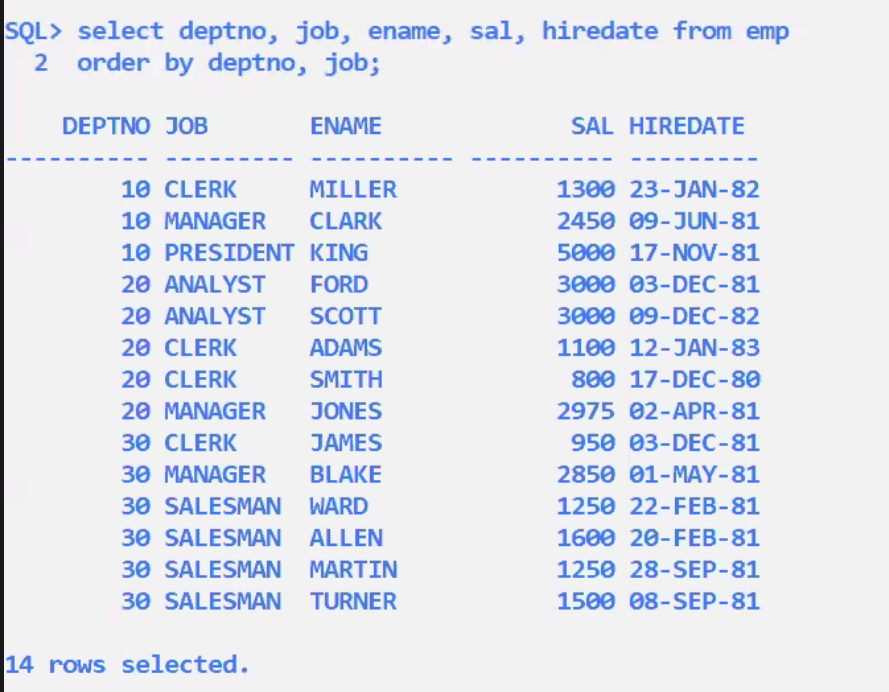
select deptno, job,ename,sal, hiredate from emp

order by deptno, job;



select deptno, job,ename,sal, hiredate from emp

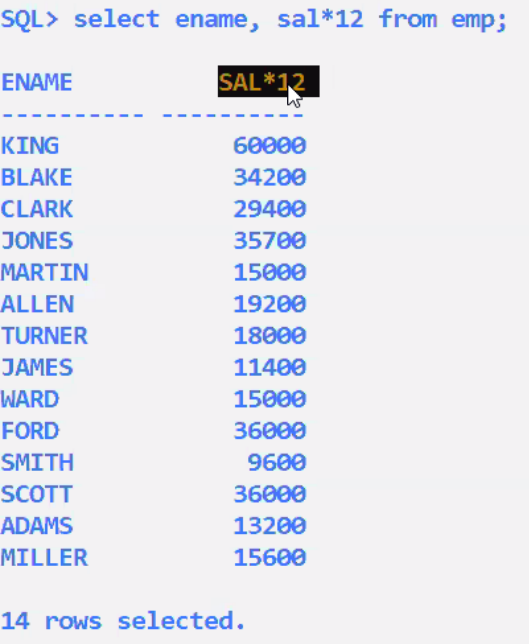
order by deptno desc, job;

select deptno, job, ename, sal, hiredate from emp

order by deptno, job desc;

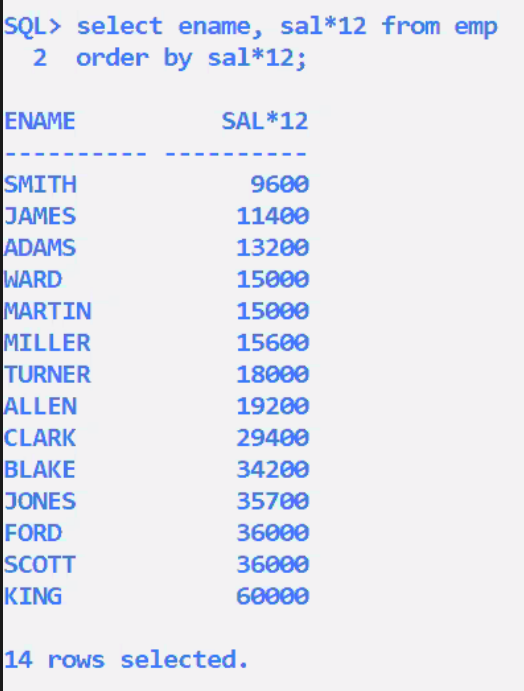
* No upper limit on the number of columns in order by clause
* Select ………………………….
* Order by country, state, district,city;
* Separated by commas
* If you have large number of rows in the table,and if you have large number of columns in ORDER by clause, then your SELECT statement will be slow; because that much sorting has to take place in server ram
* Sorting is one operation that always slows down your select statement

Select ename, sal\*12 from emp;



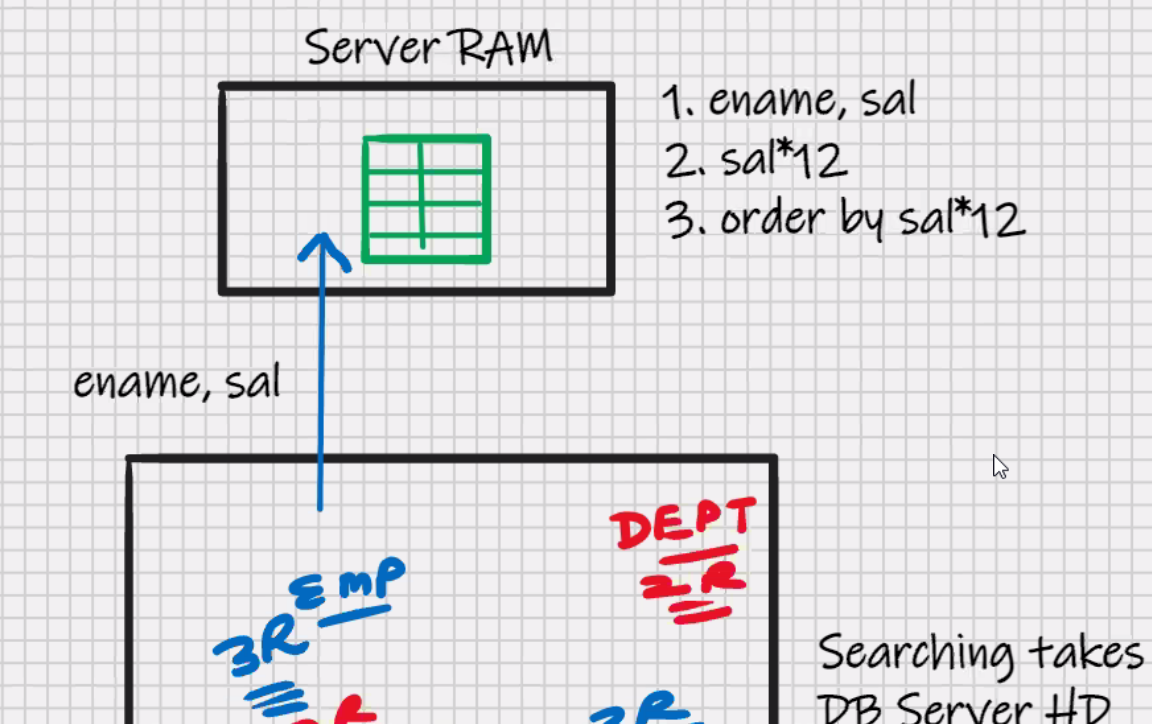
Select ename, sal\*12 from emp;

Order by sal\*12;



Select ename, sal\*12 from emp;

Order by sal\*12 desc;

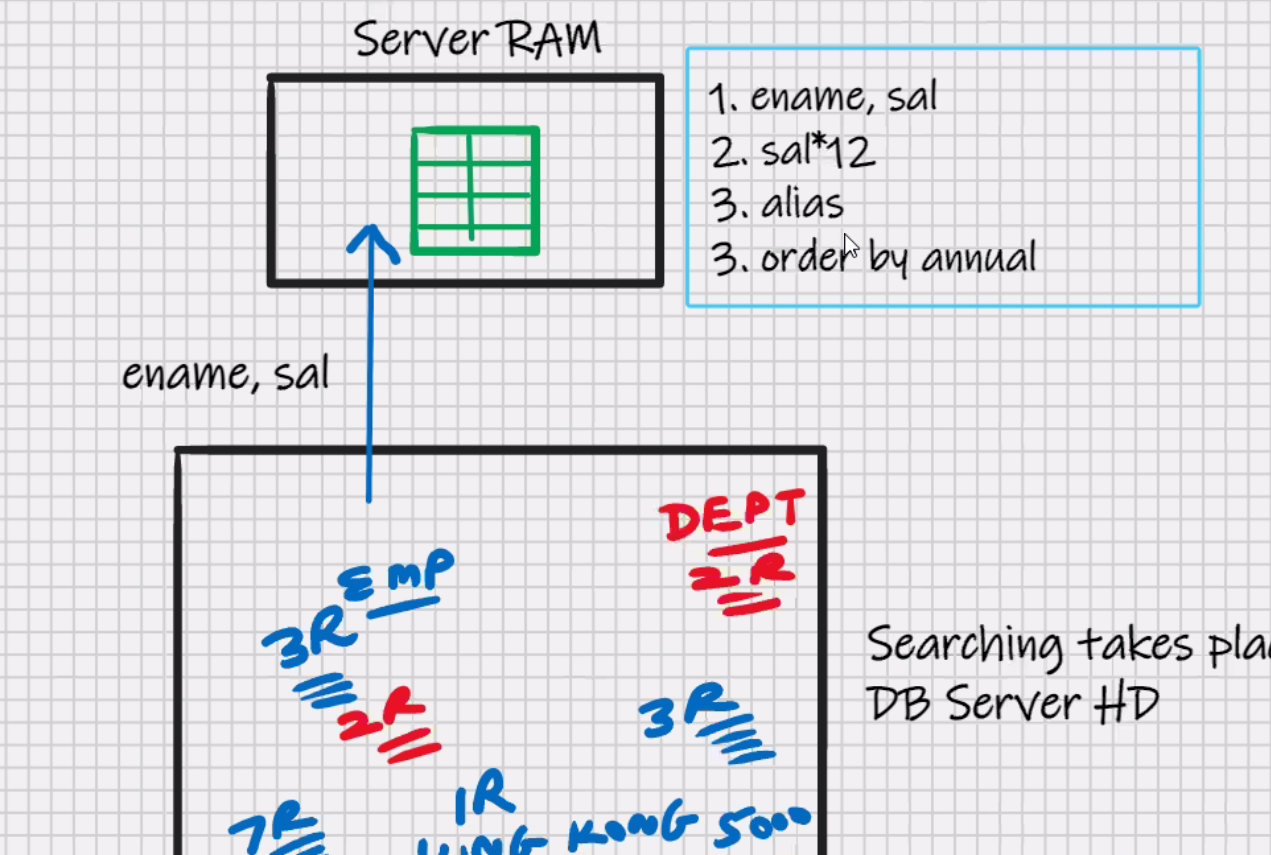


Select ename, sal\*12 annual from emp;

Order by annual;

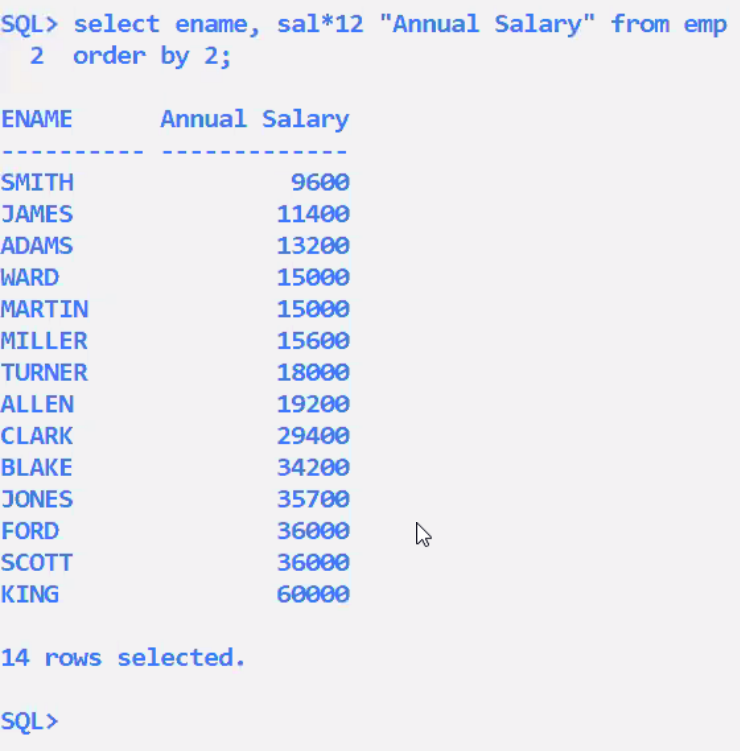
Select ename, sal\*12 “annual” from emp;

Order by “annual”;



Select ename, sal\*12 “annual salary” from emp;

Order by 2;



Select \* from emp

Order by 2;

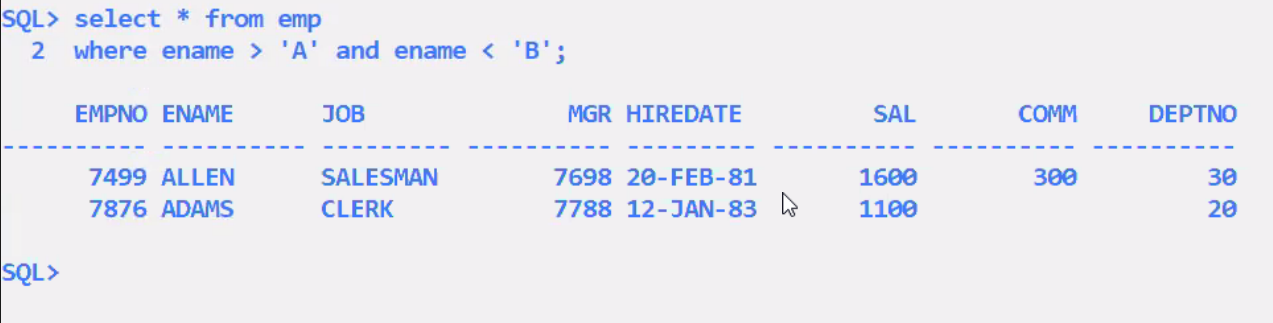
**MySQL – SQL-Special operator (like,between,any)**

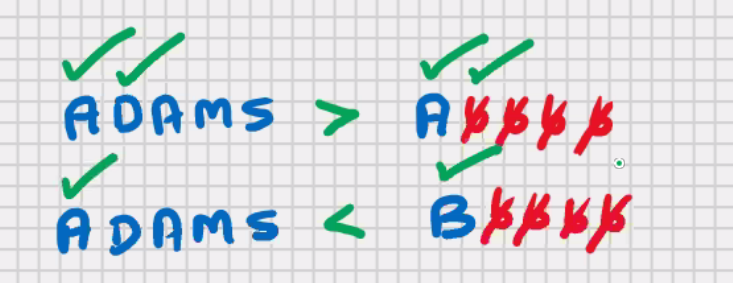
**EMP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMPNO** | **ENAEM** | **SAL** | **CITY** | **DEPTNO** |
| **1** | **ADAMS** | **1000** | **MUMBAI** | **10** |
| **2** | **BLAKE** | **2000** | **DELHI** | **10** |
| **3** | **ALLEN** | **2500** | **MUMBAI** | **20** |
| **4** | **KING** | **3000** | **DELHI** | **25** |
| **5** | **FORD** | **4000** | **BHOPAL** | **30** |

Select \* from emp

Where ename > ‘A’ and ename < ‘B’ ;





Blank-padded comparision semansistics:-

When you compare two strings of different lengths, the shorter of the 2 strings is temporarily padded with blank spaces on RHS, such that their lengths become equal; then it will start the comparisons, character by character, based on ASCII value

Select \* from emp

Where ename >= ‘A’ and ename < ‘B’ ;

Select \* from emp

Where ename like ‘A%’;

Wildcards(used for pattern matching)

% any character and any number of characters

\_ any 1 character

To make it case insensitive , solution for oracle:-

Select \* from emp

Where ename like ‘A%’ or ename like ‘a%’;

Select \* from emp

Where ename like ‘%A’; //ending with A

Select \* from emp

Where ename like ‘%A%’; //any where with A

Select \* from emp

Where ename like ‘\_ \_A%’; //all rows with A as third letter

Select \* from emp

Where ename like ‘\_ \_ \_ \_’; //king ford

Select \* from emp

Where ename like ‘A \_ \_ B’;

Select \* from emp

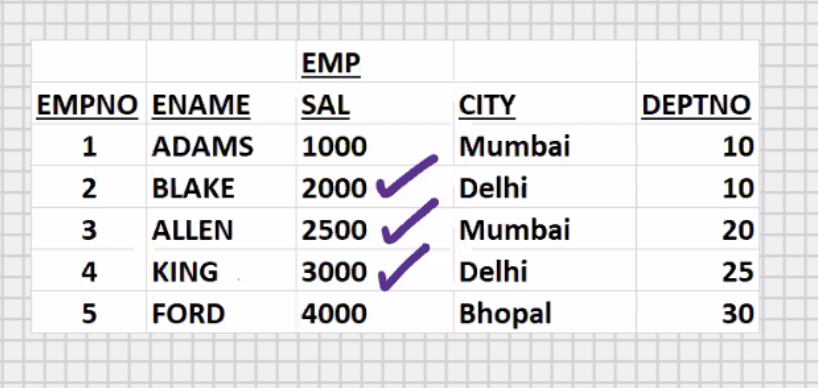
Where ename like ‘A%’;

Select \* from emp

Where ename not like ‘A%’; //not starting with A

Select \* from emp

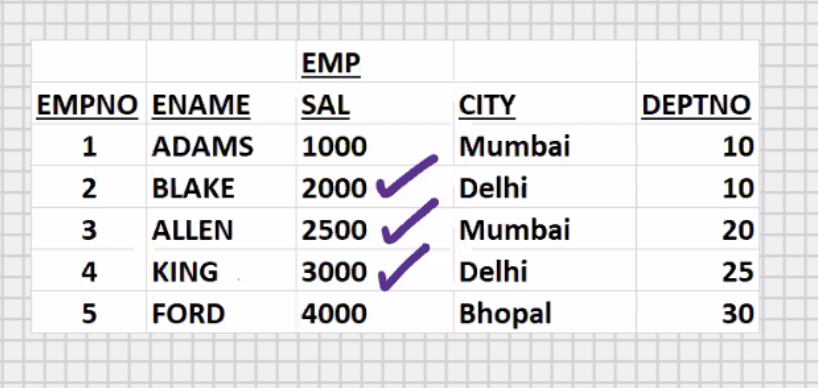
Where sal >= 2000 and sal <= 3000;



Select \* from emp

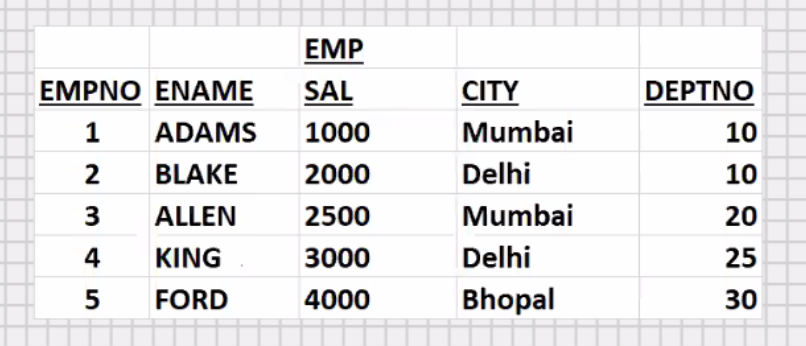
Where sal between 2000 and 3000; // mutually inclusive *\*recommended*

Easier to write and works faster



Select \* from emp

Where sal not between 2000 and 3000; //exclusive





Select \* from emp

Where hiredate >= ‘2021-01-01’ and hiredate <= ‘2021-12-31’;

Select \* from emp

Where hiredate between ‘2021-01-01’ and ‘2021-12-31’;

*\*Recommended*

Select \* from emp

Where ename >= ‘A’ and ename <= ‘F’;

Select \* from emp

Where ename between ‘A’ and ‘F’;

*\*Recommended*

Select \* from emp

Where ename like ‘A%’ or ename like ‘F%’;

Special operator – Any,in

Select \* from emp

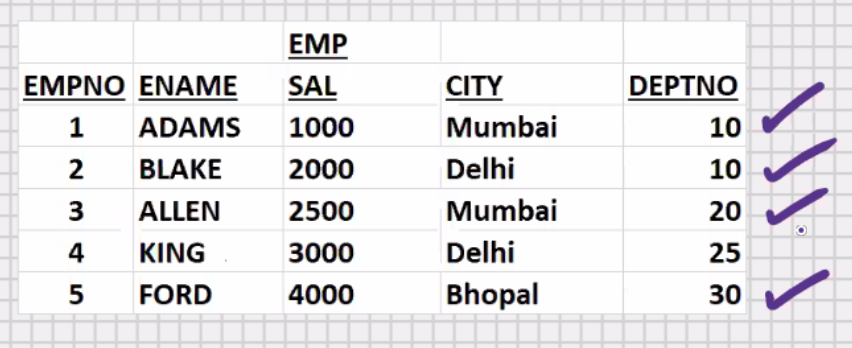
Where deptno=10 or deptno=20 or deptno=30;

Select \* from emp

Where deptno=any(10,20,30);

Select \* from emp

Where deptno (=,>,>=,<,<=,!=,==) any(10,20,30);



* **Any** operator will perform **Logical OR**
* Easier to write and Works faster

Select \* from emp

Where deptno in(10,20,30);

* **in** operator will perform **Logical OR**
* Easier to write and Works faster(**Fastest**)

Select \* from emp

Where deptno not in(10,20,30);

* In operator is faster than any operator
* Any operator is more powerful than in operator
* With in operator, you can only check for **in and not in**
* With any operator you can check for = any, !=any, >any, >=any, <=any
* If you want to check for equality or inequality, then use the in operator if you want to check for >,>=,<,<= then use the **any** operator
* In and any operator works with **all data types**.

Select \* from emp

Where city in(‘mumbai’,’bhopal’);

Select \* from emp

Where city not in(‘mumbai’,’bhopal’);

Select \* from emp

Where city = any(‘mumbai’,’bhopal’);

* Any operator does not work directly in MySQL e.g.

Select \* from emp

Where deptno= any (10,20); //not supported in mysql

* In mysql , **any** operator has to be used with **sub-query**
* In mysql, **use the in operator**

MySQL – SQL

DDL -> create, drop

DML -> insert , update , delete

DQL -> select

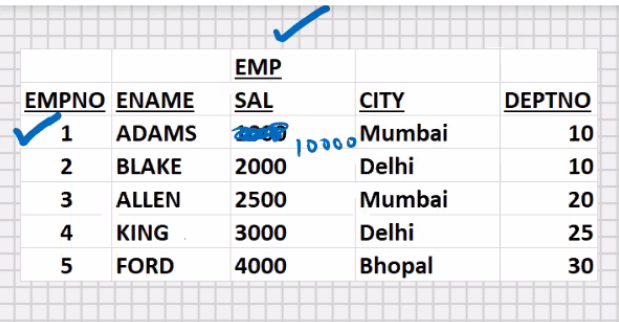
UPDATE

--------------------------------------------------------------------------------------------

Update emp

Set sal = 10000

Where empno = 1;



Update emp

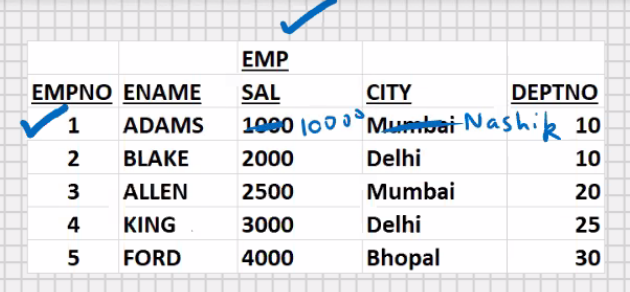
Set sal = sal + sal \* 0.4

Where empno = 1; //increased by 40%

Update emp

Set sal = 10000, city = ‘nashik’

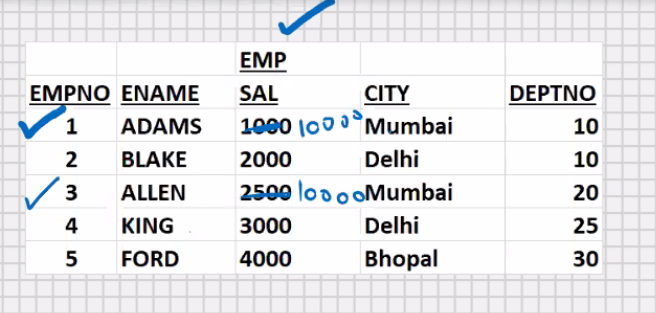
Where empno = 1;



Update emp

Set sal = 10000

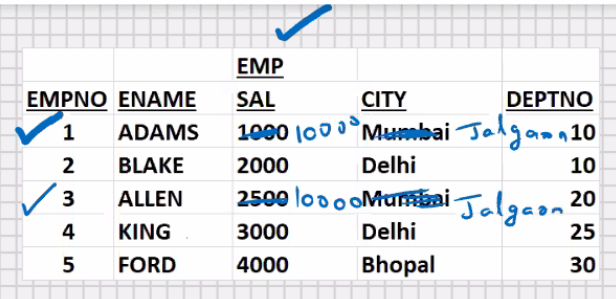
Where city =’mumbai’;



Update emp

Set sal = 10000 , city = ‘jalgaon’

Where city =’mumbai’;



* You can update multiple rows and multiple columns simultaneously, but **only 1 table at a time**
* If you want to update 2 or more tables, then a separate update command is required for each table

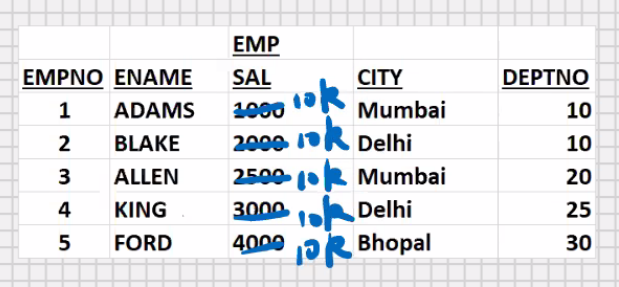
Update emp

Set sal = 10000 , city = ‘jalgaon’

Where empno between 1 to 3;

Update emp

Set sal = 10000 ; // all rows will get updated

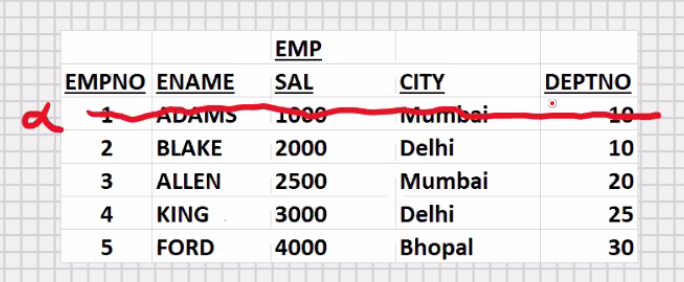


DELETE

--------------------------------------------------------------------------------------------

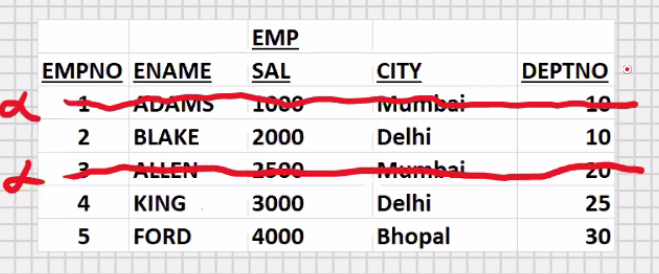
Delete from emp

Where empno = 1;

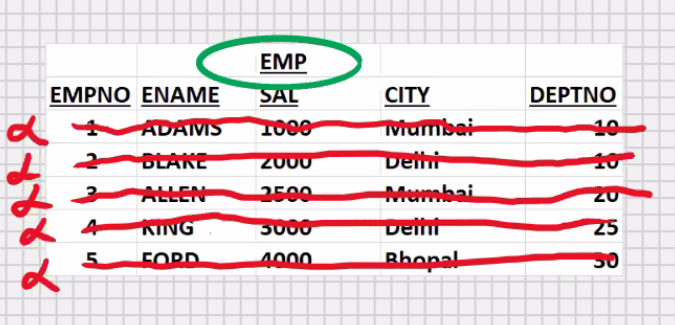


Delete from emp

Where city = ‘mumbai’;



Delete from emp;



Drop table emp; // delete the table forever

Drop table emp, dept; //deleting multiple tables

* You cannot use where clause with drop table,

Because drop table is a DDL command

* Update and delete commands without where clause will not be allowed in mysql workbench

To use update and delete without **where clause in MySql** workbench:-

Click on edit (menu at the top) ->preferences -> SQL editor ->

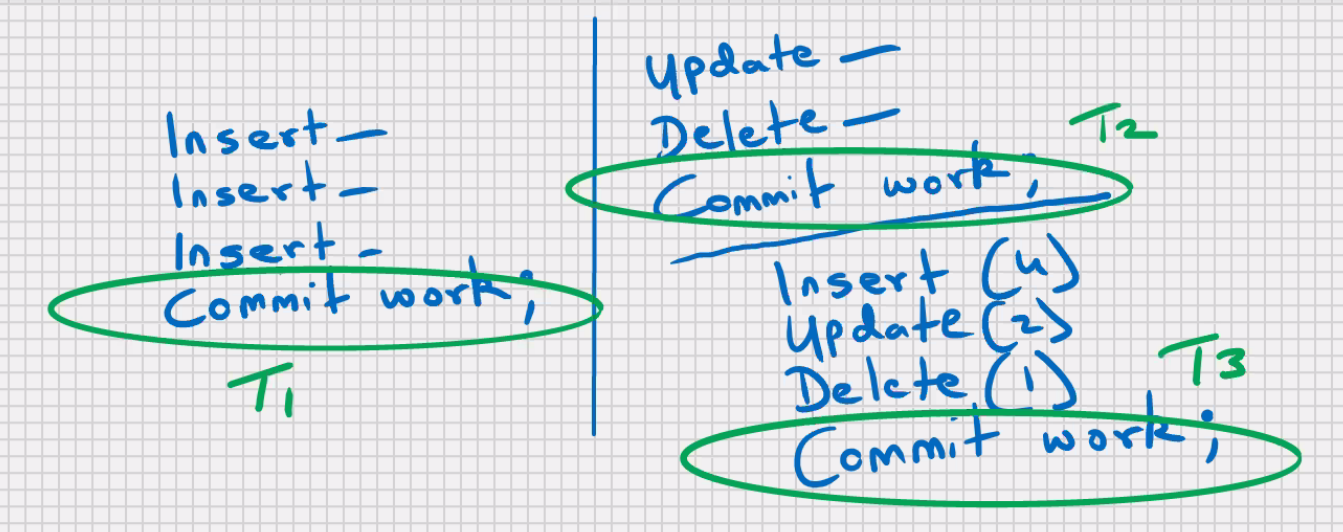
“safe updates” (checkbox at the bottom) -> uncheck it -> click on OK

This requires a reconnection to the server

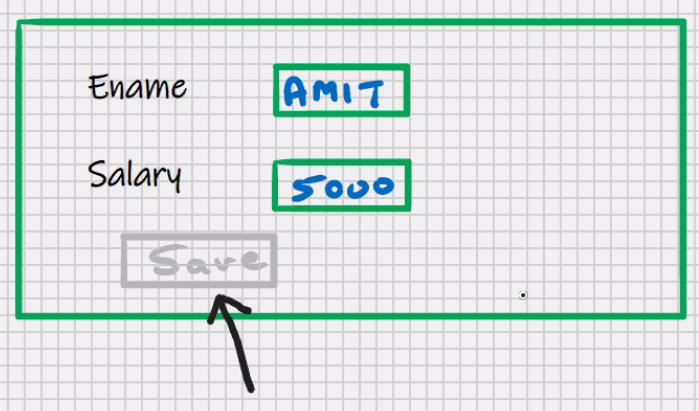
Click on Query (menu at the top) -> Reconnect to Server -> click on it

MySQL- SQL – Transaction Processing

* Commit will save all the DML(insert,update,delete) changes since the last committed state



* When the user issues a commit, it is known as end of Transaction
* Commit will make the Transaction permanent



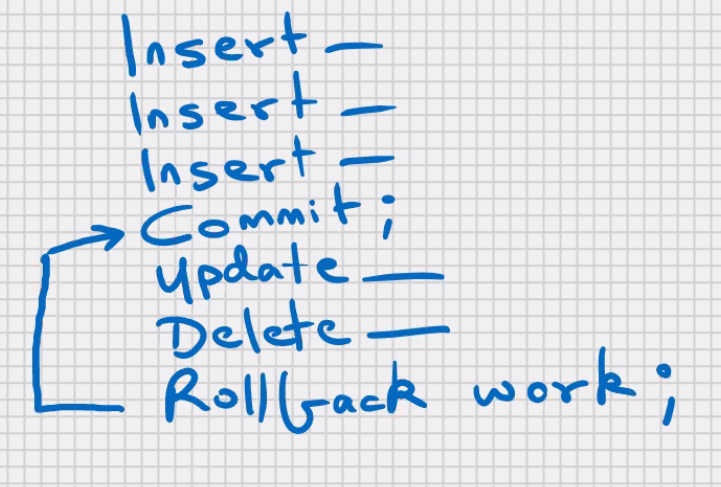
Commit work;

Work -> ANSI SQL

Work -> optional in MySQL and Oracle

Commit;

Total Work Done = T1 + T2 + T3 + … + Tn;



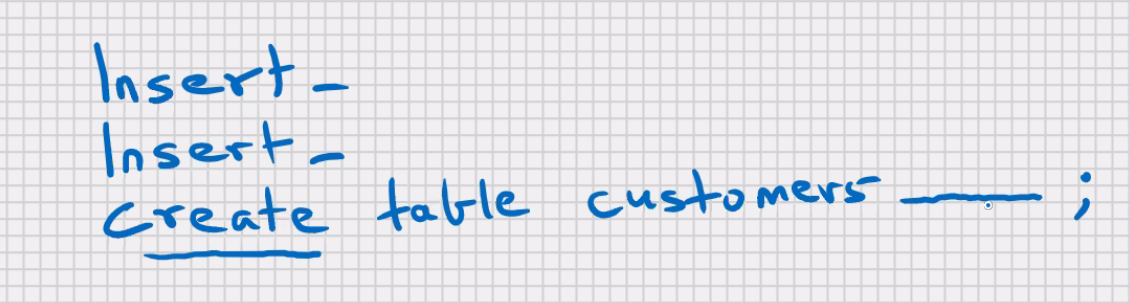
* Rollback will undo all the DML changes since the last commited state what is committed, that cannot be rolled back

Rollback work;

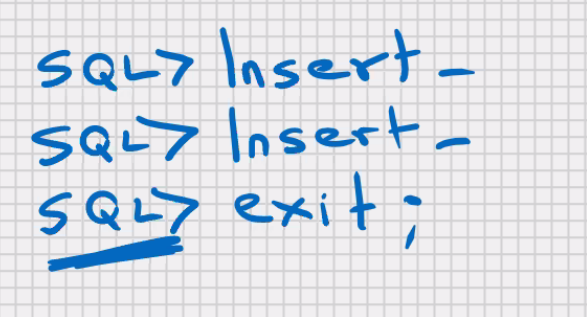
Or

Rollback

* Work is ANSI SQL
* Work is optional in MySQL and Oracle
* Only the DML commands are affected by rollback and commit
* **Any DDL command, it Automatically commits**



* **In oracle** when you exit from **SQL plus** it automatically commits



* Any kind of power failure, network failure, system failure, pc reboot, window close , improper exit, etc.; your last uncommitted transaction is automatically rolled back

**To try out rollback, commit, and savepoint in mysql workbench**

**Click on query (menu at the top) ->auto commit transactions -> uncheck it**

Assignment

Oracle 1-4

SQL 1-2