_count
Engineering	4
Marketing	2
HR	2
Finance	2

## The SELECT List Rule in GROUP BY Clause

In standard SQL, if you use GROUP BY, each column in your SELECT list must either:

1. Be included in the GROUP BY clause, OR
2. Be used within an aggregate function

Example:

```
SELECT department, COUNT(*), AVG(salary)
FROM employees
GROUP BY department;
```

department	COUNT(*)	AVG(salary)
Engineering	4	80250.000000
Marketing	2	69000.000000
HR	2	63000.000000
Finance	2	89000.000000

## What is the HAVING Clause?

The HAVING clause filters groups, similar to how the WHERE clause filters individual rows. It's like setting requirements for which groups should appear in your results.

Example:

```
SELECT department, COUNT(*) AS employee_count FROM
employees
GROUP BY department
HAVING COUNT(*) > 2;
```

department	employee_count
Engineering	4

This shows only departments that have more than 2 employees.

## ORDER BY Clause

The ORDER BY clause sorts the result set of a query based on one or more columns. Think of it like organizing a list alphabetically or numerically to make it easier to find what you're looking for.

Order By Basic Syntax:

```
SELECT column1, column2, ... FROM table_name
```

```
ORDER BY column1 [ASC|DESC], column2 [ASC|DESC], ...;
```

Examples:

**(a) Sort employees by last name alphabetically**

**SELECT employee\_id, first\_name, last\_name  
FROM employees ORDER BY last\_name;**

employee_id	first_name	last_name
8	Jennifer	Anderson
5	David	Brown
3	Michael	Davis
2	Sarah	Johnson
6	Lisa	Miller
10	Jessica	Moore
1	John	Smith
7	Robert	Taylor
9	William	Thomas
4	Emily	Wilson

**(b) Sort employees by salary from highest to lowest**

**SELECT employee\_id, first\_name, last\_name, salary FROM  
employees**

**ORDER BY salary DESC;**

employee_id	first_name	last_name	salary
7	Robert	Taylor	90000.00
10	Jessica	Moore	88000.00
9	William	Thomas	85000.00
3	Michael	Davis	82000.00
5	David	Brown	79000.00
1	John	Smith	75000.00
6	Lisa	Miller	70000.00
2	Sarah	Johnson	68000.00
8	Jennifer	Anderson	64000.00
4	Emily	Wilson	62000.00

## Sorting by Multiple Columns - ORDER BY CLAUSE

We can sort by multiple columns, creating a hierarchy of sorting criteria:

**a. Sort by department first, then by salary (highest to lowest) within each department**

**SELECT employee\_id, first\_name, last\_name, department, salary FROM  
employees**

**ORDER BY department ASC, salary DESC;**

employee_id	first_name	last_name	department	salary
9	William	Thomas	Engineering	85000.00
3	Michael	Davis	Engineering	82000.00
5	David	Brown	Engineering	79000.00
1	John	Smith	Engineering	75000.00
7	Robert	Taylor	Finance	90000.00
10	Jessica	Moore	Finance	88000.00
8	Jennifer	Anderson	HR	64000.00
4	Emily	Wilson	HR	62000.00
6	Lisa	Miller	Marketing	70000.00
2	Sarah	Johnson	Marketing	68000.00

In this example, MySQL first groups all employees by department alphabetically. Then, within each department, it sorts employees by their salary in descending order.

## Working with Two Tables Using Equi-Join

An equi-join is a type of join operation that combines rows from two tables based on a condition where the values in specified columns are equal. This is one of the most common ways to relate data across tables in a relational database.

**Syntax:**

**SELECT \* FROM <table1>, <table2>**

**WHERE table1.column =**

**table2.column;**

Consider below two tables for **Customers** and **Orders** for Join concept.

**Customers Table**

customer_id	name	email	city
1	Alice	alice@example.com	New York
2	Bob	bob@example.com	Los Angeles
3	Charlie	charlie@example.com	Chicago
4	Diana	diana@example.com	Miami

**Orders Table**

order_id	customer_id	product	amount	date
101	1	Laptop	1200.00	2023-01-15
102	3	Smartphone	800.00	2023-01-20
103	1	Headphones	150.00	2023-02-05
104	2	Monitor	300.00	2023-02-10
105	4	Keyboard	80.00	2023-02-15

**Example 1. To see all orders with customer information**

```
SELECT o.order_id, o.product, o.amount, c.name, c.email, c.city
FROM Orders o , Customers c where o.customer_id =
c.customer_id;
```

order_id	product	amount	name	email	city
101	Laptop	1200.00	Alice	alice@example.com	New York
103	Headphones	150.00	Alice	alice@example.com	New York
104	Monitor	300.00	Bob	bob@example.com	Los Angeles
102	Smartphone	800.00	Charlie	charlie@example.com	Chicago
105	Keyboard	80.00	Diana	diana@example.com	Miami

## MULTIPLE CHOICE QUESTIONS

**Q1. Which aggregate function returns the number of rows in a result set, including rows with NULL values?**

- A) COUNT(column)                      B) COUNT(\*)  
C) SUM(column)                        D) AVG(column)

**Answer : B**

**Q2. Which statement is true about the AVG () function?**

- A) It includes NULL values in the calculation    B) It ignores NULL values in the calculation  
C) It treats NULL values as zero                      D) It returns NULL if any value in the column is NULL

**Answer : B**

**Q3. Which of the following is NOT a valid aggregate function in MySQL?**

- A) MIN ()
- B) SUM()
- C) MAX ()
- D) MEDIAN()

**Answer : D**

**Q4. If a table has no rows, what will COUNT (\*) return?**

- A) NULL
- B) Error
- C) 0
- D) -1

**Answer : C**

**Q5. What is the difference between COUNT (DISTINCT column) and COUNT(column)?**

- A) COUNT (DISTINCT column) counts only unique values, while COUNT(column) counts all non-NULL values
- B) COUNT (DISTINCT column) counts all values including NULL, while COUNT(column) counts only unique values
- C) They are exactly the same
- D) COUNT (DISTINCT column) always returns a lower count than COUNT(column)

**Answer : A**

**Q6. In a GROUP BY query, which statement is correct?**

- A) You can select only columns that appear in the GROUP BY clause
- B) You can select any column regardless of whether it appears in GROUP BY
- C) You can select columns that appear in GROUP BY or are used in aggregate functions
- D) You can only select aggregate functions

**Answer : C**

**Q7. What is the correct order of clauses in a SQL query that includes GROUP BY?**

- A) SELECT, GROUP BY, FROM, WHERE
- B) SELECT, FROM, WHERE, GROUP BY
- C) SELECT, FROM, GROUP BY, WHERE
- D) FROM, SELECT, WHERE, GROUP BY

**Answer : B**

**Q8. What is the primary purpose of the HAVING clause?**

- A) To filter individual rows before grouping
- B) To filter groups after the GROUP BY has been applied
- C) To specify the columns to be grouped
- D) To sort the final result set

**Answer : B**

**Q9. What will the following query return?**

```
SELECT department, COUNT(*) AS  
employee_count FROM employees
```

GROUP BY department HAVING COUNT(\*) > 5;

- A) All departments
- B) Departments with exactly 5 employees
- C) Departments with more than 5 employees
- D) Departments with fewer than 5 employees

**Answer : C**

**Q10. What will happen if we use ORDER BY with a column that contains NULL values?**

- A) NULL values will be sorted first
- B) NULL values will be sorted last
- C) NULL values will cause an error
- D) NULL values will be excluded from the result

**Answer : A**

**Q11. How many columns can you specify in an ORDER BY clause?**

- A) Only one
- B) Maximum of three
- C) Maximum of five
- D) As many as needed

**Answer : D**

**Q12. What is the result of this query?**

```
SELECT name, department,  
       salary FROM employees  
ORDER BY department ASC, salary DESC;
```

- A) Results are sorted by salary in descending order, then by department
- B) Results are sorted by department in ascending order, then by salary in descending order
- C) Results are sorted by department in descending order, then by salary in ascending order
- D) Results are sorted by salary in ascending order, then by department

**Answer : B**

**Q13. What is the result of MONTH('2023-07-15')?**

- A) July
- B) 7
- C) 07
- D) 6

**Answer : B**

### **Assertion and Reasoning Based Questions**

Each question consists of an assertion (A) and a reason (R). You need to determine whether each statement is true or false, and if the reason correctly explains the assertion. Options are as follows:

- A. Assertion and Reason both are correct and Reason is correct explanation of assertion
- B. Assertion and Reason both are correct and Reason is not correct explanation of assertion
- C. Assertion is True and Reason is False
- D. Assertion is False and Reason is True
- E. Assertion and Reason both are False

**Question 1. Assertion (A):** COUNT(column\_name) and COUNT() *always return the same value.*

**Reason (R):** Both COUNT(column\_name) and COUNT() count all rows in a table or result set.

**Answer : E**

**Question 2 Assertion (A):** The AVG() function returns NULL if all values in the column are NULL.

**Reason (R):** Aggregate functions in MySQL ignore NULL values in their calculations.

**Answer : A**

**Question 3 Assertion (A):** The SUM() function always returns a numeric value, even when applied to an empty result set.

**Reason (R):** MySQL aggregate functions return 0 or NULL when applied to empty sets.

**Answer : D**

**Question 4 Assertion (A):** The GROUP BY clause in MySQL must include all non-aggregated columns that appear in the SELECT clause.

**Reason (R):** MySQL requires that each column in the SELECT list that is not part of an aggregate function must be included in the GROUP BY clause to ensure deterministic results. **Answer : A**

**Question 5 Assertion (A):** Using GROUP BY on a column with NULL values will place all NULL values into a single group.

**Reason (R):** MySQL treats all NULL values as identical when comparing them for equality.

**Answer : C**

**Question 6 Assertion (A):** The COUNT(column\_name) and COUNT(\*) functions can produce different results when used with GROUP BY.

**Reason (R):** COUNT(column\_name) counts only non-NULL values in the specified column, while COUNT(\*) counts all rows regardless of NULL values.

**Answer : A**

**Question 7 Assertion (A):** The WHERE and HAVING clauses can be used interchangeably when filtering rows that contain aggregate values.

**Reason (R):** Both WHERE and HAVING filter rows, but they are processed at different stages of query execution.

**Answer : D**

**Question 8 Assertion (A):** In MySQL, the default sort order is ascending (ASC) if neither ASC nor DESC is specified.

**Reason (R):** The ASC keyword is implicitly applied when no sort direction is specified to provide a consistent default behaviour.

**Answer : A**

**Question 10 Assertion (A):** In MySQL, you can sort by a column that is not included in the SELECT clause.

**Reason (R):** The ORDER BY clause operates independently of the columns listed in the SELECT clause.

**Answer : A**

**Question 11 Assertion (A):** When using ORDER BY with NULL values, MySQL sorts NULL values first when using ASC order.

**Reason (R):** MySQL treats NULL as the lowest possible value in the column's data type hierarchy.

**Answer : A**

**Question 12 Assertion (A):** The DATE() function extracts the date part from a datetime expression, removing any time component.

**Reason (R):** DATE() truncates the time portion of a datetime value to standardize date comparisons.

**Answer : A**

**Question 13 Assertion (A):** When joining tables with an Equi Join, the joined columns must have the same name.

**Reason (R):** Equi Joins compare columns that have identical values, which requires the columns to have matching names.

**Answer : E**

**Question 14 Assertion (A):** In an Equi Join, columns with NULL values will never match, even if both columns have NULL.

**Reason (R):** In SQL, NULL = NULL evaluates to NULL (unknown), not TRUE, so rows with NULL values in join columns are excluded.

**Answer : A**

**Question 15 Assertion (A):** The NOW() and CURRENT\_TIMESTAMP() functions in MySQL return identical results.

**Reason (R):** Both functions retrieve the current date and time from the server's system clock at the moment the statement begins to execute.

**Answer : A**

**NOTE: ALONG WITH THESE QUESTION ANSWERS LEARN FROM PREVIOUS MY SQL PRACTICE PAPER**