**DATABASE:**

It is a collection of logically related information about a specific topic. Ex: Student database.

ADVANTAGES:

* Removes data duplicasy to a large extent.
* It removes data inconsistency to a large.
* It provides sharing and security.
* It provides integrity constraints.
* Database must be standerised.

**DBMS:**

Database Management System: It is a system which provides storing, modifying, relating as well accessing the data from the database.

Rational DBMS: It is a system in which information is stored in tabular form i.e. rows and columns. If the data is stored in multiple tables then it is related by common column.

Example: 1.MySQL 2.INGRES 3.ORACLE 4. SYBASE

**Table/Relation:**

It is a two dimension representation of data i.e. collection of rows and columns. While creating table we must keep the following things in mind:-

1. Data must be in atomic form.
2. No two rows as well as no two columns must be identical.
3. The way of storing data may be in any order.

**INTEGRITY CONSTRAINTS:**

These are the restrictions which are applied by programmer so the user will not be able to enter invalid data.

Ex: Grade is either A or B or C.

**PRIMARY KEY:**

It is a key which can never be null and can never be duplicate. All the columns of table depend on primary.

Ex:

Admission number of student in school.

**CANDIDATE KEY/ COMPOSITE KEY**

It is a key or a set of keys to uniquely identify the record/truple in the relation. It can ac as primary if the primary key is not known.

Ex: [class+section+rollno] -> candidate key

**ALTERNATE KEY:**

All the keys(columns) other than primary key in the table known as alternate key. Ex: Name and Marks.

**FOREIGN KEY/ REFERENCE KEY/ DERIVED KEY /RELATIONSHIP KEY**

It is key which is derived from another table where it acts as primary key. It is used to set the relationship b/w the tables.

**UNIQUE KEY:**

It can never be duplicate but can be NULL. Ex: phone no.

NOT NULL: It must be defined as we cannot leave it empty. Ex: Name

**DEFAULT:**

It specify a value which is used if no value will be provided at the time of storing the data. Ex: Medical stream.

|  |  |
| --- | --- |
| DEGREE | CARDINALITY |
| Total number of columns in a table is known as | Total number of record in a table is known as |
| Ex: Student table has 3 columns | Ex: Student table 2 records. |

**MY SQL**

+ Structured Query Language [SQL]

-/ It is a language which can create and operate on relational databases. It is case insensitive language.

-/ It is free and open source relational database.

-/ It is easy to learn and use.

-/ It contains a set of commands which are divided in following categories:

|  |  |
| --- | --- |
| DDL-Data Defination Language | DML- Data Manipulation Language |
| It contains commands which are responsible for defining the memory area. | It contains the commands to store the data, modify the data or to remove the data from the database. |
| Ex: Create table, alter table, drop table | Ex: insert update, delete |

Q1. How to display all the databases in your system ?

* show databases;

Q2. How to create a database.

* create database database\_name;

Q3. How to open the database ?

* use database\_name;

Q4. Display name of all tables in your current databasr.

* Show tables;

**DATATYPES OF MYSQL:**

It specify the type of data and its range of any memory location.

1. char/varchar
2. int / integer
3. decimal /float
4. Date

Q5. How to create table.

* Create table tablename (column name datatype(size) constraint, …);

Ex: create table student (

Rollno int (2) Primary Key,

Name varchar(20) NOT NULL,

Marks float (4,1)

Dob date);

Q6. How to display the structure of table.

* Desc/describe table\_name;

Q7. Write command to display the structure of student table.

* Desc student;

**DISTINCT :** ‘DISTINCT’ clause is used to eliminate duplicate olumns from specified columns while displaying.

Q8. Display name of all students from student table but name should not be repeated.

* Select distinct name from student;

|  |
| --- |
| STUDENT TABLE |
| Raj |
| Geeta |
| Priya |
| Raj |
| Priya |

OUTPUT-:

|  |
| --- |
| NAME |
| RAJ |
| GEETA |
| PRIYA |

**OPERATORS:-** These are the symbols which are used to do certain operation on the given operants. Ex: 5+2 -> 5,2 operands +:operator

These are of following types:-

* Arithmatic or Mathematical operators: Ex: +,/,\*,-,%
* Relational or comparison operators: >,=,<,>=,<=,<>-Not equal
* Logical operator: combine to or more operator. Ex: AND,OR,NOT

**WHERE-** It is used to put a condition on each row table.

Ex: select name,dob from student where marks>=60;

**BETWEEN OPERATOR:-** It is used to provide a range of values to compare but include the boundary values also. It is an alternative of ‘AND’ Operator.

Ex: column\_name between value1 and value2;

Q9. Display the detail of all students whose marks are in between 91 and 97.

* select \* from student where marks between 91 and 97;

or

select \* from student where marks>=91 and marks<=97;

**IN OPERATOR:** It provides a list of values and any one value can be selected from their. It is an alternative of ‘OR’ operator. Ex: column\_name in (values);

**LIKE OPERATOR:** It is used to provide the pattern matching of all the stings with the help of wild card characters. Syntax:

% -> used to compare any number of letters.

* -> to compare single character.

‘R%’ name starts with letter R

‘%R’ name ends with R

‘R%R’ name ends with R

‘%R%’ name must contain R

‘- - R - - P %’ 3rd letter must be R and 6th letter must be P

**ORDER BY:-** It is used to arrange the data in an order, i.e. ascending or descending. By default ascdnding.

ASC- ascending

DESC – descending

**FUNCTION IN MY SQL :**

Or Aggregate/Group/Multiple row function

* SUM()
* AVG()
* MAX()
* MIN()
* Count(\*): to return total number of records in the table which are NULL or NOT NULL.
* Count(column\_name):- to return total number of rowswhich are not null.

**UPDATE:** To change the data stored in the table.

Syntax:

Update table\_name set, columnname=value, where condition; set,-> Keyword

**DELETE:** To remove the data from the table.

Syntax: Delete from table name where condition;

Q10. Remove the detail of all employee who joined the organization in year 1981.

* Delete from employee where hiredata like ‘1981%’ or where hiredate between ‘1981-01-01’ and ‘1981-12-31’ or where hiredate >=’1981-02-01’ and hiredate <= ‘1981-18-31’;

**ALTER TABLE:** Alter table command is used to change the structure i.e. adding columns, removing, columns, modifying datatype, modifying size or the constraints.

* **Adding columns:** Alter table tablename **Add**  columnname datatype(size) constraint,…;
* **Remove Columns:** Alter table tablename **Drop** columnname ;
* **Changing the datatype or Size of any columns:** Alter table tablename **Modify** columnname datatype(size) constraints .. ;

**DROP:** This command is used to remove the table or database permanently from memory , i.e. data as well as structure. Ex: DROP table emp;

|  |  |
| --- | --- |
| WHERE | HAVING |
| It is used to put a condition based on each row of the table. | It is used to put a condition based on specific group defined by group by clause. |

**DOMAIN:** It is a pool of values fro which the actual values appearing in a given column are drawn.

**KEY:** It is a column to access the data.

|  |  |
| --- | --- |
| Selection | Projection |
| It is horizontal subset of relation. | It is vertical subset of relation. |
| Representation by “Sigma”. | Represented by “Pie”. |

**JOIN:**  Extraction of data from multiple tables. It is of following types:

1. Equi Join
2. Natural Join
3. Non- Equi Join
4. Self Join

**UNION:**  Extraction of data from multiple tables but the duplicate data will appear once. We can apply union only if the following conditions are true:

* Degree of both tables are equal :Ith column of 1 table must be equal to Ith  column of other table.

**CARTESIAN PRODUCT:** It is represented by ‘Cross’ Symbol. It will prodce a new table by comparing each row of one table with each row of one table with each row of other. So the resultant table will get product of cardinality and sum of degrees in the resultant table.

Ex:

EMP

|  |  |  |  |
| --- | --- | --- | --- |
| EMPNO. | ENAME | SALARY | DEPTNO. |
| 1 | RAJ | 5000 | 1 |
| 2 | GEETA | 7000 | 2 |
| 3 | SITA | 6000 | 1 |

DEPT.

|  |  |
| --- | --- |
| DEPTNO. | DNAME |
| 1 | SALES |
| 2 | FINANCE |

Command: Select \* from EMP,DEPT;

**Cartesian product:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EMPNO | ENAME | SALARY | DEPTNO | DEPTNO | DNAME |
| 1 | Raj | 5000 | 1 | 1 | Sales |
| 1 | Raj | 5000 | 1 | 2 | Finance |
| 2 | Geeta | 7000 | 2 | 1 | Sales |
| 2 | Geeta | 7000 | 2 | 2 | Finance |
| 3 | Sita | 6000 | 1 | 1 | Sales |
| 3 | Sita | 6000 | 1 | 2 | Finance |