MySQL

Concepts to be discussed:

- MySQL features/Advantages
- Types of MySQL commands
- Data Types
- Use of Create command
- Insert command different ways
- Display information
- Show selective information
- Checking empty values
- Unique information
- Use of ALL keyword
- Show date and time
- Inserting data into another table
- Delete a particular record
- Change in particular record
- Add new column
- Change column name
- Displaying information alphabetically
- Display same type of information together
- Differences

SQL is an acronym of Structured Query language

It is standard language developed and used for accessing and modifying relational databases.

SQL is being used by many database management systems. Some of them are:

- MySQL
- PostgreSQL
- Oracle
- SQLite
- Microsoft SQL Server

MySQL

- is open source, multi user, multi threaded database management system.
- MySQL is especially popular on the web
- It is oneof the parts of the very popular LAMP platform or WIMP platform.
- LAMP(Linux, Apache, MySQL and PHP) or WIMP(Windows, Apache, MySQL and PHP)

MySQL Features

Open Source & Free of Cost:
It is Open Source and available at free of cost.

- Portability:- Small enough in size to install and run it on any types of Hardware and OS like Linux,MS Windows or Mac etc.
- Security :- Its Databases are secured & protected with password.
- Connectivity:- Various APIs are developed to connect   it with many programming languages.

MySQL Features contd…

❑ Query Language

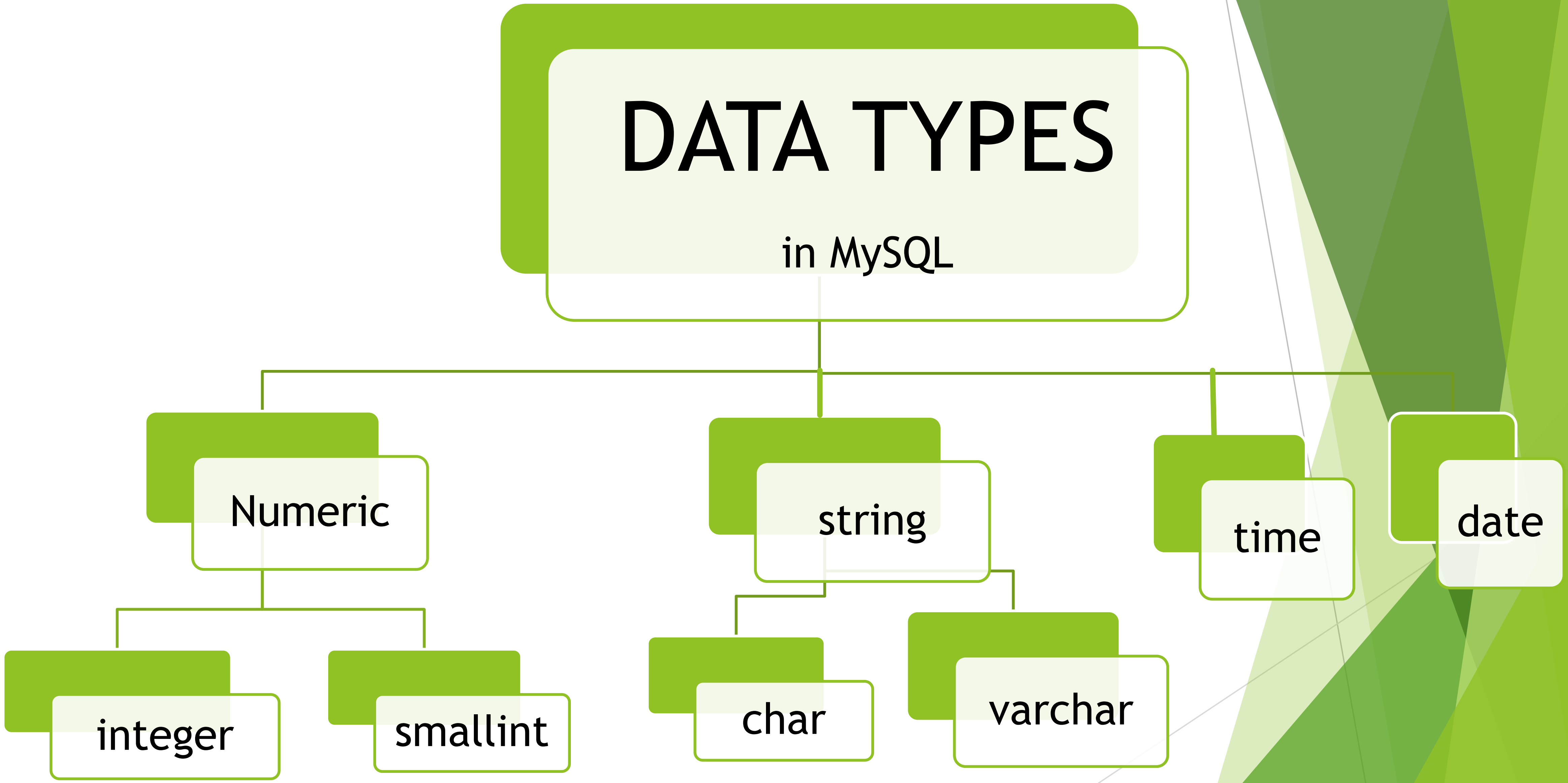
It supports SQL (Structured Query Language) for handling database.

- Interactive Language-This language can be used for communicating with the databases and receive answers to the complex questions in seconds.
- Multiple data views-The users can make different views of database structure and databases for the different users.
- No coding needed-It is very easy to manage the database systems without any need to write the substantial amount of code by using the standard SQL.
- Well defined standards- Long established are used by the SQL databases that is being used by ISO and ANSI.

Types of SQL Commands

- DDL (Data Definition Language)
  - To create database and table structure-commands like CREATE , ALTER , DROP etc.
- DML (Data Manipulation Language) Record/rows related operations.commands like SELECT, INSERT, DELETE, UPDATE etc.
- DCL (Data Control Language)
  - used to manipulate permissions or access rights to the tables. commands like GRANT , REVOKE etc.
- Transactional control Language.

  Used to control the transactions.commands like COMMIT, ROLLBACK, SAVEPOINT etc.


DATA TYPES
in MySQL
Numeric
string
time
date
integer
smallint
char
varchar

Data type in MySQL

- **Numeric Data Types:**
  - **INTEGER or INT** – up to 11 digit number without decimal.
  - **SMALLINT** – up to 5 digit number without decimal.
  - **FLOAT (M,D) or DECIMAL(M,D) or NUMERIC(M,D)**

    Stores Real numbers upto **M** digit length (including .) with **D** decimal places.

    e.g. Float (10,2) can store 1234567.89
- **Date & Time Data Types:**
  - **DATE** - Stores date in YYYY-MM-DD format.
  - **TIME** - Stores time in HH:MM:SS format.
- **String or Text Data Type:**
  - **CHAR(Size)**

    A fixed length string up to 255 characters. (default is 1)
  - **VARCHAR(Size)**

    A variable length string up to 255 characters.

Char, **Varchar**, **Date** and **Time** values should be enclosed with single (‘ ’) or double (“”) quotes in MySQL. varchar is used in MySQL and varchar2 is used in Oracle.

Create Database

- syntax:

```
create database database_name;
```

e.g.

```
create database db1;
```

USE Database

USE command to open the database

- syntax:

```
use database_name;
```

- e.g.

```
use db1;
```

Show command- to display all databases/tables

```
show databases;
or
show tables;
```

Create Table

- syntax:

```
create table table_name    (column_name1 data_type(size) constraint ,
                             column_name2 data_type(size) constraint,
                             column_name3 data_type(size) constraint
                             :
                    :
                     );
```

e.g.

```
create table emp (empid integer, name char(30), salary decimal(7,2),
                            designation char (30)
                      );
```

describe or desc command-
to see the structure of a table

- syntax:

```
describe table_name;
```

or

```
desc table_name;
```

e.g.

```
describe emp;
```

or

```
desc emp;
```

insert command-
to insert record into the table

- syntax1: to insert a record in all the columns

insert into table_name values (value1, value2, value3 ………..);

e.g.

insert into emp values(1,’teena’,45678,’manager’);

- syntax2: to insert multiple records in all the columns

insert into table_name values (value1, value2, value3 ………..), (value1, value2, value3 ………..);

e.g.

insert into emp values(3,’leena’,43000,’clerk’),(4,’meena’,47000,’analyst’);

insert command-
to insert record into the table

- syntax3: to insert records in to specific columns

insert into table_name(column_name1,column_name2) values (value1, value2);

e.g.

insert into emp(name, empid) values(‘sheena’,2);

- syntax4: inserting null values

insert into table_name values (value1, value2, value3 ………..);

e.g.

insert into emp values(3,’heena’,50000,NULL);

Columns that are not listed in the insert command will have their default value if it is defined, otherwise take null value

select command-
to display/print records of the table

- syntax1: display all columns of the table

```
select * from table_name;
```

e.g.

```
select * from emp;
```

- syntax2: display table with specific/particular column(s)
- select column_name1,column_name2….. from table_name;

e.g.

```
select name,designation,empid from emp;
```

select command-

To Perform Simple Calculations using Select Query

- syntax1:

select value1 operator value2;

e.g.

```
select 2*3;
```

- syntax2: to do calculation in particular column on temporary basis

```
SELECT empno, empname, sal+3000  from emp;
```

where command-is used to display/print records of the table with condition

- syntax1: display all columns with condition from the table

select * from table_name where column_name=value;

e.g.

select * from emp where name="teena";

- syntax2: display table with specific/particular column(s)
- select column_name1,column_name2….. from table_name where column_name=value;

e.g.

select name,designation,empid from emp where name="teena";

where command-

relational operator- >,< ,<= ,>= ,=(equal to),<>(not equal to)

- display the employee information whose name is not teena from the table emp

```
select * from table_name where column_name relational operator value;
```

e.g.

```
select * from emp where name<>“teena”;
```

- to display name and designation of those employees ,from the table emp, whose salary is more than 48000.

```
select name,designation from emp where salary>48000;
```

where command-
logical operator- and, or , not

- display the employee information whose salary is more than 48000 and name is not teena from the table emp

select * from emp where salary >48000 and name<>“teena”;

- to display name and designation of those employees ,from the table emp, whose salary is more than 48000 or designation is clerk.

select name,designation from emp where salary>48000 or designation =‘clerk’;

where command-
between …and keyword- condition based on a range

- syntax:

select */ column_name from table where column_name between value1 and value2;

e.g.

1. display the employee information whose salary is in the range of 45000 to 55000 from the table emp

```
select * from emp where salary between 45000 and  55000;
```

2. display the employee information whose salary is not between 45000 to 55000 from the table emp

```
select * from emp where salary not between 45000 and 55000;
```

where command-
in keyword- condition based on a list

- syntax:

select */ column_name from table where column_name in (value1, value2 …);

e.g.

1. display only manager and clerk employee information from the table emp

select * from emp where designation in(‘manager’, ‘clerk’);

OR

select * from emp where designation =‘manager’ or designation =‘clerk’;

2. display all designation except manager and clerk from the table emp

select * from emp where designation not in(‘manager’, ‘clerk’);

OR

select * from emp where designation<>’manager’ or designation <>‘clerk’;

where command-
Like operator- condition based on pattern matches

- 1. percent(%)- to match any substring
- syntax:

select */ column_name from table where column_name like ‘ % ‘;

e.g.

(i). To list those employee information whose name is starting from ‘m’ from the table emp

select * from emp where name like ‘m%’;

(ii). display employee information whose name is ending with ‘a’ from the table emp

select * from emp where name like ‘%a’;

where command-
Like operator- condition based on pattern matches

- 2. underscore(_)- to match any single character
- syntax:

select */ column_name from table where column_name like ‘ __ ‘;

e.g.

(i). To list those employee information whose name’s length is 5 from the table emp

select * from emp where name like ‘_ _ _ _ _’;

(ii). display employee information whose name is 5 character long and second character must be ‘e’ from the table emp

select * from emp where name like ‘_e_ _ _’;

(iii) display employee information whose name’s second character must be ‘e’ from the table emp

select * from emp where name like ‘_e%’;

where command-
NULL operator- searching for NULL

- syntax:

select */ column_name from table where column_name is NULL;

e.g.

(i). To list those employee information whose designation is not filled.

```
select * from emp where designation is NULL;
```

(ii) display employee information where designation is not empty.

```
select * from emp where designation is not NULL;
```

Distinct keyword- show only unique values

- syntax:

```
select distinct  column_name from table;
```

e.g.

(i). To list the different types of designation from the table emp

```
select distinct designation from emp;
```

ALL keyword- show all values (retains duplicates values)

- syntax:

select all column_name from table;

or

select * from table;

e.g.

(i). To list the designation from the table emp

```
select all designation from emp;
```

Current date and time

select curdate(); - to show system current date

select curtime(); - to show system current time

Insert data into another table

syntax

```
insert into newtable_name select  * from table_name where condition ;
```

Add those employee information into table emp2 where salary is less than 50000;

e.g.

```
insert into emp2 select * from emp where salary<50000;
```

UPDATE-
modify/change data in a table

syntax

update tablename set column_name=value where condition

e.g.

(i) Increase the salary of all employee by 200;

```
update emp set salary= salary +200;
```

(ii)Decrease the salary of all employee by 2%;

```
update emp set salary= salary –salary *0.02;
```

(iii) Increase the salary of those employee by 200 whose salary is less than 40000;

```
update emp set salary= salary +200 where salary <40000;
```

DELETE command
- to Delete a record/row from the table

syntax

delete from table_name where condition

e.g.

(i) Delete all data from the emp table;

```
delete from emp;
```

(ii)Delete those information whose designation is analyst;

```
delete from emp where designation =‘analyst’;
```

(iii) Increase the salary of those employee by 200 whose salary is less than 40000;

```
update emp set salary= salary +200 where salary <40000;
```

DROP – to Delete the table

syntax

```
drop table if exists table_name;
            OR
drop table table_name;
```

e.g.

(i) To delete the table emp;

```
drop table emp;
```

ALTER– to add/modify the column

syntax

1. To add a new column in to existing table

alter table table_name add (column_name datatype(size));

e.g.

Alter table emp add(phone integer(10));

(2) To modify the size of a particular column into existing table;

Alter table emp modify phone integer(11);

ALTER- to change the name of the column

syntax

1. alter table table_name change old_column_name new column_name datatype(size);

   (version 5.x)

e.g.

```
alter table emp change name empname varchar(30);
```

2. alter table table_name rename column old_column_name to new_column_name;

   (version 8.x)

```
alter table emp rename column name to empname;
```

ORDER BY- to sort the result in a particular order ascending/descending

syntax

1. select * /columm_names

   from table name

   where condition

   order by column_name asc/desc

* asc for ascending. desc for descending. sDefault is ascending order.

Q : To display employee's name in descending order

select empname from emp order by empname desc;

Q : To display employee information in descending order of their name where salary is more than 5000

select * from emp where sal > 5000 order by empname desc;

Aggregrate Functions-These functions return a single value after calculating from a group of values.

frequently used Aggregrate functions.

- avg(),
- sum(),
- max(),
- min(),
- count(column_name)-Count returns the number of rows present in the table either based on some condition or without condition.
- count(distinct)
- SELECT COUNT(distinct salary) from emp;

GROUP BY- is used to group the results of a SELECT query based on one or more columns. . It is also used with SQL aggregate functions to group the result

syntax

```
1.    select * /columm_name(s)
      from table name
      where condition
      group by column_name
```

** Group By clause will always come at the end.*

Q : To show name and sum of the salary of employees according to their designation

- SELECT name,sum(sal) from Emp group by designation;

HAVING- It is used to give more precise condition for a statement. It is used to mention condition in Group based SQL functions, just like WHERE clause.

syntax

1.
```
select * /columm_name(s)
from table name
where condition
group by column_name
having condition
```

Q : To show name and sum of the salary of employees of whose designation count is more than 2

```
SELECT name,sum(sal) from Emp group by designation having count(*) >2;
```

Q : To show name and sum of the salary of employees whose designation is clerk

```
SELECT name,sum(sal) from Emp group by designation having designation=‘clerk’;
```

SQL Alias- Alias is used to give an another name to a table or a column.

- **Syntax: Alias name to table**

SELECT column-name from table-name table_alias-name;

Example

- SELECT * from Employee_detail ed;

- **Syntax: Alias name to column**

SELECT column-name “alias-name “ from table-name;

- SELECT customer_id “cid” from Emp;

SQL View- A view in SQL is a logical subset of data from one or more tables.
View is used to restrict data access.

- Syntax:

```
CREATE  view view_name AS
SELECT column_name(s) FROM table_name WHERE condition
```

- Example

Write a command that will store the result in sale_view from table “emp” where employee name is Alex

```
CREATE  view sale_view as select * from emp where empname= 'Alex';
```

Displaying View-displaying a view is similar to fetching data from table using Select statement..

- Syntax:

```
SELECT */column_name(s) FROM view_name where condition;
```

- Example

```
SELECT * from sale_view;
```

Update a view- Update command for view is same as for tables.

- Syntax:

```
UPDATE view-name
set value
WHERE condition;
```

******If we update a view it also updates base table data automatically.***

Delete a view- delete command for view is same as for tables.

- Syntax:

```
Drop view viewname;
```

- Example

```
Drop view sale_view;
```

Differences

DDL	DML
Data definition language	Data manipulation language
Create	Insert
Alter	Update
Drop	Delete Select

ALTER	UPDATE
DDL command	DML command
It is used to add /delete/change name of a particular column	It is used to change/modify the value(s) in particular column(s)
alter table tablename add/modify/change column_name datatype(size)	update tablename set column_name=value where column_name=value

DROP	DELETE
DDL command	DML command
It is used to delete the table or database permanently	It is used to delete a particular record(s) from the table
drop table table_name	delete from table_name [where condition]

WHERE	HAVING
Where- Where clause is used to specify condition on single row.	having- It is used to mention condition in Group
Where clause is used mostly with Select, Update and Delete command/query	Having clause is used only with group by clause

ORDER BY	GROUP BY
It is used to arrange records in a particular order(asc/desc)	It is used to group together similar types of information
select */column_name from table_name order by column_name asc/desc	**Select */column_name from table_name group by column_name**

DROP TABLE	DROP VIEW
This command will delete the table permanently,	This command will delete the view permanently
if any view is created from this table then view will not be deleted	**By deleting view there will be no effect on table**
Drop table table_name	**Drop view view_name**

CREATE TABLE	CREATE VIEW
Table will be created by using create command	This command will derive the data from one or more base tables
if any view is created from this table then view will not be deleted	**to create this , no need to create table**
A table is structured with columns and rows	while a view is a virtual table extracted from a database/table.
A table consists of rows and columns to store and organized data in a structured format	view is a result set of SQL statements