Q1. Which clause is used to “Identifies table from which to draw table and how the table should be joined”?

1. **FROM**
2. SELECT
3. ORDER By
4. WHERE

Q2. Which clause is used to “Filters out unwanted data”?

1. FROM
2. **WHERE**
3. SELECT
4. ORDER BY

Q3. Which clause is used to “group rows together by common columns values”?

1. SELECT
2. **GROUP BY**
3. FROM
4. WHERE

Q4. Which clause is used to “filter out unwanted Groups”?

1. **HAVING**
2. FROM
3. WHERE
4. SELECT

Q5. Which clause is used to “sort the rows of the final result set by one or more columns”?

1. HAVING
2. **ORDER BY**
3. WHERE
4. FROM

Q6. Which clause is used to “Modify the existing field of the table”?

1. **ALTER**
2. FROM
3. SELECT
4. MODIFY

Q7. Which among the following is not a “query clause”?

1. **WHERE**
2. MODIFY
3. ALTER
4. FROM

Q8. “MODIFY” is used with which “Query clause”?

1. **ALTER**
2. FROM
3. WHERE
4. ORDER BY

Q9. You can add a row using SQL in a database with which of the following?

1. ADD
2. CREATE
3. **INSERT**
4. MAKE

Q10. The command to remove rows from a table 'CUSTOMER' is:

1. REMOVE FROM CUSTOMER ...
2. DROP FROM CUSTOMER ...
3. **DELETE FROM CUSTOMER WHERE ...**
4. UPDATE FROM CUSTOMER ...

Q11. The command to eliminate a table from a database is:

1. REMOVE TABLE CUSTOMER;
2. **DROP TABLE CUSTOMER;**
3. DELETE TABLE CUSTOMER;
4. UPDATE TABLE CUSTOMER;

Q12. Which of the following is valid SQL for an Index?

1. **CREATE INDEX ID;**
2. CHANGE INDEX ID;
3. ADD INDEX ID;
4. REMOVE INDEX ID;

Q13. Which of the following clause you will use to replace the existing VIEW?

1. **REPLACE**
2. REPLACES
3. REPLACED
4. None of the above.

Q14. Which statement provides the definition of the VIEW?

1. **SELECT statement**
2. UPDATE statement
3. Both A and B
4. None of the above

Q15. The table name can be specified as?

1. **db\_name.tbl\_name to create the table in a specific database.**
2. tbl\_name.db\_name to create the table in a specific database.
3. user\_name.tbl\_name to create the table in a specific database.
4. All of the above.

Q16. Prevents an error from occurring if the table exists at the time of creating new table?

1. **IF NOT EXISTS**
2. IF EXISTS
3. Both A and B
4. None of the above.

Q17. If you want to create TEMPORARY table, you can use the?

1. **TEMPORARY keyword to create the table.**
2. TEMP keyword to create the table.
3. TEMPERARY keyword to create the table.
4. None of the above.

Q18. To create one table from another

1. CREATE TABLE new\_tbl FROM orig\_tbl;
2. **CREATE TABLE new\_tbl AS SELECT \* FROM orig\_tbl;**
3. CREATE TABLE orig\_tbl AS SELECT \* FROM new\_tbl;
4. Both A and B

Q19. Which of the following option is used to create an empty table based on the definition of another table.

1. CREATE TABLE new\_tbl FROM orig\_tbl;
2. CREATE TABLE new\_tbl AS orig\_tbl;
3. **CREATE TABLE new\_tbl LIKE orig\_tbl;**
4. CREATE TABLE new\_tbl IN orig\_tbl;

Q20. Syntax for creating views is

1. **CREATE VIEW AS SELECT**
2. CREATE VIEW AS UPDATE
3. DROP VIEW AS SELECT
4. CREATE VIEW AS UPDATE

Q21. You can delete a view with \_\_\_\_\_\_\_\_\_\_\_ command.

1. **DROP VIEW**
2. DELETE VIEW
3. REMOVE VIEW
4. TRUNCATE VIEW

Q22. You can update a view by using the following syntax:

1. CREATE OR UPDATE VIEW view\_name AS SELECT column\_name(s) FROM table\_name WHERE condition
2. UPDATE OR REPLACE VIEW view\_name AS SELECT column\_name(s) FROM table\_name WHERE condition
3. **CREATE OR REPLACE VIEW view\_name AS SELECT column\_name(s) FROM table\_name WHERE condition**
4. All of the above.

Q23. The following view statements contain the name, job title and the annual salary of employees working in the department 20:

1. create view DEPT20 is select ENAME, JOB, SAL\*12 ANNUAL SALARY from EMP where DEPTNO = 20;
2. **create view DEPT20 as select ENAME, JOB, SAL\*12 ANNUAL SALARY from EMP where DEPTNO = 20;**
3. create view DEPT20 as select ENAME, JOB, SAL\*12 ANNUAL SALARY from EMP where DEPTNO <> 20;
4. None of the above.

Q24. A file manipulation command that extracts some of the records from a file is called?

1. **SELECT**
2. PROJECT
3. JOIN
4. PRODUCT

Q25. The statement in SQL which allows to change the definition of a table is

1. **Alter.**
2. Update.
3. Create.
4. Select.

Q26. To delete a particular column in a relation the command used is:

1. UPDATE TABLE
2. TRUNCATE COLUMN
3. **ALTER, DROP**
4. DELETE COLUMN

Q27. The operator is used to compare a value to a list of literals values that have been specified.

1. BETWEEN
2. ANY
3. **IN**
4. ALL

Q28. Function divides one numeric expression by another and returns the remainder

1. POWER
2. **MOD**
3. ROUND
4. REMAINDER

Q29. A data manipulation command the combines the records from one or more tables is?

1. SELECT
2. PROJECT
3. **JOIN**
4. PRODUCT

Q30. Which special character is used to query all the columns from the table without listing each column by name?

1. %
2. &
3. @
4. **\***

Q31. What command is use to connect to MySQL server?

1. **c:\> mysql -h host -u user -p**
2. c:\> mysql -hostNM host -u user -p
3. c:\> mysql -hostname host -u user -p
4. All of the above.

Q32. Use the statement to find out what databases currently exist on the server:

1. **SHOW DATABASE;**
2. SHOW DATABASES();
3. SHOW DATABASES;
4. GET DATABASES;

Q33. For creating an empty copy of an existing table, write statement like this:

1. **CREATE TABLE new\_tbl\_name like tbl\_name**
2. CREATE TABLE new\_tbl\_name as tbl\_name
3. CREATE TABLE new\_tbl\_name as like tbl\_name
4. None of these.

Q34. What SQL statement will you provide to CREATE a duplicate table with structure and its records?

1. **CREATE table new\_tbl\_name as SELECT \* from tbl\_name;**
2. CREATE table new\_tbl\_name LIKE SELECT \* from tbl\_name;
3. CREATE table new\_tbl\_name AS GOOD AS SELECT \* from tbl\_name;
4. None of the above.

Q35. What SQL statement will you provide to CREATE a duplicate table with structure and its records?

1. **CREATE table new\_tbl\_name SELECT \* from tbl\_name;**
2. CREATE table new\_tbl\_name LIKE SELECT \* from tbl\_name;
3. CREATE table new\_tbl\_name AS GOOD AS SELECT \* from tbl\_name;
4. None of the above.

Q36. What is the proper syntax to insert multiple rows?

1. Insert into tbl\_name values (v1, v2, v3, …), values (v1, v2, v3, …);
2. Insert into tbl\_name multiple values (v1, v2, v3, …), (v1, v2, v3, …);
3. **Insert into tbl\_name values (v1, v2, v3, …), (v1, v2, v3, …);**
4. All of the above.

Q37. What is the proper syntax to insert multiple rows?

1. Insert all into tbl\_name values (v1, v2, v3, …), (v1, v2, v3, …);
2. **Insert into tbl\_name value (v1, v2, v3, …), (v1, v2, v3, …);**
3. Insert into tbl\_name multiple values (v1, v2, v3, …), (v1, v2, v3, …);
4. All of the above.

Q38. Which single-row function could you use to return a specific portion of a character string?

1. **INSTR**
2. SUBSTR
3. LPAD
4. LEAST

Q39. Which of the following MySQL function is used to returns the numeric value of the leftmost character of the string str.

1. ASC(str)
2. **ASCII(str)**
3. ANSI(str)
4. None of the above.

Q40. Which of the following MySQL function is used to return a string consisting of the characters given by the code values of those integers.

1. CHR(65)
2. **CHAR(65)**
3. CHARACTER(65)
4. NUMBER\_TO\_CHAR(65)

Q41. Which of the following MySQL function is used to find the length of string.

1. len(str)
2. **length(str)**
3. string\_length(str)
4. All of the above.

Q42. Which of the following MySQL function is used to find the length of string.

1. len(str)
2. Length\_of\_String(str)
3. **char\_length(str)**
4. All of the above.

Q43. Which of the following MySQL function is used to find the length of string.

1. len(str)
2. String\_length(str)
3. **character\_length(str)**
4. All of the above.

Q44. Which of the following MySQL function is used to return the string that results from concatenating the arguments.

1. **CONCAT(str1, str2)**
2. CONNECT(str1, str2)
3. COMBIND(str1, str2)
4. None of these.

Q45. Which of the following symbol is used to concatenating the arguments in MySQL.

1. +
2. |
3. ||
4. **None of the above.**

Q46. Which of the following MySQL function is used to return the Nth element of the list of strings.

1. **ELT()**
2. ELEMENT()
3. Nth\_ELEMENT()
4. None of the above.

Q47. Which of the following MySQL function is used to return the index (position) of str in the str1, str2, str3, ... list.

1. SUBSTR()
2. INSTR()
3. **FIELD()**
4. FIELDS()

Q48. Which of the following MySQL function is used to return the string in lower case.

1. **lcase(str)**
2. lower\_case(str)
3. case\_lower(str)
4. lowercase(str)

Q49. Which of the following MySQL function is used to return the string in lower case.

1. **lower(str)**
2. lower\_case(str)
3. case\_lower(str)
4. lowercase(str)

Q50. Which of the following MySQL function is used to return the leftmost len characters from the string str.

1. **LEFT(str, len)**
2. LEFT\_Most(str, len)
3. TO\_LEFT(str, len)
4. FROM\_LEFT(str, len)

Q51. Which of the following MySQL function is used to returns the string str, left-padded with the string *padstr* to a length of len characters.

1. LEFTPAD()
2. PADLEFT()
3. **LPAD()**
4. Both A and C

Q52. Which of the following MySQL function is used to returns the string str with leading space characters removed.

1. LPAD()
2. LEFT()
3. **LTRIM()**
4. LEFT\_TRIM()

Q53. Which of the following MySQL function is used to return a specified number of characters from a particular position of a given string.

1. **SUBSTR(str, pos)**
2. SUB\_STR(str, pos)
3. Both A and B
4. None of the above.

Q54. Which of the following MySQL function is used to return a specified number of characters from a particular position of a given string.

1. SUBSTR(str, pos)
2. SUBSTRRING(str, pos)
3. **Both A and B**
4. None of the above.

Q55. Which of the following are specifiers of MySQL TRIM()?

1. **BOTH**
2. BOTHSIDE
3. FORM\_BOTH
4. Both A and C

Q56. Which of the following are specifiers of MySQL TRIM()?

1. LEFT
2. LEFT\_SIDE
3. **LEADING**
4. None of the above.

Q57. Which of the following are specifiers of MySQL TRIM()?

1. RTRIM
2. **TRAILING**
3. Both A and B
4. None of the above.

Q58. Which of the following MySQL function is used to return the string in upper case.

1. **ucase(str)**
2. upper\_case(str)
3. case\_upper(str)
4. uppercase(str)

Q59. Which of the following MySQL function is used to return the string in upper case.

1. **upper(str)**
2. upper\_case(str)
3. case\_upper(str)
4. uppercase(str)

Q60. Which of the following MySQL function is used to return a specified number of characters from a particular position of a given string.

1. MOD(str, pos, len)
2. **MID(str, pos, len)**
3. MIDDLE(str, pos, len)
4. None of the above

Q61. Which of the following MySQL function is used to return the string str with all occurrences of the string from\_str replaced by the string to\_str.

1. **REPLACE(str, from\_str, to\_str)**
2. REPLACE\_STR(str, from\_str, to\_str)
3. UPDATE(str, from\_str, to\_str)
4. Both A and B.

Q62. Which of the following MySQL function is used to return the string str with the order of the characters reversed.

1. **REVERSE(str)**
2. STRREVERSE(str)
3. STRING\_REVERSE(str)
4. None of the above.

Q63. Which of the following MySQL function is used to return the rightmost len characters from the string str.

1. RIGHTMOST(str, len)
2. FROM\_RIGHT(str, len)
3. WRITE(str, len)
4. **RIGHT(str, len)**

Q64. Which of the following MySQL function is used to return the string str, right-padded with the string padstr to a length of len characters.

1. **RPAD(str, len, padstr)**
2. RIGHTPAD(str, len, padstr)
3. RIGHTP(str, len, padstr)
4. Both A and B

Q65. Which of the following MySQL function is used to return the string str with trailing space characters removed.

1. **RTRIM(str)**
2. RIGHT(str)
3. TRIM\_RIGHT(str)
4. All of the above.

Q66. With LIKE you can use the following two wildcard characters in the pattern:

1. % matches any number of characters, even zero characters.
2. \_ matches exactly one character.
3. **Both A and B**
4. None of the above.

Q67. With LIKE you can use the following two wildcard characters in the pattern:

1. \% matches any number of characters, even zero characters.
2. \\_ matches exactly one character.
3. Both A and B
4. **None of the above.**

Q68. With LIKE you can use the following two wildcard characters in the pattern:

1. $ matches any number of characters, even zero characters.
2. & matches exactly one character.
3. Both A and B
4. **None of the above.**

Q69. With LIKE you can use the following two wildcard characters in the pattern:

1. @ matches any number of characters, even zero characters.
2. \* matches exactly one character.
3. Both A and B
4. **None of the above.**

Q70. Which of the following MySQL function is used to returns the average value of expr.

1. **AVG(expr)**
2. AVERAGE(expr)
3. GET\_AVG(expr)
4. Both A and B

Q71. Which of the following MySQL function is used to return the average of the distinct values of expr.

1. **AVG(DISTINCT expr)**
2. AVGERAGE(DISTINCT expr)
3. DISTINCT(AVG expr)
4. DISTINCT AVG expr

Q72. Which of the following MySQL function is used to return a count of the number of non-NULL values of expr in the rows retrieved by a SELECT statement.

1. CNT(expr)
2. **COUNT(expr)**
3. COUNTIF(expr)
4. COUNT\_NOTNULL(expr)

Q73. Which statement gives you the version of MySQL?

1. **SELECT VERSION ();**
2. SELECT VER();
3. SELECT VERSIONS ();
4. SELECT VERS()

Q74. Which object are created using CREATE statement in MySQL?

1. **Database**
2. Class
3. Row
4. Column

Q75. Which of the following SQL statements will generate an error when executed.

1. CREATE DATABASE `students`;
2. CREATE DATABASE students;
3. CREATE DATABASE IF NOT EXISTS students;
4. **IF NOT EXISTS CREATE DATABASE `STUDENTS`;**

Q76. Which of the following SQL statements will generate an error when executed.

1. CREATE TABLE `students` (c1 int, c2 int);
2. CREATE TABLE `students` (`c1` int, c2 int);
3. CREATE DATABASE IF NOT EXISTS students (c1 int, c2 int);
4. **IF NOT EXISTS CREATE DATABASE `STUDENTS` (c1 int, c2 int);**

Q77. Which of the following scripts will run successfully?

1. SELECT customer name FROM customers;
2. SELECT FROM `customers` 'customer name';
3. SELECT `customer name` FROM customers ORDER BY zone WHERE cat\_id = 12;
4. **SELECT `customer name` FROM customers WHERE cat\_id = 12 ORDER BY cat\_id;**

Q78. What command is used to permanently remove a record from a database table?

1. DROP
2. REMOVE
3. **DELETE**
4. CUT

Q79. Which statement are used to combine the tables including the duplicate rows.

1. JOIN
2. **UNION ALL**
3. LEFT JOIN
4. OUTER JOIN

Q80. Which statement is used to combine the table but does not look for duplicate rows.

1. UNION ALL
2. **UNION**
3. LEFT JOIN
4. OUTER JOIN

Q81. Which function is used to return specific portion of string in a given string.

1. **SUBSTR**
2. INSTR
3. STRSTR
4. STRISTR

Q82. Which function gives character position in a specified string?

1. SUBSTR
2. STRSTR
3. **INSTR**
4. STRISTR

Q83. How do I find out all databases starting with ‘test‘?

1. SHOW DATABASES LIKE '%test% ';
2. SHOW DATABASES LIKE '%test';
3. SHOW DATABASES LIKE 'test'%';
4. **SHOW DATABASES LIKE 'test%';**

Q84. How can we get the number of records or rows in a table?

1. **Using COUNT**
2. Using NUM
3. Using NUMBER
4. Both a and c

Q85. Which of the following will insert a record into EMP table having empno and ename as columns?

1. **INSERT INTO EMP VALUES (1000,'SRI');**
2. INSERT EMP VALUES (1000,'SRI');
3. INSERT FROM EMP VALUES (1000,'SRI');
4. INSERT EMP (1000,'SRI');

Q86. The SQL function ABS(-50) gives value as

1. -50
2. -1
3. 0
4. **50**

Q87. With SQL, how do you select all the records from a table named "Persons" where the value of the column "FirstName" starts with an "as"?

1. SELECT \* FROM Persons WHERE FirstName LIKE '%as'
2. SELECT \* FROM Persons WHERE FirstName='%as%'
3. **SELECT \* FROM Persons WHERE FirstName LIKE 'as%'**
4. SELECT \* FROM Persons WHERE FirstName='as'

Q88. In a LIKE clause, you can ask for any 6 letter value by writing:

1. **LIKE \_\_\_\_\_\_ (that's six underscore characters)**
2. LIKE ...... (that's six dots)
3. LIKE .{6}
4. LIKE ??????

Q89. The command to remove rows from a table 'CUSTOMER' is:

1. REMOVE FROM CUSTOMER ...
2. DROP FROM CUSTOMER ...
3. **DELETE FROM CUSTOMER WHERE ...**
4. UPDATE FROM CUSTOMER ...

Q90. Which of the following MySQL function is used to extracts the date part of the date or datetime expression.

1. **DATE(expr)**
2. TO\_DATE(expr)
3. NOW(expr)
4. Both A and C.

Q91. Which of the following MySQL function is used to return the time value after which a certain time interval has been added.

1. **ADDTIME()**
2. ADD\_TIME()
3. TIMEADD()
4. None of the above

Q92. Which of the following MySQL function is used to return the time value after which a certain time interval has been subtracted.

1. **SUBTIME()**
2. SUB\_TIME()
3. TIMESUB()
4. None of the above

Q93. Which of the following MySQL function is used to return the name of the weekday for date.

1. WEEKDAYNAME
2. **DAYNAME**
3. WEEK\_NAME
4. DAY\_NAME

Q94. Which of the following MySQL function is used to return the day of the month for date.

1. **DAYOFMONTH()**
2. MONTH\_DAY
3. DAY\_OF\_MONTH
4. None of the above.

Q95. Write a SQL statement to create a table countries set a constraint NULL.

1. **CREATE TABLE IF NOT EXISTS countries (COUNTRY\_ID varchar(2) NOT NULL, COUNTRY\_NAME varchar(40) NOT NULL, REGION\_ID decimal(10,0) NOT NULL);**
2. CREATE TABLE IF NOT EXISTS countries (COUNTRY\_ID varchar(2) NOTNULL, COUNTRY\_NAME varchar(40) NOTNULL, REGION\_ID decimal(10,0) NOTNULL);
3. CREATE TABLE IF NOT EXISTS countries (COUNTRY\_ID varchar(2) IS NOT NULL, COUNTRY\_NAME varchar(40) IS NOT NULL, REGION\_ID decimal(10,0) IS NOT NULL);
4. All of the above.

Q96. Which of the following option will you use to create temporary table?

1. CREATE not permanent table tbl\_name ...
2. CREATE non-permanent table tbl\_name ...
3. CREATE short-term table tbl\_name ...
4. **None of the above.**

Q97. Which of the following option will you use to create temporary table?

1. CREATE not permanent table tbl\_name ...
2. CREATE non-permanent table tbl\_name ...
3. **CREATE temporary table tbl\_name ...**
4. All of the above.

Q98. What is the syntax for CREATE TABLE?

1. **CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl\_name**
2. CREATE [TEMPORARY] TABLE [NOT IF EXISTS] tbl\_name
3. CREATE [TEMPORARY] TABLE [EXISTS IF NOT] tbl\_name
4. Both A and B

Q99. The user wants to create STUDENT table having columns (ID, NAME, and ADDRESS) where ID must be auto\_increment. Which of the following option user will use to create the table.

1. **CREATE TABLE STUDENT (ID int primary key auto\_increment, NAME varchar(10), ADDRESS text)**
2. CREATE TABLE STUDENT (ID int auto\_increment primary, NAME varchar(10), ADDRESS text);
3. CREATE TABLE STUDENT (ID int auto\_increment\_by primary key, NAME varchar(10), ADDRESS text);
4. Bot A and C

Q100. The user wants to create STUDENT table having columns (ID, NAME, and ADDRESS) where ID must be auto\_increment. Which of the following option user will use to create the table.

1. **CREATE TABLE STUDENT (ID int auto\_increment primary key, NAME varchar(10), ADDRESS text);**
2. CREATE TABLE STUDENT (ID int auto\_increment primary, NAME varchar(10), ADDRESS text);
3. CREATE TABLE STUDENT (ID int auto\_increment\_by primary key, NAME varchar(10), ADDRESS text);
4. Bot A and C

Q101. The user wants to drop multiple table (STUDENTS and COURSES), what command he will issue.

1. DROP TABLE STUDENT and COURSES
2. **DROP TABLE STUDENT, COURSES**
3. DROP multiple TABLE (STUDENT and COURSES)
4. None of the above.

Q102. The user wants to drop multiple table (STUDENTS and COURSES), what command he will issue.

1. DROP TABLE STUDENT and COURSES
2. DROP TABLE (STUDENT and COURSES)
3. **DROP TABLE STUDENT, COURSES**
4. Both B and C

Q103. Write a SQL statement to create a table named countries including columns id, country\_name and make sure that no countries except Italy, India and USA will be entered in the table.

1. **CREATE TABLE COUNTRIES (ID int, country\_name enum ('Italy','India','China')) engine=InnoDB;**
2. CREATE TABLE COUNTRIES (ID int, country\_name list ('Italy','India','China')) engine=InnoDB;
3. CREATE TABLE COUNTRIES (ID int, country\_name in ('Italy','India','China')) engine=InnoDB;
4. CREATE TABLE COUNTRIES (ID int, country\_name like ('Italy','India','China')) engine=InnoDB;

Q104. To drop multiple columns in a single statement, do this:

1. ALTER TABLE tbl\_name DROP COLUMN (col1, col2);
2. ALTER TABLE tbl\_name DROP COLUMNS (col1, col2);
3. ALTER TABLE tbl\_name DROP COLUMN col1, col2;
4. **ALTER TABLE tbl\_name DROP COLUMN col1, DROP COLUMN col2;**

Q105. If STUDENT table is currently not an InnoDB table, what statement changes its storage engine to InnoDB?

1. **ALTER TABLE STUDENT ENGINE = InnoDB;**
2. ALTER TABLE STUDENT CHANGE ENGINE = InnoDB;
3. ALTER TABLE STUDENT CHANGE MyISAM to InnoDB
4. ALTER TABLE STUDENT CHANGE TO InnoDB from MyISAM.

Q106. To reset the current auto-increment value:

1. ALTER AUTO\_INCREMENT = 13;
2. **ALTER TABLE STUDENT AUTO\_INCREMENT = 13;**
3. ALTER TABLE STUDENT AUTO\_INCREMENT BY 13;
4. ALTER TABLE STUDENT AUTO\_INCREMENT TO 13;

Q107. What is true about ALTER TABLE OPERATIONS?

1. That are not performed in place,
2. That are performed in place,
3. Make a temporary copy of the original table.
4. **Both A and C**

Q108. While ALTER TABLE is executing,

1. **The original table is readable by other sessions.**
2. The original table is writable by other sessions.
3. The original table is readable and writable by other sessions.
4. None of the above.

Q109. You can rename a column using a

1. ALTER TABEL STUDENT RENAME old\_col\_name new\_col\_name column\_definition clause
2. ALTER TABEL STUDENT RENAME old\_col\_name to new\_col\_name column\_definition clause
3. **ALTER TABEL STUDENT CHANGE old\_col\_name to new\_col\_name** **column\_definition clause**
4. Both A and C

Q110. To change a column's data type but not the name, which of the following MySQL statement you will issue.

1. ALTER TABLE STUDENT CHANGE id id INT;
2. ALTER TABLE STUDENT MODIFY id INT;
3. **Both A and B**
4. None of the above.

Q111. How to drop the primary key.

1. **ALTER TABLE *tbl\_name* DROP PRIMARY KEY;**
2. ALTER TABLE *tbl\_name* DROP COLUMN PRIMARY KEY;
3. ALTER TABLE *tbl\_name* PRIMARY KEY DROP;
4. ALTER TABLE *tbl\_name* DROP PRIMARY

Q112. To insert "INSERT INTO countries VALUES (501, 'India', 185);" a record into the table countries to ensure that, a country\_id and region\_id combination will be entered once in the table. What of the following options you will apply to create the countries table.

1. CREATE TABLE countries (COUNTRY\_ID integer, COUNTRY\_NAME varchar(40) NOT NULL, REGION\_ID integer, PRIMARY KEY COUNTRY\_ID, REGION\_ID);
2. CREATE TABLE countries (COUNTRY\_ID integer PRIMARY KEY, COUNTRY\_NAME varchar(40) NOT NULL, REGION\_ID integer PRIMARY KEY);
3. **CREATE TABLE countries (COUNTRY\_ID integer, COUNTRY\_NAME varchar(40) NOT NULL, REGION\_ID integer, PRIMARY KEY (COUNTRY\_ID, REGION\_ID));**
4. CREATE TABLE countries (COUNTRY\_ID integer, COUNTRY\_NAME varchar(40) NOT NULL, REGION\_ID integer, PRIMARY KEY (COUNTRY\_ID and REGION\_ID));

Q113. Write a query to display the names (first\_name, last\_name) and department ID of all employees in departments 30 or 100 in ascending alphabetical order by department ID.

1. SELECT first\_name, last\_name, department\_id FROM employees WHERE department\_id IN (30, 100) ORDER department\_id BY ASC;
2. **SELECT first\_name, last\_name, department\_id FROM employees WHERE department\_id IN (30, 100) ORDER BY department\_id ASC;**
3. SELECT first\_name, last\_name, department\_id FROM employees WHERE department\_id IN (30, 100) ORDER BY USING (department\_id) ASC;
4. None of the above

Q114. Write a query to display the names (first\_name, last\_name) and salary for all employees whose salary is not in the range $10,000 through $15,000 and are in department 30 or 100.

1. SELECT first\_name, last\_name, salary, department\_id FROM employees WHERE salary != (10000 AND 15000) AND department\_id IN (30, 100);
2. **SELECT first\_name, last\_name, salary, department\_id FROM employees WHERE salary NOT BETWEEN 10000 AND 15000 AND department\_id IN (30, 100);**
3. SELECT first\_name, last\_name, salary, department\_id FROM employees WHERE salary <> 10000 AND sal <> 15000 AND department\_id IN (30, 100);
4. Both A and C

Q115. What is to equivalent: SELECT SAL FROM EMP WHERE SAL NOT BETWEEN 1000 AND 3000;

1. **SELECT sal from EMP where sal < 1000 or sal> 3000;**
2. SELECT sal from EMP where sal < 1000 and sal > 3000;
3. SELECT sal from EMP where sal <> 1000 or sal<>3000;
4. SELECT sal from EMP where sal <> 1000 and sal<>3000;

Q116. "IS NULL" is the keyword that performs the

1. **Boolean comparison**
2. String comparison
3. Numerical comparison
4. Both A and B

Q117. "IS NOT NULL" is the keyword that performs the

1. **Boolean comparison**
2. String comparison
3. Numerical comparison
4. Both A and B

Q118. What are the names of the tables (or relations) in sample\_db? Use the command:

1. show table.
2. **show tables.**
3. show relations
4. None of the above.

Q119. Write a SQL statement to display a string "This is SQL Exercise, Practice and Solution".

1. **SELECT "This is SQL Exercise, Practice and Solution";**
2. SELECT data as "This is SQL Exercise, Practice and Solution";
3. SELECT \* from "This is SQL Exercise, Practice and Solution";
4. This is not possible.

Q120. Write a SQL statement to find those salesmen with all information who come from the city either Paris or Rome.

**Sample table: SALESMAN (**salesman\_id, name, city, commission)

1. SELECT \* FROM salesman WHERE city = 'Paris' OR city = 'Rome';
2. SELECT \* FROM salesman WHERE city <=> 'Paris' OR city <=> 'Rome';
3. SELECT \* FROM salesman WHERE city in( 'Paris', 'Rome');
4. **All of the above.**

Q121. Write a query to filter those salesmen with all information who likes to leave other cities than Paris and Rome.

1. SELECT \* FROM salesman WHERE city NOT IN ('Paris', 'Rome');
2. SELECT \* FROM salesman WHERE NOT city IN ('Paris', 'Rome');
3. **Both A and B**
4. None of the above.

Q122. Write a query to sort out those customers with all information whose ID value is within any of 3007, 3008 and 3009.

1. SELECT \* FROM customer WHERE customer\_id IN (3007, 3008, 3009);
2. SELECT \* FROM customer WHERE customer\_id IN 3007 or 3008 or 3009;
3. SELECT \* FROM customer WHERE customer\_id IN ('3007', '3008', '3009');
4. **Both A and C**

Q123. Write a SQL statement to find those salesmen with all information who get the commission within a range of 0.12 and 0.14.

1. **SELECT \* FROM salesman WHERE commission BETWEEN 0.12 AND 0.14;**
2. SELECT \* FROM salesman WHERE commission BETWEEN (0.12 AND 0.14);
3. SELECT \* FROM salesman WHERE commission IN BETWEEN 0.12 AND 0.14;
4. None of the above.

Q124. User defined variables are written as

1. **@var\_name**
2. $var\_name
3. &var\_name
4. None of the above.

Q125. How will you create user defined variable in MySQL?

1. SET x := 1001;
2. **SET @x := 1001;**
3. SET &x := 1001;
4. None of the above

Q126. How will you get the value from user defined variable in MySQL?

1. SELECT x from dual;
2. SELECT &x from dual;
3. **SELECT @x from dual;**
4. Both A and C

Q127. What MySQL select statement you will write to display increased salary of all employees from EMP table, by Rs.1000.

1. SELECT sal 'Original Salary', sal+1000 'Increased Salary' from EMP;
2. SELECT sal 'Original Salary', @'Increase Salary by 1000' := 1000, sal + @'Increase Salary by 1000' from EMP;
3. SELECT sal 'Original Salary', @'Increase Salary by 1000' := 1000 + sal from EMP;
4. **All of the above.**

Q128. What MySQL select statement you will write to display increased salary of all employees from EMP table, by Rs.1000.

1. **SELECT sal 'Original Salary', sal + 1000 'Increased Salary' from EMP;**
2. SELECT sal 'Original Salary', @'Increase Salary by 1000' = 1001 + sal from EMP;
3. SELECT sal 'Original Salary', @'Increase Salary by 1000' = 1000, sal + @'Increase Salary by 1000' from EMP;
4. All of the above.

Q129. Which command to use in order to delete the data inside the table, and not the table itself.

1. DELETE
2. TRUNCATE
3. **Both TRUNCATE & DELETE**
4. DROP

Q130. Which one is correct syntax for Insert Statement?

1. Insert table\_name Columns (Col1, Col2, Col3);
2. **Insert into table\_name (Col1, Col2, Col3) VALUES (Val1, Val2, Val3);**
3. Insert Columns (Col1, Col2, Col3) VALUE (Val1, Val2, Val3) into table\_name;
4. None of the above.

Q131. Which one is correct syntax for Update Statement?

1. Update Table table\_name Columns (Col1, Col2, Col3);
2. Update into table\_name (Col1, Col2, Col3) VALUES (Val1, Val2, Val3);
3. **Update table\_name Set Col\_name=Value;**
4. None of the above

Q132. What will be the consequence of omitting ‘Where’ clause in Update Statement?

1. No effect on the query as well as on table.
2. **All records present in the table will be updated**
3. Only one record will be updated
4. None of the above

Q133. In any case, can update statement be used in place of insert statement?

1. **Yes, if record exists with Identity value.**
2. Yes, in every case
3. No, it is not possible at all.
4. None of the above.

Q134. Which one is correct syntax for Where clause in MySQL.

1. SELECT WHERE "Condition" Col1, Col2 FROM "Table”;
2. SELECT "Condition" Col1, Col2 FROM "Table" WHERE;
3. **SELECT Col1, Col2 FROM "Table" WHERE "condition";**
4. None of the above

Q135. Which one is correct syntax for applying UNION operator?

1. SELECT column\_name(s) FROM table\_name1 UNION table\_name2
2. **SELECT column\_name(s) FROM table\_name1 UNION SELECT column\_name(s) FROM table\_name2**
3. UNION SELECT column\_name(s) FROM table\_name1 SELECT column\_name(s) FROM table\_name2
4. SELECT FROM table\_name1 AND table\_name2

Q136. . How can we get all records (redundant as well as non-redundant) from union operator?

1. **Using 'ALL' operator with UNION.**
2. Using 'Distinct' operator with UNION.
3. We get all records (redundant as well as non-redundant) with UNION operator by default.
4. None of the above.

Q137. What is true about order by with Union operator?

1. Order By can be issued in each result set.
2. **It can be issued for the overall result set.**
3. Both A & B.
4. None of the above.

Q138. Which arithmetic operators can be apply with date?

1. +
2. -
3. \*
4. **Both A and B**

Q139. Which arithmetic operators can be apply with date?

1. **+**
2. /
3. \*
4. None of the above

Q140. Which arithmetic operators can be apply with date?

1. \
2. /
3. \*
4. **None of the above**

Q141. Which SELECT statement would you apply in a stored procedure to query the employee table and retrieve the last name and salary of the employee whose ID is 3?

1. SELECT last\_name, salary FROM employee;
2. **SELECT last\_name, salary FROM employee WHERE id=3;**
3. SELECT l last\_name, salary FROM employee INTO v\_last\_name, v\_salary WHERE id=3;
4. None of the above.

Q142. Which SELECT statement would you apply in a stored procedure to query the employee table and retrieve the last name and salary of the employee whose ID is 3?

1. SELECT last\_name, salary FROM employee;
2. **SELECT last\_name, salary INTO v\_last\_name, v\_salary FROM employee WHERE id=3;**
3. SELECT l last\_name, salary FROM employee INTO v\_last\_name, v\_salary WHERE id=3;
4. All of the above

Q143. What is the proper sequence of clauses you will apply for SELECT statement in stored procedure?

1. SELECT FROM INTO WHERE
2. SELECT INTO WHERE FROM
3. **SELECT INTO FROM WHERE**
4. Both A and C

Q144. What command you will apply to modify the existing VIEW?

1. **ALTER VIEW...**
2. UPDATE VIEW…
3. MODIFY VIEW…
4. None of the above.

Q145. What command you will apply to CREATE or REPLACE the existing VIEW?

1. **CREATE OR REPLACE VIEW …**
2. CREATE AND REPLACE VIEW …
3. CREATE OR CHANGE VIEW…
4. None of the above.

Q146. Which script would you use to query the data dictionary to view primary key constraints?

1. SELECT \* from information\_schema.table\_constraint;
2. SELECT \* from information\_schema.table\_const\_columns;
3. **SELECT \* from information\_schema.table\_constraints;**
4. SELECT \* from information\_schema.constraints;

Q147. Which statement would you use to add a primary key constraint to the patient table using the id\_number column immediately enabling the constraint?

1. This task can’t be accomplished.
2. **ALTER TABLE patient ADD CONSTRAINT pat\_id\_pk PRIMARY KEY (id\_number);**
3. ALTER TABLE patient ADD (id\_number CONATRAINT pat\_id\_pk PRIMARY KEY);
4. ALTER TABLE patient MODIFY (id\_number CONSTRAINT pat\_id\_pk PRIMARY KEY);

Q148. What statement will you apply to show the length of names in EMP table. Eliminate duplicate length. Do not show the names.

1. SELECT unique length (ename) from EMP;
2. **SELECT distinct length (ename) from EMP;**
3. SELECT length (distinct ename) from EMP;
4. All of the above.

Q149. What SQL statement will you apply to find the average annual salary per job in each dept.?

1. **SELECT job, deptno, avg (sal) from EMP group by job, deptno;**
2. SELECT job, deptno, avg (sal) from EMP group by deptno;
3. SELECT job, deptno, average (sal) from EMP group by job, deptno;
4. SELECT job, deptno, avg sal from EMP group by job, deptno;

Q150. Calculate the avg, min and max salary of those groups of employees having the job as CLERK or MANAGER.

1. SELECT job, avg (Sal), min(sal), max(sal) from EMP where job in ('CLERK', 'MANAGER') group by job;
2. SELECT job, avg (Sal), min(sal), max(sal) from EMP where job in = 'CLERK' or job = 'MANAGER' group by job;
3. SELECT job, avg (Sal), min(sal), max(sal) from EMP where job like = 'CLERK' or job like = 'MANAGER' group by job;
4. **Both A and B**

Q151. When you make a column an identity column, MySQL automatically assigns a \_\_\_\_\_\_\_\_ number to this column with every row you insert.

1. next
2. random
3. decimal
4. **sequenced**

Q152. What statement will you apply to print all the orders showing order number, amount, company name and credit limit of customers.

Orders table: Order\_no, Cust, Prodt, Qty, Amt, Discount

Customers table: Custnbr, Company, Custrep, Creditlim

1. Select Order\_no, Amt, Company, Creditlim from Customers outer join Orders on customers.custnbr = orders.cust;
2. Select Order\_no, Amt, Company, Creditlim from Customers left outer join Orders on customers.custnbr = orders.cust;
3. **Select Order\_no, Amt, Company, Creditlim from Customers inner join Orders on customers.custnbr = orders.cust;**
4. Select Order\_no, Amt, Company, Creditlim from Customers right outer join Orders on customers.custnbr = orders.cust;

Q153. Find all the customers with orders more than 500 or credit limits greater than or equal to 500.

Customers table: Custnbr, Company, Custrep, Creditlim

Orders table: Order\_no, Cust, Prodt, Qty, Amt, Discount

1. Select distinct Custnbr from Customers Right JOIN Orders on Custnbr = Cust where (Creditlim >= 500 OR Amt > 500)
2. Select distinct Custnbr from Customers LEFT JOIN Orders on Custnbr = Cust where (Creditlim > 500 OR Amt >= 500)
3. Select Custnbr from Customers LEFT JOIN Orders on Custnbr = Cust where (Creditlim > 500 OR Amt > 500)
4. **Select distinct Custnbr from Customers LEFT JOIN Orders on Custnbr = Cust where (Creditlim >= 500 OR Amt > 500)**

Q154. What will the result of the following statement?

SELECT \* from EMP join DEPT where null=null;

1. False
2. True
3. Error
4. **Empty set**

Q155. Write a SQL statement to display the commission with the percent sign ( % ) with salesman ID, name and city columns for all the salesmen

Sample table: salesman

salesman\_id, name, city, commission

1. SELECT salesman\_id, name, city,'%' ,commission\*100 FROM salesman;
2. SELECT salesman\_id, name, city,"%", commission\*100 FROM salesman;
3. SELECT salesman\_id, name, city, %, commission\*100 FROM salesman;
4. **Both A and B**

Q156. Write a SQL statement to find out the number of orders booked for each day and display it in such a format like "For 2001-10-10 there are 15 orders".

Sample table: orders

ord\_no, purch\_amt, ord\_date, customer\_id, salesman\_id

1. **SELECT ' For', ord\_date, ‘there are', COUNT (DISTINCT ord\_no), 'orders.' FROM orders GROUP BY ord\_date;**
2. SELECT ' For' || ord\_date || ', there are', COUNT (DISTINCT ord\_no) || 'orders.' FROM orders GROUP BY ord\_date;
3. SELECT ' For' + ord\_date + ', there are', COUNT (DISTINCT ord\_no) + 'orders.' FROM orders GROUP BY ord\_date;
4. SELECT ' For' # ord\_date # ', there are', COUNT (DISTINCT ord\_no) # 'orders.' FROM orders GROUP BY ord\_date;

Q157. Write a query to display the orders according to the order number arranged by ascending order.

Sample table: orders

ord\_no, purch\_amt, ord\_date, customer\_id, salesman\_id

1. SELECT \* FROM orders ORDER BY ord\_no;
2. SELECT \* FROM orders ORDER BY ord\_no asc;
3. SELECT \* FROM orders ORDER BY ascending order on ord\_no;
4. **Both A and B**

Q158. Write a SQL statement to arrange the orders according to the order date in such a manner that the latest date will come first then previous dates.

Sample table: orders

ord\_no, purch\_amt, ord\_date, customer\_id, salesman\_id

1. **SELECT \* FROM orders ORDER BY ord\_date DESC;**
2. SELECT \* FROM orders ORDER BY DESC (ord\_date);
3. SELECT \* FROM orders SORT BY ord\_date DESC;
4. SELECT \* FROM orders ORDERED BY ord\_date DESC;

Q159. Write a SQL statement to display the orders with all information in such a manner that, the older order date will come first and the highest purchase amount of same day will come first.

Sample table: orders

ord\_no, purch\_amt, ord\_date, customer\_id, salesman\_id

1. **SELECT \* FROM orders ORDER BY ord\_date, purch\_amt DESC;**
2. SELECT \* FROM orders ORDER BY ord\_date DESC, purch\_amt ASC;
3. SELECT \* FROM orders ORDER BY DESC ord\_date, DESC purch\_amt;
4. All of the above

Q160. Write a SQL statement to prepare a list with salesman name, customer name and their cities for the salesmen and customer who belongs to the same city.

Sample table: salesman

salesman\_id, name, city, commission

Sample table: customer

customer\_id, cust\_name, city, grade, salesman\_id

1. SELECT s.name "Salesman", c.cust\_name, c.city FROM salesman s, customer c WHERE s.city = c.city;
2. SELECT salesman.name AS "Salesman", customer.cust\_name, customer.city FROM salesman, customer WHERE salesman.city = customer.city;
3. **Both A and B**
4. None of the above.

Q161. Which command you will apply to show the table definition in Mysql?

1. DESC TABLE\_NAME;
2. EXPLAIN TABLE\_NAME
3. SHOW TABLE\_NAME
4. **BOTH A AND B**

Q162. Which among the following is the correct syntax for defining “Constraint Check” in Mysql?

1. **gender char(1) check( gender IN (‘M’, ‘F’)),**
2. gender char (1) check,
3. gender char (1) check ( gender ),
4. None of these

Q163. Which data type character merge the "Check Constraint" into a data type definition?

1. **ENUM**
2. ENUM1
3. ENUM2
4. None of these

Q164. Which among the following is the correct syntax for defining "ENUM" in Mysql?

1. **gender ENUM ('M', 'F'),**
2. gender ENUM,
3. gender ENUM ( ),
4. None of these

Q165. Which among the following is the correct syntax for defining "SET" in Mysql?

1. **gender SET ('M', 'F'),**
2. gender SET,
3. gender SET ( ),
4. None of these

Q166. Which among the following is the correct syntax for modifying the definition of an existing table?

1. **ALTER TABLE person MODIFY person\_id SMALLINT AUTO\_INCREMENT;**
2. ALTER TABLE person person\_id SMALLINT AUTO\_INCREMENT;
3. ALTER TABLE person MODIFY person\_id;
4. ALTER TABLE person

Q167. Examine the following code.

DROP FUNCTION if EXISTS F1;

delimiter $$

CREATE function F1(x INT) RETURNS int

BEGIN

RETURN x;

END$$

delimiter ;

Which command you will apply to execute the FUNCTION.

1. CALL F1
2. EXECUTE F1
3. RUN F1
4. **None of the above.**

Q168. Which clause is applied to "filter out unwanted Groups"?

1. **HAVING**
2. FROM
3. WHERE
4. SELECT

Q169. Which clause is applied to "sort the rows of the final result set by one or more columns"?

1. HAVING
2. **ORDER BY**
3. WHERE
4. FROM

Q170. Which clause is applied to "Modify the existing field of the table"?

1. **ALTER**
2. FROM
3. SELECT
4. MODIFY

Q171. Which among the following is not applied in "query clause"?

1. WHERE
2. **MODIFY**
3. LIMIT
4. FROM

Q172. "MODIFY" is applied with which "Query clause"?

1. **ALTER**
2. FROM
3. WHERE
4. ORDER BY

Q173. Consider the following schema −

LOCATIONS (subject\_code, department\_name, location\_id, city);

Which code snippet will alter the table LOCATIONS and change the datatype of the column CITY to varchar(30) in MySQL?

1. ALTER TABLE locations MODIFY COLUMN city varchar(30);
2. MODIFY TABLE locations ADD (city varchar(30));
3. ALTER TABLE locations MODIFY city varchar(30);
4. **Both A and C**

Q174. Consider the following schema −

STUDENTS (student\_code, first\_name, last\_name, email, phone\_no, date\_of\_birth, honours\_subject, percentage\_of\_marks)

Which of the following query would display names of all the students whose honours subject is English and percentage of marks more than 80, or honours subject is Spanish and percentage of marks more than 80?

1. **SELECT first\_name, last name from STUDENTS where (honours\_subject = “English” or honours\_subject = “Spanish” ) and percentage\_of\_marks > 80;**
2. SELECT first\_name, last name from STUDENTS where honours\_subject = “English” or honours\_subject = “Spanish” and percentage\_of\_marks > 80;
3. SELECT first\_name, last name from STUDENTS where (honours\_subject = “English” or honours\_subject = “Spanish” and percentage\_of\_marks > 80);
4. SELECT first\_name, last name from STUDENTS where (honours\_subject = “English”) or honours\_subject = “Spanish” and percentage\_of\_marks > 80;

Q175. What command you will give to reset Auto-increment next value

1. **ALTER TABLE tbl\_name SET AUTO\_INCREMENT=val;**
2. ALTER AUTO\_INCREMENT SET=val;
3. ALTER TABLE tbl\_name RESET AUTO\_INCREMENT=val;
4. Both A and C

Q176. Write the SQL statement to display all the records in descending order by commission, but null commission must placed at top.

1. **SELECT \* from EMP order by isnull(comm) desc, comm desc;**
2. SELECT \* from EMP order by comm desc, isnull(comm) desc;
3. SELECT \* from EMP order by isnull(comm) desc , comm;
4. SELECT \* from EMP order by comm desc, isnull(comm) asc;

Q177. Which of the following SQL statement you will apply to display the count of all employees whose name starts with 'A'.

1. **SELECT ENAME, COUNT(\*) FROM EMP WHERE ENAME LIKE 'A%' GROUP BY ENAME;**
2. SELECT ENAME, COUNT(\*) FROM EMP GROUP BY ENAME WHERE ENAME LIKE 'A%';
3. SELECT ENAME, COUNT(\*) FROM EMP WHERE ENAME = 'A%' GROUP BY ENAME;
4. Any of the above.

Q178. Which of the following SQL statement you will display addition of two number.

Q179. You have issued the following set of statements

SET @mysize = 'medium';

CREATE TABLE sizes (size ENUM('small', @mysize, 'large'));

1. You also cannot employ a user variable as an enumeration value.
2. This pair of statements do not work.
3. **Both A and B**
4. No error

Q180. You have issued the following set of statements

CREATE TABLE TEMP (col SET('a', 'b', 'c', 'd'));

Line 1 - INSERT INTO TEMP (col) VALUES ('a,d');

Line 2 - INSERT INTO TEMP (col) VALUES ('d,a');

Line 3 - INSERT INTO TEMP (col) VALUES ('a,d,a');

Line 4 - INSERT INTO TEMP (col) VALUES ('a, d, a');

Which INSERT statement will raise an error?

1. Line 1
2. Line 2
3. Line 3
4. **Line 4**

Q181. You have issued the following set of statements

CREATE TABLE TEMP (col SET('a', 'b', 'c', 'd'));

Line 1 - INSERT INTO TEMP (col) VALUES ('a,d');

Line 2 - INSERT INTO TEMP (col) VALUES ('d,a');

Line 3 - INSERT INTO TEMP (col) VALUES ('a,d,a');

Line 4 - INSERT INTO TEMP (col) VALUES ('a,d,z');

Which INSERT statement will raise an error?

1. Line 1
2. Line 2
3. Line 3
4. **Line 4**

Q182. What happens when you issue the following statement?

CREATE TABLE myset (col SET('a', 'b', 'c', 'd', 'a'));

1. The table is created.
2. Error, because double quotes (") must be used for passing set values.
3. **Error, because Column 'col' has duplicated value 'a' in SET.**
4. Invalid SET options.

Q183. The USER has issued the following set of statements.

CREATE TABLE TEMP (COL1 int ,COL2 bool, COL3 boolean);

INSERT INTO T VALUES(1, TRUE ,FALSE);

What will the output of the following SELECT command?

SELECT \* FROM TEMP;

1. 1, TRUE, FALSE
2. **1, 1, 0**
3. 1, 0, 1
4. 1, 1, 1

Q184. Which of the following SQL statement you will issue to display all employee details whose department number and password are same.

1. SELECT emp.\* from EMP, DEPT where emp.deptno=dept.deptno and emp.pwd= dept.pwd;
2. SELECT \* from EMP WHERE (deptno, pwd) = (SELECT deptno, pwd from DEPT WHERE deptno=30);
3. SELECT \* from EMP WHERE ROW (deptno, pwd) = (SELECT deptno, pwd from DEPT WHERE deptno=30);
4. **All of the above.**

Q185. Query to display the last day of the month three months before the current month.

1. SELECT last\_day(subdate(now(), interval 3 month));
2. SELECT last\_day(subdate(now(), interval -3 month));
3. SELECT last\_day(date\_sub(now(), interval 3 month));
4. **Both A and C**

Q186. Write a query to get the current date in the following format (Sunday July 2017).

1. **SELECT DATE\_FORMAT(NOW(), '%W %M %Y');**
2. SELECT FORMAT\_DATE(NOW(), '%W %M %Y');
3. SELECT DATE\_FORMAT(NOW(), '%W' '%M' '%Y');
4. SELECT FORMAT(NOW(), '%W %M %Y');

Q187. Write a query to extract the year from the current date.

1. **SELECT EXTRACT(YEAR FROM NOW());**
2. SELECT EXTRACT(FROM NOW() YEAR);
3. SELECT EXTRACT(YEAR FROM NOW);
4. SELECT EXTRACT(FROM NOW YEAR);

Q188. What is the result of select strcmp(substr("Infoway",1,4),left("Infoway",4));

1. **0**
2. 1
3. -1
4. null

Q189. The user has executed the following code.

Line 1 DELIMITER //

Line 2 CREATE TRIGGER contacts\_after\_insert AFTER INSERT ON contacts FOR EACH ROW

Line 3 BEGIN

Line 4 DECLARE vUser varchar(50);

Line 5

Line 6 -- Find username of person performing the INSERT into table

Line 7 SELECT USER() INTO vUser;

Line 8

Line 9 -- Insert record into audit table

Line 10 INSERT INTO contacts\_audit (contact\_id, deleted\_date, deleted\_by) VALUES (NEW.contact\_id, SYSDATE(), vUser);

Line 11 END; //

Line 12 DELIMITER;

Do you find any error in the trigger?

1. Error at Line 1
2. Error at Line 4
3. Error at Line 11
4. **No Errors**

Q190. Use has issued the following code.

drop trigger if exists T1;

DELIMITER //

CREATE TRIGGER T1 AFTER INSERT ON DEPT FOR EACH ROW

BEGIN

set new.deptno = 1001;

END; //

DELIMITER ;

1. The trigger will execute properly.
2. AFTER INSERT trigger type is invalid.
3. **You cannot change the NEW bind variable value in AFTER trigger type.**
4. Invalid variable name.

Q191. What will be the output of the following statement?

SELECT round(1234, -3)

1. 1234
2. **1000**
3. 1200
4. Null

Q192. What will be the output of the following statement?

SELECT round(1254, -3)

1. 1234
2. **1000**
3. 1200
4. Null

Q193. What will be the output of the following statement?

SELECT round(1554, -3)

1. 1234
2. 1000
3. **2000**
4. Null

Q194. What will be the output of the following statement?

SELECT truncate(1234, -2);

1. 1234
2. **1200**
3. 2000
4. Null

Q195. What will be the output of the following statement?

SELECT truncate(1235, -2);

1. 1234
2. **1200**
3. 2000
4. Null

Q196. What will be the output of the following statement?

SELECT truncate(1265, -2);

1. 1234
2. **1200**
3. 2000
4. Null

Q197. Do you find an error in this code?

DELIMITER $$

CREATE TRIGGER before\_employee\_update BEFORE UPDATE ON employees FOR EACH ROW

BEGIN

INSERT INTO employees\_audit SET action = 'update', employeeNumber = OLD.employeeNumber, lastname = OLD.lastname, changedate = NOW();

END$$

DELIMITER ;

1. Yes, you cannot use BEFORE with UPDATE DML command
2. Yes, you cannot use FOR EACH ROW with BEFORE trigger type;
3. Yes, invalid INSERT command syntax;
4. **No Error.**

Q198. The user executer the following code.

drop trigger if exists T1;

DELIMITER //

CREATE TRIGGER T1 BEFORE INSERT or UPDATE ON DEPT FOR EACH ROW

BEGIN

declare x int;

SET x:=1001;

END; //

DELIMITER ;

1. **Error, because one trigger can have only one triggering event.**
2. Error, invalid declaration of variable.
3. Error, invalid assignment of value to the variable.
4. No Error.

Q199. List the details of the emps whose salaries more than the employee 'BLAKE'

1. SELECT A.\* from EMP A, EMP B where A.sal > B.sal and B.ename = 'BLAKE';
2. SELECT \* from EMP where sal > (SELECT sal from EMP where ename = 'BLAKE');
3. **Botha A and B**
4. None of the above.

Q200. List the details of the emps whose job is same as ALLEN.

1. SELECT \* from EMP where job = (SELECT job from EMP where ename ='ALLEN');
2. SELECT \* from EMP where job in (SELECT job from EMP where ename ='ALLEN');
3. SELECT \* from EMP where job like (SELECT job from EMP where ename ='ALLEN');
4. **All of the above**

Q201. In MySQL, to remove duplicate rows from the result set of a SELECT use the following keyword:

1. NO DUPLICATE
2. UNIQUE
3. **DISTINCT**
4. None of the above

Q202. In a LIKE clause, you can could ask for any value ending in "qpt" by writing

1. **LIKE %qpt**
2. LIKE \*qpt
3. LIKE qpt$
4. LIKE ^.\*qpt$

Q203. Which of the following clause is used in the CREATE TRIGGER command to create a row level trigger?

1. FOR ROW
2. ROW LEVEL
3. ROW LEVEL TRIGGER
4. **None of the above.**

Q204.

Q205.

Q206.

Q207.

Q208.

Q209.

Q210.

Q211.

Q212.

Q213.

Q214.

Q215.

Q216.

Q217.

Q218.

Q219.

Q220.