***MySQL***

**1.Database:**A database is an organised collection of data.

Software used to manage databases is called Data Base Management System (DBMS).

2. **Relational Database:**A database in which the data is stored in the form of relations

(also called tables) is called a Relational Database. In other words a Relational

Database is a collection of one or more tables.

3. **RDBMS:**A DBMS used to manage Relational Databases is called an RDBMS (Relational Data Base Management System). Some popular RDBMS software available are: Oracle, MySQL, Sybase, Ingress

P**- REVISION TOUR**

4. **Benefits of using a DBMS are:**

a. Redundancy can be controlled

b. Inconsistence can be avoided

c. Data can be shared

d. Security restrictions can be applied.

5. **MySQL:**It is an Open Source RDBMS Software. It is available free of cost.

6. **Relation/Table:**A table refers to a two dimensional representation of data arranged

in columns (also called fields or attributes) and rows (also called records or tuples). E

7. **Key:**A column or a combination of columns which can be used to identify one or

more rows (tuples) in a table is called a key of the table.

8. **Primary Key:**The group of one or more columns used to uniquely identify each row

of a relation is called its Primary Key.

9. **Candidate Key:**A column or a group of columns which can be used as the primary

key of a relation is called a candidate key because it is one of the candidates

available to be the primary key of the relation

10. **Alternate Key:**A candidate key of a table which is not made its primary key is called its Alternate Key. H

**11.Degree**of a table is the number of columns in the table.

12.**Cardinality** of a table is the number of rows in a table.

13. **SQL (Structured Query Language):**It is the language used to manipulate and

manage databases and tables within them using an RDBMS.

14. **DDL (Data Definition Language):**This is a category of SQL commands. All the

commands which are used to create, destroy, or restructure databases and tables

come under this category. Examples of DDL commands are - CREATE, DROP, ALTER.

15. **DML (Data Manipulation Language):**This is a category of SQL commands. All the

commands which are used to manipulate data within tables come under this

category. Examples of DML commands are - INSERT, UPDATE, DELETE.

16. **DCL (Data Control Language):**This is a category of SQL commands. All the

commands which are used to control the access to databases and tables fall under

this category. Examples of DCL commands are - GRANT, REVOKE.

17.Commands

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1.    Create database | CREATE DATABASE <databasename>; | DDL COMMAND |
| 2.    OPEN DATABSE | USE <databasename>; | DML |
| 3. To show the name of the current database | SELECT DATABASE(); | DML |
| 4. To show a list of tables present in the current  database. | SHOW TABLES; | DML |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1.    To create a table | CREATE TABLE <tablename>  (<column name1> <data type1> | DDL |
| 2.    To show the structure of a table. | DESCRIBE <tablename>; or  DESC <tablename>; | DML |
| 3.    To add a new column to a table | ALTER TABLE <tablename>  ADD <columnname> <datatype>; | DDL |
| 4. To modify a column in a table | ALTER TABLE <tablename>  MODIFY <column> <new\_definition>; | DDL |
| 5. To delete a column from a table | ALTER TABLE <tablename>  DROP <columnname>; | DDL |
| 6. To insert a new record onto the table | INSERT INTO <tablename>  [<column1>, <column2>, ..., <columnn>]  VALUES (<value1>, <value2>, ... <value n>); | DML |
| 7. To change the data present in the table | UPDATE <tablename>  SET <column name> = <value>  [,<column name> = <value>, …]  [WHERE <condn>]; | DML |
| 8. To delete data from a table | DELETE FROM < tablename>  [ Where < condition>]; | DML |

Following are the clauses which can be used with SELECT command:

1. ***DISTINCT*** Used to display distinct values from a column of a table.

1. ***WHERE*** Used to specify the condition based on which rows of a table are displayed.

1. ***BETWEEN*** Used to define the range of values within which the column values must fall to make a condition true. Range includes both the upper and the lower values.

1. ***IN*** Used to select values that match any value in a list of Specified values

1. ***LIKE*** Used for pattern matching of string data using wildcard characters **%**and **\_**

1. ***IS NULL***  Used to select rows in which the specified column is NULL (or is NOT NULL NOT NULL)
2. ***ORDER BY*** Used to display the selected rows in ascending or in descending order of the specified column/expression.

**Functions in MySQL**

**Numeric Functions:**

1.POWER(*x*,*y*) Returns the value of *x*raised to the power of *y*.

**OR**

POW(*x*,*y*)

2. ROUND(*x*) Rounds the argument *x*to the nearest INTEGER.

3. ROUND(*x,d*) Rounds the argument *x*to d decimal places.

4. TRUNCATE(*x,d*) Truncates the argument *x*to d decimal places.

**String Functions:**

1. **LENGTH**(str) Returns the length of a column or a string in bytes.

2. **CONCAT**(str1,str2,...) Returns the string that results from concatenating the arguments. May have one or more arguments.

3. **INSTR**(str,substr) Returns the position of the first occurrence of substring <substr> in the string <str>.

4. **LOWER**(str) Returns the argument <str> in lowercase. i.e., It **or**LCASE(str) changes all the characters of the passed string to lowercase.

5. **UPPER**(str) Returns the argument <str> in uppercase. i.e., It **or**UCASE(str) changes all the characters of the passed string to uppercase.

6. **LEFT**(str,n) Returns the first <n> characters from the string <str>

7. **RIGHT**(str,n) Returns the last <n> characters from the string <str>

8. **LTRIM**(str) Removes leading spaces, i.e., removes spaces from the left side of the string <str>.

9. **RTRIM**(str) Removes trailing spaces, i.e., removes spaces from the right side of the string <str>.

10. **TRIM**(str) Removes both leading and trailing spaces from the string <str>.

th 11. **SUBSTRING**(str,m,n) Returns <n> characters starting from the m **OR**character of the string <str>.

**SUBSTR**(str, m, n) If the third argument <n> is missing, then starting th **OR**from m position, the rest of the string is returned.

**MID**(str,m,n) If <m> is negative, the beginning of the substring is th the m character from the end of the string.

12**. ASCII**(str) Returns the ASCII value of the first character of the string <str>. Returns 0 if <str> is the empty string. Returns NULL if <str> is NULL.

**Date and Time Functions:**

1 **CURDATE**() Returns the current date in YYYY-MM-DD format or YYYYMMDD format, depending on whether the function is used in a string or numeric context.

2 **NOW**() Returns the current date and time in 'YYYY-MM-DD HH:MM:SS' or YYYYMMDDHHMMSS.uuuuuu format, depending on whether the function is used in a string or numeric context.

3 **SYSDATE**() Returns the current date and time in 'YYYY-MM-DD HH:MM:SS' or YYYYMMDDHHMMSS.uuuuuu format, depending on whether the function is used in a string or numeric context.

4 **DATE**(expr) Extracts the date part of a date or datetime expression <expr>.

5 **MONTH**(date) Returns the numeric month from the specified date, in the range 0 to 12. It returns 0 for dates such as '0000-00-00' or '2010-00-00' that have a zero month part.

6 **YEAR**(date) Returns the year for specified date in the range 0 to 9999. It returns 0 for the "zero" date. Returns values like 1998, 2010,1996 etc.

7 **DAYNAME**(date) It returns the name of the weekday for the specified date.

8 **DAYOFMONTH**(date) Returns the day of the month in the range 0 to 31.

9 **DAYOFWEEK**(date) Returns the day of week in number as 1 for Sunday, 2 for Monday and so on.

10 **DAYOFYEAR**(date) Return the day of the year for the given date in numeric format in the range 1 to 366.