***Manipulating Data(DML)***

**Practice 8 Solutions**

Insert data into the MY\_EMPLOYEE table.

1. Run the statement in the lab8\_1.sql script to build the MY\_EMPLOYEE table that will be used for the lab.
2. **Ans:**

**CREATE TABLE my\_employee**

**(id NUMBER(4) CONSTRAINT my\_employee\_id\_nn NOT NULL,**

**last\_name VARCHAR2(25),**

**first\_name VARCHAR2(25),**

**userid VARCHAR2(8),**

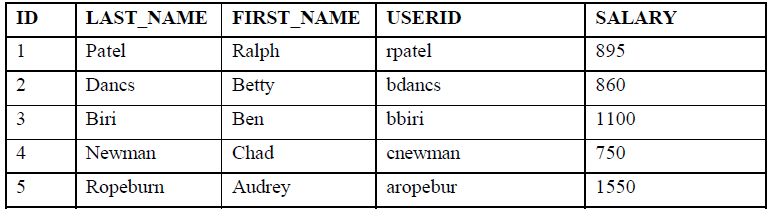
**salary NUMBER(9,2));**

1. Describe the structure of the MY\_EMPLOYEE table to identify the column names.

**Ans:**

**DESCRIBE my\_employee;**

3. Add the first row of data to the MY\_EMPLOYEE table from the following sample data. Do not list the columns in the INSERT clause.



**INSERT INTO my\_employee**

**VALUES (1, 'Patel', 'Ralph', 'rpatel', 895);**

1. Populate the MY\_EMPLOYEE table with the second row of sample data from the preceding list. This time, list the columns explicitly in the INSERT clause.

**INSERT INTO my\_employee (id, last\_name, first\_name,**

**userid, salary)**

**VALUES (2, 'Dancs', 'Betty', 'bdancs', 860);**

5. Confirm your addition to the table.

**Ans:**

**SELECT \* FROM my\_employee;**

6. Write an insert statement in a text file named loademp.sql to load rows into the

MY\_EMPLOYEE table. Concatenate the first letter of the first name and the first seven characters of

the last name to produce the userid.

**SET ECHO OFF**

**SET VERIFY OFF**

**INSERT INTO my\_employee**

**VALUES (&p\_id, '&p\_last\_name', '&p\_first\_name',**

**lower(substr('&p\_first\_name', 1, 1) ||**

**substr('&p\_last\_name', 1, 7)), &p\_salary);**

**SET VERIFY ON**

**SET ECHO ON**

7. Populate the table with the next two rows of sample data by running the insert statement in the

script that you created.

**SET ECHO OFF**

**SET VERIFY OFF**

**INSERT INTO my\_employee**

**VALUES (&p\_id, '&p\_last\_name', '&p\_first\_name',**

**lower(substr('&p\_first\_name', 1, 1) ||**

**substr('&p\_last\_name', 1, 7)), &p\_salary);**

**SET VERIFY ON**

**SET ECHO ON**

8. Confirm your additions to the table.

**SELECT \* FROM my\_employee;**

9. Make the data additions permanent.

**COMMIT;**

Update and delete data in the MY\_EMPLOYEE table.

10. Change the last name of employee 3 to Drexler.

**UPDATE my**\_**employee**

**SET last**\_**name = 'Drexler'**

**WHERE id = 3;**

11. Change the salary to 1000 for all employees with a salary less than 900.

**UPDATE my\_employee**

**SET salary = 1000**

**WHERE salary < 900;**

12. Verify your changes to the table.

**SELECT last**\_**name, salary**

**FROM my**\_**employee;**

13. Delete Betty Dancs from the MY\_EMPLOYEE table.

**DELETE**

**FROM my**\_**employee**

**WHERE last**\_**name = 'Dancs';**

14. Confirm your changes to the table.

**SELECT \***

**FROM my\_employee;**

15. Commit all pending changes.

**COMMIT;**

Control data transaction to the MY\_EMPLOYEE table.

16. Populate the table with the last row of sample data by modifying the statements in the script that you

created in step 6. Run the statements in the script.

**SET ECHO OFF**

**SET VERIFY OFF**

**INSERT INTO my\_employee**

**VALUES (&p\_id, '&p\_last\_name', '&p\_first\_name',**

**lower(substr('&p\_first\_name', 1, 1) ||**

**substr('&p\_last\_name', 1, 7)), &p\_salary);**

**SET VERIFY ON**

**SET ECHO ON**

17. Confirm your addition to the table.

**SELECT \* FROM my\_employee;**

18. Mark an intermediate point in the processing of the transaction.

**SAVEPOINT step\_18;**

19. Empty the entire table.

**DELETE FROM my\_employee;**

20. Confirm that the table is empty.

**SELECT \* FROM my\_employee;**

21. Discard the most recent DELETE operation without discarding the earlier INSERT operation.

**ROLLBACK TO step\_18;**

22. Confirm that the new row is still intact.

**SELECT \* FROM my**\_**employee;**

23. Make the data addition permanent.

**COMMIT;**