Experiment 3

Name: Bhagath Reddy Dandala **UID: 18BCS6196**

Branch: AIML2 **Section/ Group: 2 B**

Semester: 7th **Date Of Performance: 31/08/21**

Subject: ADBMS LAB **Subject Code: CSP-434**

1. Aim/Overview of the practical:

To analyze and create locks and different types of locks.

2. Task to be done:

To use create and implement various locks like Row Shared, Row Exclusive, Shared table, shared row exclusive, exclusive

3. Steps for experiment/practical:

1> Run SQL Command Line.
2> Create a table and insert the data into it.
3> Create various locks in between.
4> Print the table in between steps to see the working of various commands.

4. Observations/Discussions:

Row Share Table Locks (RS):

A row share table lock (also sometimes called a subshare table lock, SS) indicates that the transaction holding the lock on the table has locked rows in the table and intends to update them.

Row Exclusive Table Locks (RX):

A row exclusive table lock (also called a subexclusive table lock, SX) generally indicates that the transaction holding the lock has made one or more updates to rows in the table.

Share Table Locks (S):

A share table lock is acquired automatically for the table specified in the following statement:

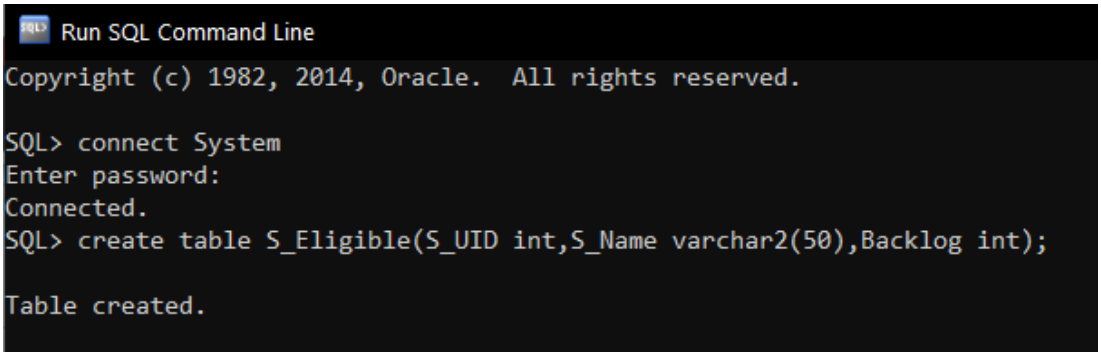
Share Row Exclusive Table Locks (SRX):

A share row exclusive table lock (also sometimes called a share-subexclusive table lock, SSX) is more restrictive than a share table lock.

Exclusive Table Locks (X):

An exclusive table lock is the most restrictive mode of table lock, allowing the transaction that holds the lock exclusive write access to the table. An exclusive table lock is acquired for a table as follows:

Result:



```
SQL> insert into S_Eligible values(6196,'Bhagath',0);

1 row created.

SQL> insert into S_Eligible values(3323,'Gannu',1);

1 row created.

SQL> select *from S_Eligible;

     S_UID S_NAME                                               BACKLOG
---------- -------------------------------------------------- ----------
      6196 Bhagath                                                    0
      3323 Gannu                                                      1

SQL> commit;

Commit complete.
```

```
SQL> Set transaction isolation level read committed;

Transaction set.

SQL> update S_Eligible set backlog=5 where S_UID=3323;

1 row updated.
```

```
SQL> select *from S_Eligible;

     S_UID S_NAME                                               BACKLOG
---------- -------------------------------------------------- ----------
      6196 Bhagath                                                    0
      3323 Gannu                                                      5

SQL> lock table S_Eligible in exclusive mode;

Table(s) Locked.

SQL> insert into S_Eligible values(1234,'harika',3);

1 row created.
```

```
SQL> update S_Eligible set Backlog=3 where S_Name='Gannu';

1 row updated.

SQL> commit;

Commit complete.

SQL> select *from S_Eligible;

     S_UID S_NAME                                              BACKLOG
---------- -------------------------------------------------- ----------
      6196 Bhagath                                                    0
      3323 Gannu                                                      3
      1234 harika                                                     3

SQL> lock table S_Eligible in share mode;

Table(s) Locked.

SQL> insert into S_Eligible values(3456,'kavya',5);

1 row created.

SQL> rollback;

Rollback complete.
```

```
SQL> select *from S_Eligible;

     S_UID S_NAME                                              BACKLOG
---------- -------------------------------------------------- ----------
      6196 Bhagath                                                    0
      3323 Gannu                                                      3
      1234 harika                                                     3

SQL>
```

5. Learning outcomes (What I have learnt):

a. Learnt about the various types of locks in SQL.

b. Learnt to work with command line SQL.

Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			