Day-2 Date: / / Page no:

① Integer types (Exact values):-
Signed or Unsigned (by default it is signed)
Tinyint (1 Byte)
Smallint (2 Bytes) AGE tinyint unsigned.
Mediumint (3 bytes)
int (4 bytes)
Bigint (8 bytes)

② Floating-point type (Approximate value):
Float (upto 7 decimals) } beyond that
Double (upto 15 decimals) } rounding takes place.

③ Fixed-Point types (exact value):-
Decimal
Store doubles as a string
e.g. "65.357"
max number of digits is 65
e.g; Monetary data.
ex:- Deloitte uses upto 20 decimal to preserve exact precision

④ Boolean-datatype (logical datatype):-
- True / False 1/0
- can insert true/false or 1/0
- output displayed 1/0

⑤ Date and Time datatypes:-
Date 1st Jan 1000 AD to 31st dec 9999 AD (ii) there is no
Time (ii) Default date problem of year (2000)
DateTime format (Y2K) in MySQL
year YYYY-MM-DD
'2024-03-19'
'24-03-19' convert this to '2024-03-19'

- In MySQL, you can have max^m 4,096 columns per table, provide the rowsize <= 65,535 Bytes.
- no limit on number of rows per table. Date: / / ~~table size <= 64 TB~~
- year values in range 70-99 are converted to 1970-1999
- year values in range 00-69 are converted to 2000-2069.
  '47-07-15' → '2047-07-15'
- 1970 is known as the Year of the Epoch
  in 1970 Unix OS was introduced.

- date1 - date2 → returns number of days b/w the 2 dates.
-

1st Jan 1000 AD → 1 2nd Jan 1000 AD → 2
19th Mar 2024 AD → 737619 (number of days since 1st Jan 1000 AD)
(Internally stored as number of days since 1st Jan 1000 AD. Internally date is stored as a fixed length number and it occupied 4 bytes of storage.

⇒ Time: ('hh:mm:ss') or ('HH:MM:SS')
- time values may range from '-839:59:59' to '839:59:59'

⇒ Date Time :-
'YYYY-MM-DD hh:mm:ss'
'1000-01-01 00:00:00' to '9999-12-31 23:59:59'
datetime1 - datetime2 → returns the number of days between the two, remainder hours, remainder minutes and remainder seconds.
default value for time is 12 am midnight
e.g.: 00:00:00

→ year: YYYY
range: 1901 to 2155

100 TB
64 TB 36 TB

Desc <table_name> ;

Date: / / Page no:

Largest Table in the world..

ORDERS (amazon.com)

100's TB of ~~data~~ daily

In oracle there is no limit on ~~the~~ table size.

EMP

ENO EName Sal City ~~Dob~~. create table Emp

(

);

- SQL commands are case-insensitive.
- ; known as delimiter (denotes the end of command)

(=> Insert into Emp (EmpNo, EName, Sal, city, dob) values

more readable more flexible. (—, ——, —, —, —);

=> Update Emp set EName = 'Sudh' where EmpNo = 1001

=> Char, varchar, date, time & datetime are in single quotes.

=> ~~Em~~ null value is stored if no value is given.

=> ~~Null~~ special treatment given to null value in all ~~Rdbms~~

=> null value having arith value of 0

=> null value is independent of data type.

=> you can insert a null value for any columns of any datatype.

=> Null value occupies only 1 Byte of storage.

=> If you specify null value for the last column, or if the row is ending with null value, then it all those columns will not occupy any space.

Date: / / Page no:

⇒ Keep columns having large number of null values, at the end of table structure to conserve on HD space.

```
insert into emp
values ('4', 'Atul');        <- ERROR not enough values.

insert into emp
values ('4', 'Atul', null, null, null);
```

null is a command and reserved word.

Insert into emp values (5000, null, null, null);

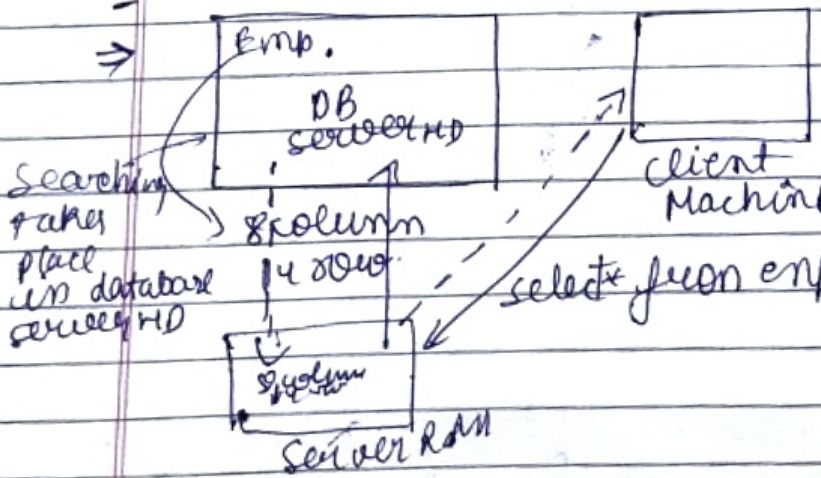
EMP

EMPNO	EName	Sal	City	DOB	GRST
1	Ajay	5000	Mum	2 →	null
2	Vinod	(18)	Del	2 →	null
3	Robin	(18)	(18)	(18)	null.

- In future if you alter a table and add a column, then its insert statement to work.

-

⇒



for a SQL query searching takes in DB server HD.

cl scr

Date: / / Page no:

→ Select * from emp;
* metacharacter (all columns)

⇒ Relational operators.

1. >
2. >=
3. <
4. <=
5. != or <>
6. =

To restrict columns & rows.

Precedence.

⇒ In mysql while inserting data is case-sensitive, while searching, queries are case-insensitive.

Select * from <table> where job = 'MANAGER' = Select * from <table> where job = 'manager'

More user-friendly, but less secure.

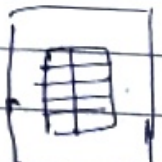
alias (as "12sal")

Select ename, sal, sal*12 from emp;

sal*12 → computed column (derived column) / Pseudocolumn / fake column.
↳ not saved in table as wastage of HD space.

Arithmetic operators: ( ) / + - *

as → ANSI SQL
as → optional in MySQL and Oracle.



1) Ename, sal
2) sal*12
3) Alias

Server RAM

exit; → save & commit & exit

→ optional double quote — for case-sensitive column name use double quote.

Select 12*sal annual from Employee;

⇒ we can specify alias for column of table also.
Select ename "EMPNAME" from emp;

⇒ Alias cannot be used in expression.
⇒ only for display purposes.

→ select job from emp;

→ select distinct job from emp;
↳ to suppress duplicate jobs.
* Whenever use distinct, sorting takes place in server RAM

*
→ Select distinct job, ename from emp;
→ distinct will operate on all columns that are present in a select statement.
→ distinct has to be first keyword after select

⇒ In a DBMS, the rows are stored in a file.
⇒ In a file, the rows are stored sequentially.
⇒ In RDBMS, table is not a file; every row is a file.
⇒ Rows inside a table are scattered (fragmented) all over the DB server HD;
⇒ The reason why does RDBMS does is to speed up the INSERT statement. Consider a multi-user environment; if multiple users are Inserting rows simultaneously into the same table, and if MySQL

Date: / / Page no:

were to store the rows sequentially, then it would be very slow.

→ When you insert a row into the table, wherever it finds the free space it will store the row there.

→ When you SELECT from the table, the order of rows in the output will depend on the row address, it will always be in ascending order of row address.

→ Once you insert a row, the row address will be constant

→ When you update the row, if the row length is increasing and if the free space is not available, then the entire row will be moved to some other address.

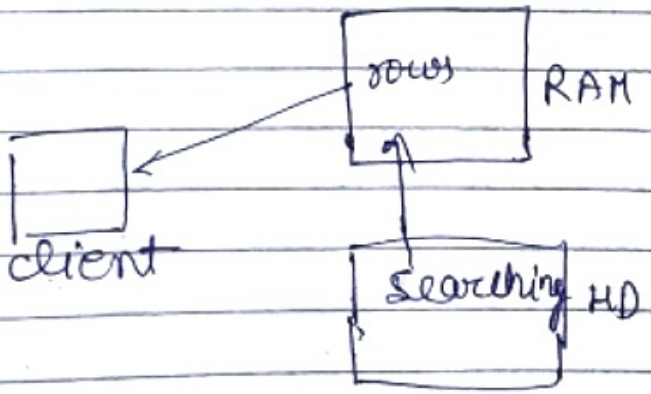
→ It's only in the case of VARCHAR that the row length may increase or decrease.

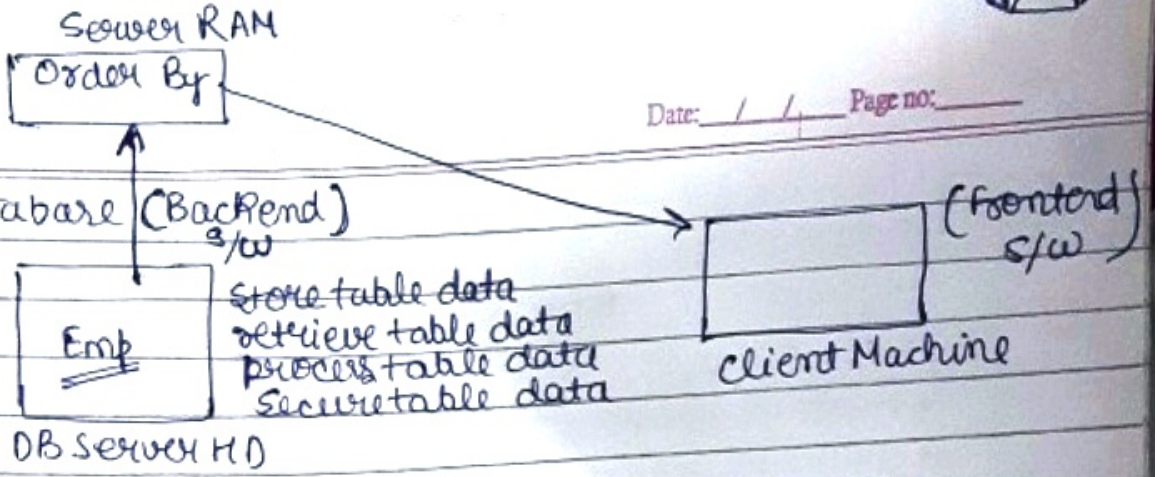
→ When you UPDATE a row, the row address may change.

→ Later when you SELECT from the table, you will see that row at some other portion.

→ Hence it is not possible to see the first 'N' rows of a table or the last 'N' rows of a table.

⇒ ORDER BY :- default ASC
↳ for sorting → in case of duplicate, rows sorted based on row address.





Date: / / Page no:

order by ename desc; | asc; → (by default)
desc;

Business Intelligence (BI)
Data Science / Data Analytics / Artificial Intelligence / ML,
SQL is pre-requisite

no upper limit on the columns in Order By claure

order by deptno, job;
↳ first by deptNo. then job.

order by deptno desc, job asc;

If you have a large number of rows in the table, then the SELECT statement will be very slow, because that much sorting has to take place in server RAM.

→ order by should be the last claure.

→ order by sal*12;

→ Select sal*12 annual from emp order by annual
↳ it will work b/c annual exist in Server RAM.

① ename, SAL
② sal*12
③ annual
④ order by

Select sal*12 "Annual" from emp. order by "Annual"

Select sal*12 from emp order by 2;
↳ sorted based on 2nd column.

AD@1234

Date: / / Page no:

→ When we install MySQL, 2 users are automatically created :-

- root — DBA priviliges = create database, use database, drop, alter, configure, create user, drop user, assign priviliges, performance, monitoring, performance tuning, performance planning, take backups etc.
- mysql.sys.
  - most important user
  - owner of database (MySQL installation)
  - startup database, shutdown database, perform recovery.

> show databases;
> use <database_name> / use mysql;
> select user from user;
↳ system table, created automatically at installation, store username.

→ Create a new user.
> create user <username> identified by <password>;
@localhost
username can connect to database using the local machine only.

Give all permissions to the new user on the new database.
> grant all priviliges on <database>.* (tables) to <username>@localhost
> flush priviliges → priviliges will be applied to new user without restart server.

float(3,2) - floating point number upto 9.99
int(3) → allocat at least 3 bits.

Date: / / Page no:

Blank-paded comparision semantics:

EMP

EMPNO	EName	Sal	City	DeptNO
1	ADAMS	1000	MUM	10
2	BLAKE	2000	DEL	10
3	ALLEN	2500	MUM	20
4	KING	3000	DEL	25
5	FORD	4000	MUM	30

Select * from emp where ename >='A' and ename < 'B';

1	ADAMS
3	ALLEN

ADAMS > A B _ _ _ (blank space)

=

ADAMS < B _ _ _ padded till same length

When we compare 2 strings of different lengths, the shorter of the 2 strings is temporarily paded with blank spaces on RHS such that their lengths become equal; then it will start the comparision character based on ASCII value.

select * from emp where ename like 'A%';

↳ wildcard

% any character & any number of characters.

select * from emp where ename = 'A%';

(comparing look for A%.

Select * from emp where ename like '%A%';

contains A - at start
- between
- at end.

Case-insensitive search.

Date: / / Page no:

select * from EMP where ename not like 'A%.';

Select * from EMP where ename like '__A%.';
↑ any one character.

Select * from emp where ename like '____';
any four letters.

Select * from emp where sal >= 2000 and sal <= 3000;

Select * from emp where sal between 2000 and 3000;
mutually-inclusive (2000 & 3000 are also included)

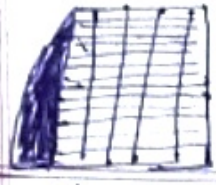
- Between → works faster because a readymad method by the name of BETWEEN is already present in the database in COMPILED format.

- NOT BETWEEN.

select * from emp where sal not between 2000 and 3000;
← exclusive.

Select * from emp where hiredate between '2023-01-01' and '2023-12-31';

select * from emp where ename between 'A' and 'F';
where ename >= 'A' and ename <= 'F';



Date: / / Page no:

IN - perform logical OR

select * from emp where deptNo = 10 or deptNo = 20 or deptNo = 30;

fastest Select * from emp where deptNo IN (10,20,30);

faster Select * from emp where deptNo = any (10,20,30);
↳ logical OR.

IN → use to check for equality & Inequality (fastest)
ANY → relational comparision.
for more complex requirements, then use relational & logical operators.
Select * from emp where deptNo >= any (10,20,30)
~~select~~ ... where deptNo <= any (10,20,30)
>
<
!=
=

ANY is more powerful than IN.

Select * from emp where city in ('Mumbai', 'Delhi');

- IN operator can be used by itself in MySQL and other RDBMS.
- ANY operator can be used by itself in other RDBMS, but any operator cannot be used by itself in MySQL.
- In MySQL, the any operator can only be used with sub-queries.

Date: / / Page no:

select * from emp where city = any ('Mumba
'Ddhi'.

↑ supported in other RDI
not in MySQL

... where city = any (select ....);

↑ supported in
MySQL.

UPDATE - to modify row

update table_name set column_name = < > where
column_name = < > ;

update emp set sal = sal + sal*0.4 where empN
" " " " , city = 'Pune'
where empNO = 1;

- we can update multiple row & colmans simultaneous
but you can update only 1 table at a time.

DELETE

Delete from emp where empNO = 1.
delete from emp; → all the rows will delete.

DROP

drop table emp, dept, customers, orders;
where clause ccan not be used with drop; bec
DROP is a DDL command.