MySQL -Datatypes

Char

- * allows any character.
- * could be alphanumeric also
- * max 255 characters * fixed length.
- * wastage of Hard isk space, * search, retrive fast

Varchar.

- * allows any Character
- * could be alpha-numeric also
- * max, 65, 535 character (64 Kb-1)
- * variable length
- * conserve (fixed size) on HD space.
- * search retrieval ~~and~~ searching slow.

char (20 B) varchar (20B)

$\underbrace{\text{Suraj. ƀƀƀ ƀƀƀƀƀƀƀƀ}}_{\because 20B}$ $\underbrace{\text{Suraj}}_{4.B.}$

→* Longtext

- * allows any character
- * could be alpha-numeric also
- * Max 4,294,967,295 characters
- * 4Gb -1

e.g:- feedback, comments, Remark, post, Resume, Experience etc.

- * Stored outside the table.
- * Stored outside the row
- * stored away from the table.
- * Stored away from the row
- * MySQL maintains a locator (HD pointer) from the longest column to the longest text.

* performance of other

☆ * Binary

* fixed length binary string
* max upto 255 bytes of binary data
* eg.- Barcodes, QR-codes, Picture codes, Sign. GIFs,

☆ Varbinary.

* variable length binary string.
* max upto 65,535 bytes of binary data
* eg :- Icons, GIFS with sound, THUMBNAIL, FAVICON.

☆ Blog :-> Binary large object

☆ Longblob

* max upto 4,294,967,295 Bytes of Binary data
* 4Gb -1
* eg :- Photograph, Charts, Maps, music, Sound, Video etc.
* width does not have to be Specified.
* eg :- Photograph longblob.
* Stored outside the table | row

* Mysql maintains a Locator R(HD pointer) from the longblob column to the longblob data

* Integer types (Exact values) :-
* Signed or unsigned
* by default it is signed.

Tiny int (1 Byte)
Small int (2 Bytes)
Mediamint (3 Bytes)
int (4 Bytes)
long int (8 bytes)

* floating point type

float - upto 7 decimals
double - -I- 15 decimals.

* Fixed point Types (Exact value) :-

Decimal.

* stores double as a string. eg "635.7"
* Max number of digits is 65.
* Used when it is important to preserve the exact precision, for example with monetary data

* Boolean
* logical datatyes.
* True false (1, 0)
* insert (True, false, 1, 0)

* Date and Time datatypes:-

Date

* 1st Jan 1000 AD to 31st Dec 9999 AD
* no problem of Y2K in mySQL
* 'YYYY-MM-DD' is the default date format in MySQL.
  e.g - '2024-10-09'
* YY-MM-DD is also supported eg-'24-10-09'
* Year value range 70-99 are converted to 1970-1999
  —"——"——"— 00-69 are converted to 2000-2069
* date1-date2 -> returns the number of days between the 2 dates

* Time.

* hh:mm:ss or 'HHH:MM:SS'
* time values may range from '-838:59:59' to '838:59:59'
* Time1-time2

* Datetime

* Date and time is stored together
* YYYY-MM-DD hh:mm:ss is the default datetime format in MySQL.
* '1000-01-01 00:00:00' to '9999-12-31 23:59:59' is the range of datetime
* datetime1-datetime2 -> returns number of days, reminder hours, minule, and seconds between the two

* Year

YYYY. , 1990 to 2005.

In MySQL :-

- * You can have max 4096 columns per table provided the row size <= 65,535 bytes
- * No limit on the numbe of rows per table, provided the table size <= 64 terabytes

Largest table in the world →

- * order table of amazon.com → 100's of Terabytes daily.

* Create table emp

```
(
  Empno char(4), Ename, varchar(25),
  sal float, city varchar(15)] Dob date);
```

Table created.

- * (;) is known as delimeter (terminator)
- * it indicates end of command
- * SQL commands are case-insensitive

* Insert

```
insert into emp
values ('1', 'Amit', 5000, 'Mumbai', '1995-01-15');
```

* When you are inserting, the data is case-sensitive
* for char, varchar, date, time, datetime use ' '

EmpNo	Ename	Sal	City	DobDate
1	Amit	5000	Mumbai	1995-01-15.
3	- - -	7000	- - - -	- - - - -

null.

null

```
insert into emp (empno, sal)
values ('3', 7000)
```

* null values :-
  * null means nuthing.
  * null values is having ASCII value 0.
  * special treatment given to null value in all RDBMS
  * Null value is independent Of datatype
  * Null values accupies only 1 Byte of storag
  * If the row is ending with null values for th last column of If the row is ending with null values,
  * Thous columns that are likely to have a large number of nulls.

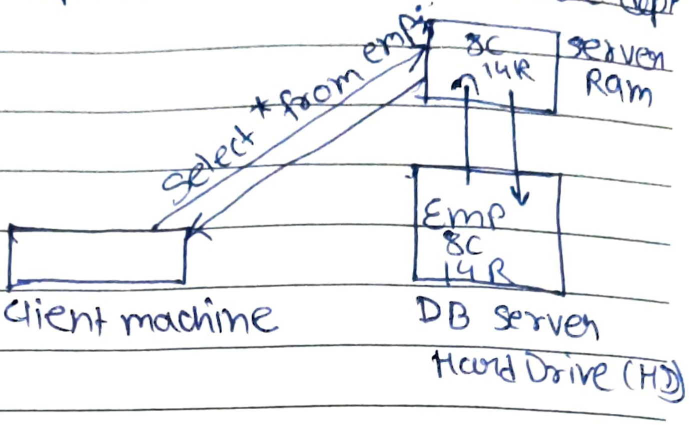
* Two multiple rows in a table using a single Insert statement.

```
insert into emp values
('1', 'Amit', 5000, 'Mumbai', '1995-01-15'),
('2', 'King', 6000, null, '1990-04-20'),
('3', null, 7000, 'Mumbai', 1994-03-2D,
```

★ Select

Select * from emp; // select the table emp
Select * from dept ; // select another table dept

1. Read.
2. Compile
3. plan
4. Execute.



To restricta the colums :-

empNo	Ename	Sal	City	DOB Date

// or Datatable

*Select empno, ename from emp;

emp No,	emp Name

~~Select~~

* Select Ename, Sal, DOBDate

ename	Sal	DOB Date

* The position of column in select statement will determine the position of columns in the output.
* this you will write as per client requirements.

select * from emp
where dept no =10;

EmpNo	Ename	JOB	MGR	HIREDATE	DepTNO	Sal
5	Suraj	Managen	7839	1981-11-17	·10	3500
7	Aditya	Clerk	7845	1990-12-25	10 ↑	1000

WHERE clause

* Where clause is used for searching
* Searching takes place in DB server: HardDrive
* where clause is used to restrict the rows
* where clause is used to retrieve the rows
from DB server HD to Server.

* Select * from emp.
where sal > 2000; // its print more than 2K
Salary.

Relational operator
> , >= , < , <= , != (not equal to) , =

* Select * from emp
where Sal > 2000 and sal < 3000;
// its print between 2K and 3K

Logical Operator :-
Not
And
Or

* select * from emp
Where (dept no = 10 or sal > 2000) and Sal < 3000;

```
select ename, Sal, Sal *12
from emp;
```

ename	Sal	Sal*12
Suraj	4000	48000
Shubham	8000	96000
Salil	3000	36000
tanmay	2000	24000

It is not a column its calculated, virtual, column.

After Insert wed then add the column.

Sal* 12

- * computed column, Calculated column, Virtual column, fake column
- * computed column, eg. Sal*12 are not saved inthe table that would be a wastage of HD spaces.

~~* as~~ and when

Arithmtic operators :-

1. ()
2. /
3. *
4. +
5. -

★ Select alias.

```
select ename, sal, sal* 12 as "Annual".
from emp
```

```
select ename, sal, sal* 12 "Annual"
from emp.
```

as → optional in MySQL . as → ANSI SQL

* Select ename, Sal, sal*12 annual
from emp;

->

ENAME	SAL	ANNUAL

It's print all data Apercase.

* select ename, sal, Sal*12 "ANNUAL"

->

ename	sal	ANNUAL

It's print only Annual in upercase using double quotes

* It's recommended that you should always use double quotes, it becomes more readable