09 May 2022 14:24

Insert-
Insert-
Insert-
Savepoint abc; MAX 30 CHARACTERS , SAME RULES
update-
update-
Savepoint pqr;
delete-
delete-
Rollback to abc;

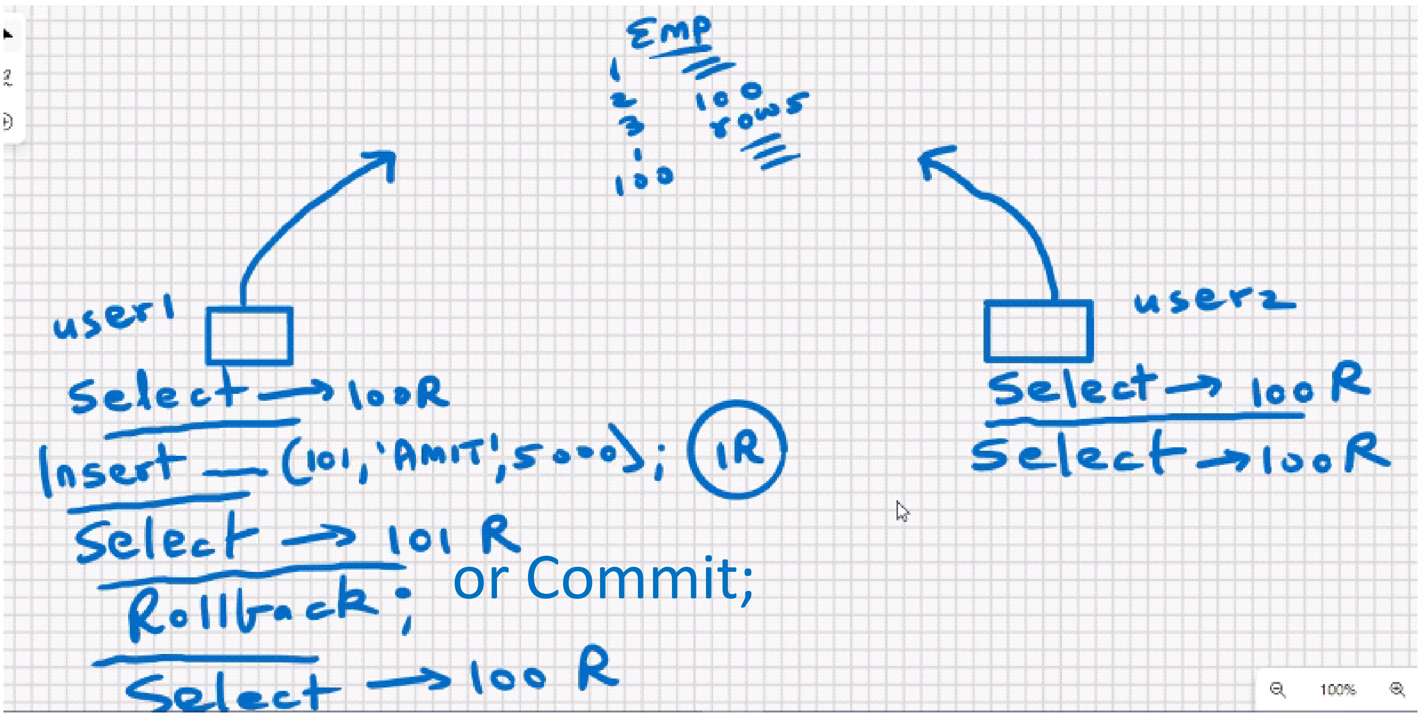
rollback work to pqr ;
Or
rollback to pqr;

- WORK in ANSI SQL
- WORK is optional in MySQL and Oracle
- You CANNOT COMMIT TO A SAVEPOINT
- Commit will save all the DML changes since the last committed state
- *You can only Rollback sequentially
- *When you Rollback or Commit, the intermediate Savepoints are cleared ; if you want to use them again, then you will have to reissue them in some new Work.
- Within a transaction, you can have 2 Savepoints with the same name; the new Savepoints overrides the previous one; the older Savepoint no longer exists.
- Savepoints will be used in small and large Transactions.

To try out Commit, Rollback and Savepoint in MYSQL Workbench:-
Click on edit -> Auto-Commit Transactions -> Uncheck it

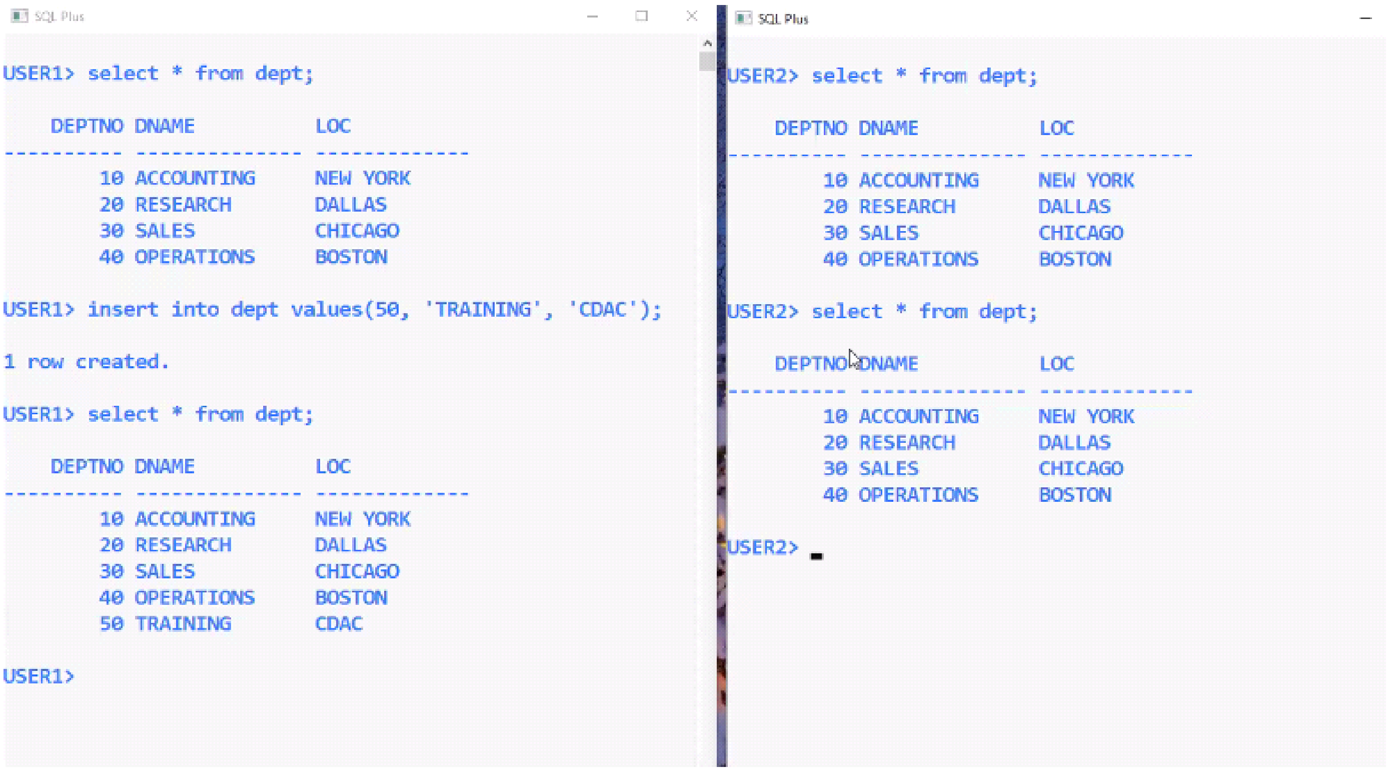
IN ORACLE => SET AUTOCOMMIT ON/OFF

READ AND WRITE CONSISTENCY :



Screen clipping taken: 09-05-2022 15:03

When you SELECT from table, you can view only the committed data of all users plus changes made by you.



```
USER1> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON

USER1> insert into dept values(50, 'TRAINING', 'CDAC');

1 row created.

USER1> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        50 TRAINING       CDAC

USER1>
```

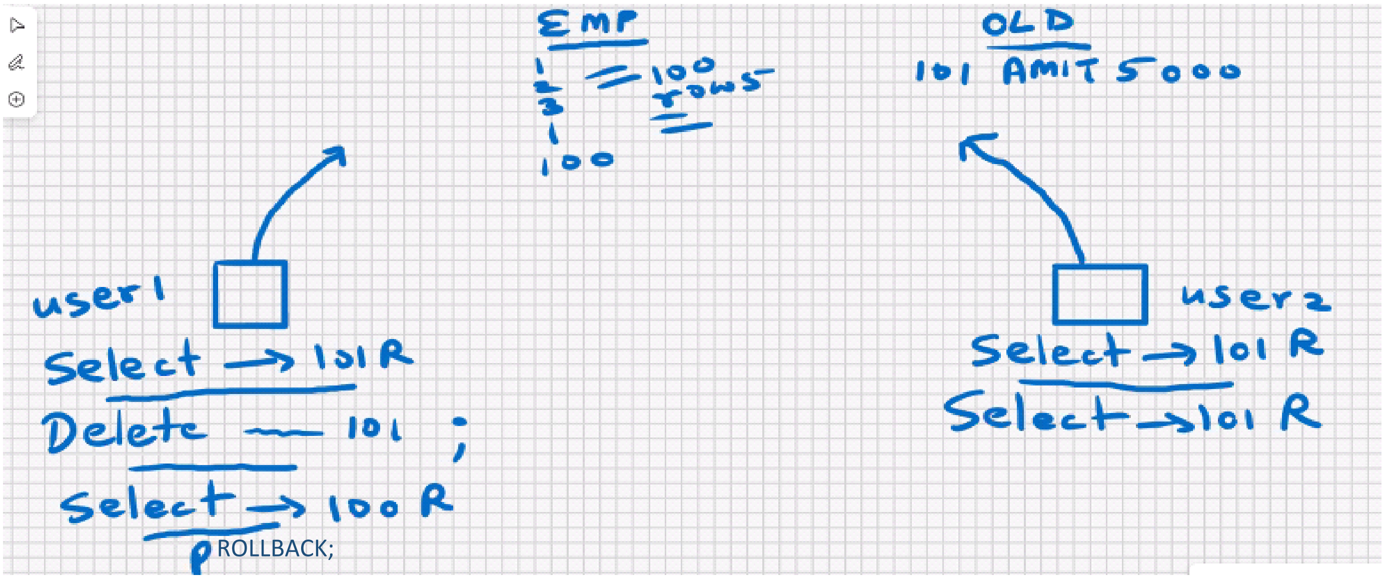
```
USER2> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON

USER2> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON

USER2>
```

Screen clipping taken: 09-05-2022 15:11

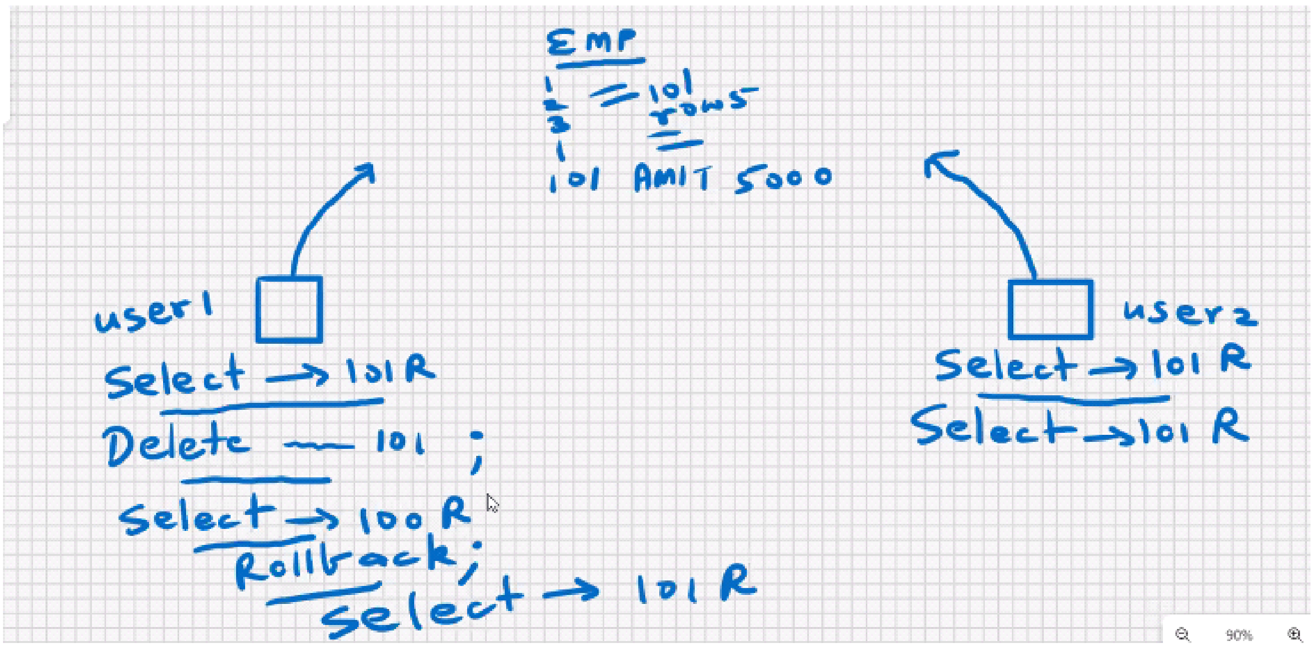
USER 2 DOESN'T KNOW ABOUT THE CHANGES MADE BY USER 1

BUT AFTER COMMIT => IT BECOMES PERMANENT => NEW ROW IS INSERTED INTO DEPT TABLE AND THE TEMPORARY TABLE IS DELETED

LIFE OF NEW TABLE IS TEMPORARY " 1.13 HR 09-05-2022 15:13 ".



Screen clipping taken: 09-05-2022 15:19



Screen clipping taken: 09-05-2022 15:22

ROW LOCKING :

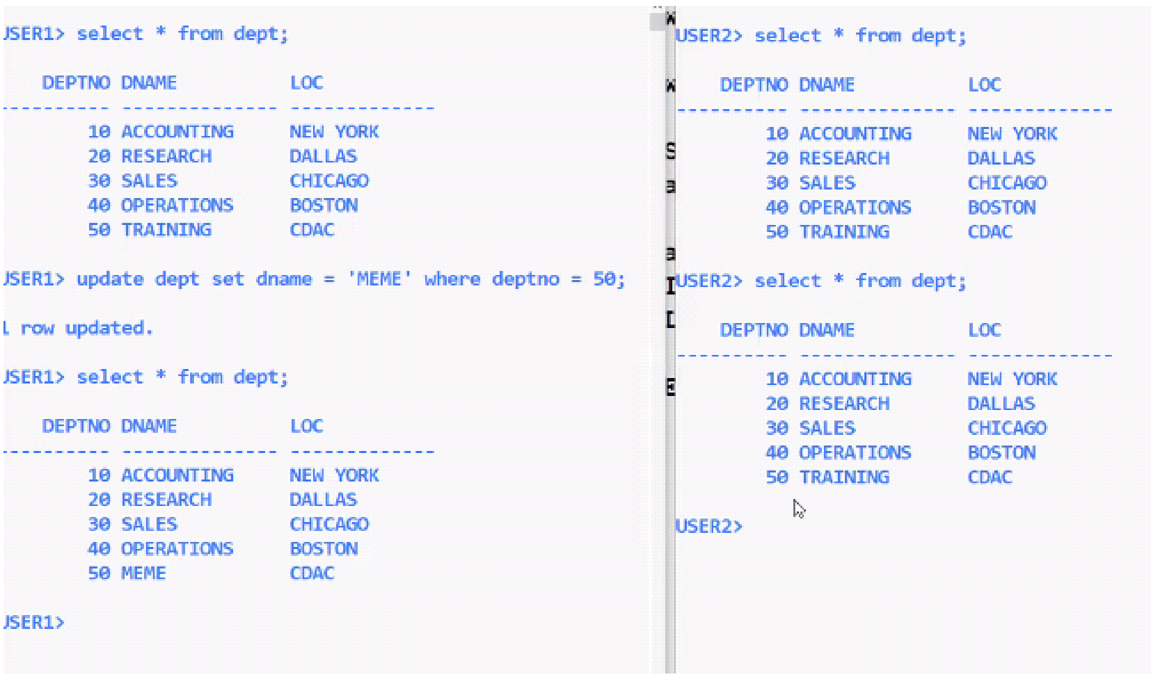
- When you UPDATE or DELETE a row, that row is automatically locked for other users.

- When you UPDATE or DELETE a row, that row becomes READ_ONLY for other users.

- ROW LOCKING IS AUTOTMATIC IN MYSQL AND ORACLE.

- Other users can SELECT from that table; they will view the old data before your changes.
- Other users can INSERT into that table.
- Other users can UPDATE or DELETE "other" rows from that table no other user can UPDATE or DELETE your locked row, till you have issued a Rollback or Commit.

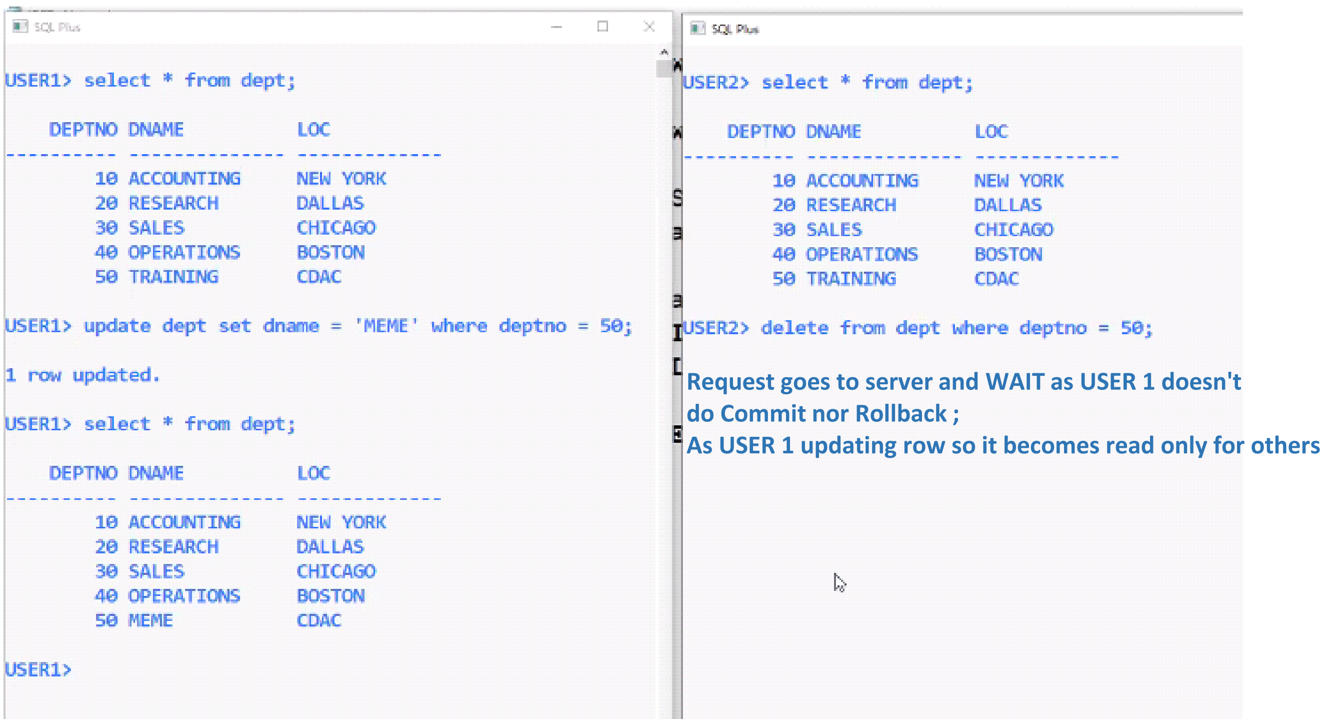
- LOCKS ARE AUTOMATICALLY RELEASED WHEN YOU ROLLBACK OR COMMIT.



```
JSER1> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        50 TRAINING       CDAC

JSER1> update dept set dname = 'MEME' where deptno = 50;

L row updated.

JSER1> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        50 MEME           CDAC

JSER1>
```

```
USER2> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        50 TRAINING       CDAC

USER2> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        50 TRAINING       CDAC

USER2>
```

Screen clipping taken: 09-05-2022 15:42



```
USER1> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        50 TRAINING       CDAC

USER1> update dept set dname = 'MEME' where deptno = 50;

1 row updated.

USER1> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        50 MEME           CDAC

USER1>
```

```
USER2> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        50 TRAINING       CDAC

USER2> delete from dept where deptno = 50;
```

Request goes to server and WAIT as USER 1 doesn't do Commit nor Rollback ;
As USER 1 updating row so it becomes read only for others

Screen clipping taken: 09-05-2022 15:43

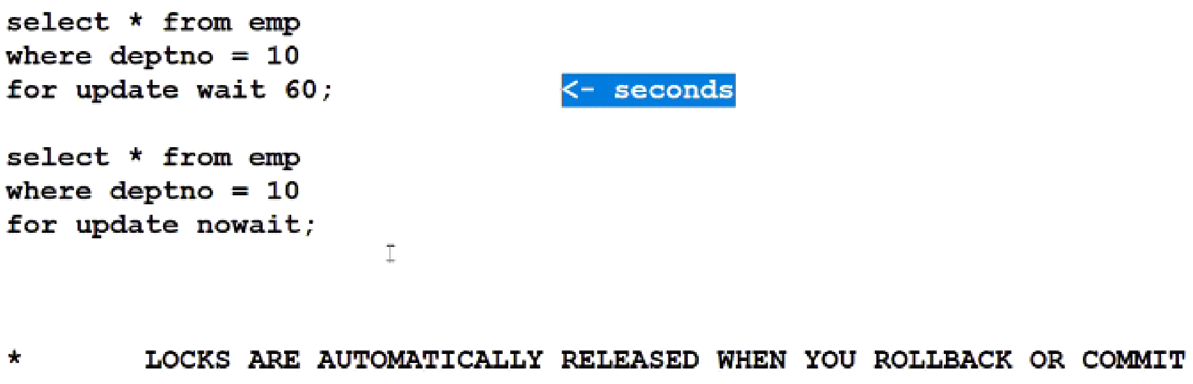
In MULTI - USER ENVIRONMENT SERVER maintains request queue (FIFO order) for same requests .

Optimistic Locking -> Automatic row locking mechanism in MYSQL and Oracle .

- Pessimistic Locking -> Lock the rows MANUALLY IN ADVANCE before issuing UPDATE or DELETE .
  - to lock the rows manually, then you have to use select statement with the FOR UPDATE clause.
    for eg: select * from emp where deptno - 10
     for update;
  - LOCKS ARE AUTOMATICALLY RELEASED WHEN YOU ROLLBACK OR COMMIT.

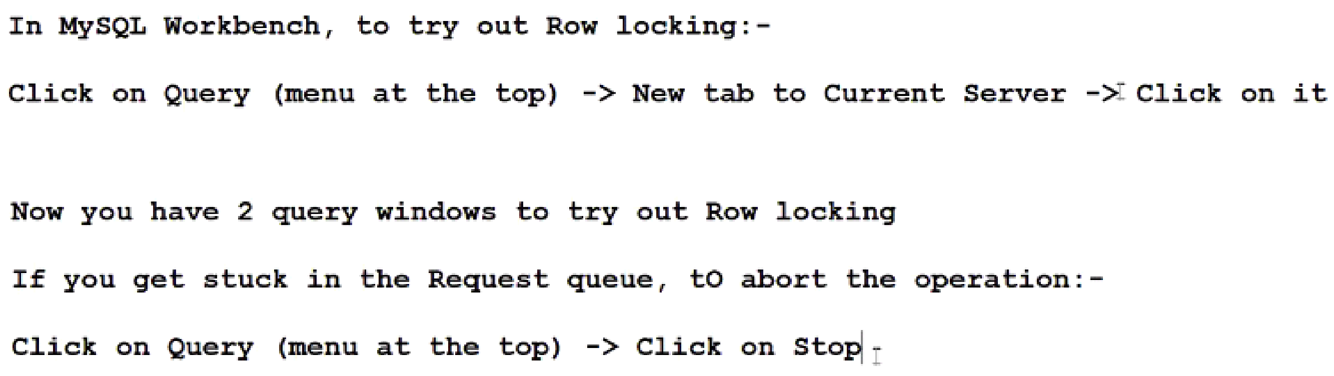
USING WAIT OR NO WAIT :

Screen clipping taken: 09-05-2022 16:13



```
select * from emp
where deptno = 10
for update wait 60;            <- seconds

select * from emp
where deptno = 10
for update nowait;


*      LOCKS ARE AUTOMATICALLY RELEASED WHEN YOU ROLLBACK OR COMMIT
```

Screen clipping taken: 09-05-2022 16:16



In MySQL Workbench, to try out Row locking:-

Click on Query (menu at the top) -> New tab to Current Server -> Click on it

Now you have 2 query windows to try out Row locking

If you get stuck in the Request queue, tO abort the operation:-

Click on Query (menu at the top) -> Click on Stop

Screen clipping taken: 09-05-2022 16:07

WAIT / NOWAIT OPTIONS ARE NOT AVAILABLE IN MYSQL .