<u>LOCKS:</u> **IT IS A MECHANISM WHICH IS USED TO PREVENT UNAUTHORIZED ACCESS FOR OUR RESOURCE. ALL DATABASE SYSTEMS ARE HAVING TWO TYPES OF LOCKS. THOSE ARE,**

- **i) ROW LEVEL LOCKS**
- **ii) TABLE LEVEL LOCKS**

<u>ROW LEVEL LOCKS:</u> **IN ROW LEVEL LOCKING WE ARE LOCKING A ROW / SET OF ROWS FROM THE TABLE. IN ALL DATABASES WHENEVER WE ARE USING COMMIT / ROLLBACK COMMAND THEN ONLY LOCKS ARE RELEASED.**

<u>EX. ON ROW LEVEL LOCKING ON A SINGLE ROW:</u>

<u>USER - 1:</u>

```
SQL> CONN SYSTEM / MANAGER;

SQL> UPDATE SCOTT.EMP SET SAL=2000 WHERE EMPNO=7788;

        1ROW UPDATED.

SQL> COMMIT; [FOR RELEASING LOCKS]
```

<u>USER - 2:</u>

```
SQL> CONN SCOTT/TIGER;

SQL> UPDATE EMP SET SAL=3000 WHERE EMPNO=7788;
```

[WE CANNOT PERFORM UPDATE OPERATION BCZ THIS ROW IS LOCKED BY THE USER SYSTEM]

<u>EX. ON ROW LEVEL LOCKING ON SET OF ROWS:</u>

WHEN WE ARE LOCKING SET OF ROWS FROM TABLE THEN USE “FOR UPDATE” CLAUSE IN SELECT QUERY.

<u>USER - 1:</u>

```
SQL> CONN SYSTEM / MANAGER;

SQL> SELECT * FROM SCOTT.EMP WHERE DEPTNO=10 FOR UPDATE;

SQL> COMMIT; [FOR RELEASING LOCKS]
```

USER - 2:

```
SQL> CONN SCOTT/TIGER;

SQL> UPDATE EMP SET SAL=3500 WHERE DEPTNO=10;

[WE CANNOT PERFORM UPDATE]
```

DEAD LOCK: IN ORACLE DEAD LOCKS OCCURS WHENEVER TWO / MORE THAN TWO SESSIONS WAITING FOR DATA IF THOSE SESSIONS ARE ALREADY LOCKED TO EACH OTHER.

THESE DEAD LOCKS ALSO RELEASED WHEN WE ARE USING COMMIT / ROLLBACK COMMAND.

EX:

USER - 1:

```
SQL> CONN SYSTEM / MANAGER;

SQL> UPDATE SCOTT.EMP SET SAL=3000 WHERE EMPNO=7369;

        1ROW UPDATED.

SQL> UPDATE SCOTT.EMP SET SAL=5000 WHERE EMPNO=7788;

ERROR: DEADLOCK DEFECTED WHILE WAITING FOR RESOURCE.

SQL> COMMIT; [FOR LOCKS RELEASE]
```

USER - 2:

```
SQL> CONN SCOTT / TIGER;

SQL> UPDATE EMP SET SAL=4000 WHERE EMPNO=7788;

        1ROW UPDATED.

SQL> UPDATE EMP SET SAL=6000 WHERE EMPNO=7369;
```

TABLE LEVEL LOCKING: IN THIS LEVEL WE ARE LOCKING A TABLE (ALL ROWS). ORACLE HAVING TWO TYPES OF TABLE LEVEL LOCKING

i) **SHARE LOCK**
ii) **EXCLUSIVE LOCK**

* Share Lock means multiple users can lock a table.

* Exclusive lock means only user will lock a table.

<u>SHARE LOCK:</u> SHARED LOCK EXISTS WHEN TWO TRANSACTIONS (USERS) ARE GRANTED READ ACCESS. ONE TRANSACTION GETS SHARED LOCK ON DATA AND WHEN THE SECOND TRANSACTION REQUESTS THE SAME DATA IT IS ALSO GIVEN A SHARED LOCK. BOTH TRANSACTIONS ARE READ-ONLY MODE. HERE AT A TIME NO. OF USERS ARE LOCKS THE RESOURCES.

UPDATING DATA NOT ALLOWED UNTIL THE SHARED LOCK IS RELEASED BY USING COMMIT/ROLLBACK.

SYNTAX:

```
LOCK TABLE <TN> IN SHARE MODE;
```

Note: A Share Lock allows multiple users to read a resource (like a table or row), but prevents others from modifying it.

* Share Lock means multiple users can lock a table.

* Exclusive lock means only user will lock a table.

<u>EX:</u>

<u>USER - 1:</u>

```
SQL> CONN SYSTEM /MANAGER;

SQL> LOCK TABLE EMP IN SHARE MODE;

SQL> COMMIT; [FOR RELEASING LOCKS]
```

<u>USER - 2:</u>

```
SQL> CONN SCOTT /TIGER;

SQL> SELECT * FROM EMP;

SQL> LOCK TABLE EMP IN SHARE MODE;
```

<u>EXCLUSIVE LOCK:</u> EXCLUSIVE LOCK WHEN A STATEMENT MODIFIES DATA. ITS TRANSACTION HOLDS AS EXCLUSIVE LOCK ON DATA THAT PREVENTS OTHER TRANSACTION FROM ACCESSING THE DATA AND ALSO HERE AT A TIME ONLY ONE USER LOCK THE RESOURCE.

SYNTAX:

```
LOCK TABLE <TN> IN EXCLUSIVE MODE;
```

Note: An Exclusive Lock allows only one session to read and write the data. No other user can read or write until the lock is released.

<u>EX:</u>

<u>USER - 1:</u>

```
SQL> CONN SYSTEM / MANAGER;

SQL> LOCK TABLE EMP IN EXCLUSIVE MODE;

SQL> COMMIT; [FOR RELEASING LOCKS]
```

USER - 2:

```
SQL> CONN SCOTT/TIGER;
SQL> SELECT * FROM EMP;
SQL> LOCK TABLE EMP IN EXCLUSIVE MODE;
        (OR)
SQL>LOCK TABLE EMP IN SHARE MODE;
```

[CANNOT PERFORM ANY LOCKS].