Experiment-4

Student Name: Gauri Prabhakar
UID: 18BCS6201
Branch: 18AITAIML-2
Section/Group: B
Semester: 7
Date of Performance: 8th September, 2021
Subject Name: Advanced Database Management Lab
Subject Code: CSP - 434

1. Aim/Overview of the practical:

To create and perform queries on sequences, synonyms and views.

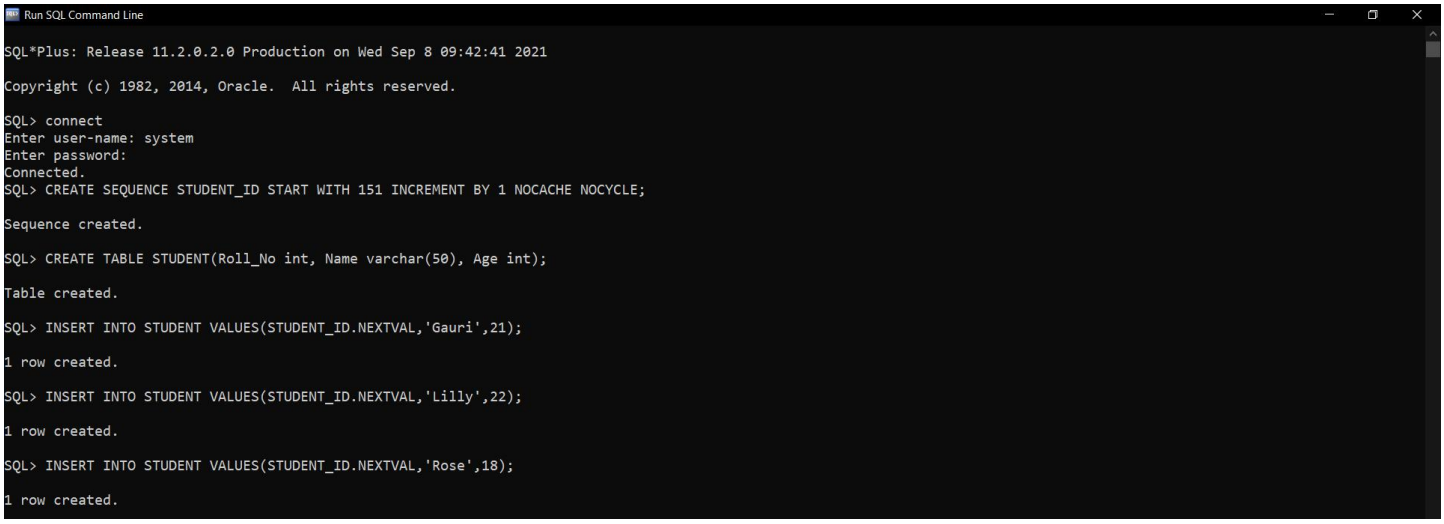
2. Task to be done:

To create and perform queries on sequences, synonyms and views.

3. Steps to be followed:

Creating a SEQUENCE, incrementing it by 1 then creating a table and inserting values into the table and setting NEXTVAL:

1. CREATE SEQUENCE STUDENT_ID START WITH 151 INCREMENT BY 1 NOCACHE NOCYCLE;
   CREATE TABLE STUDENT(Roll_No int, Name varchar(50), Age int);
   INSERT INTO STUDENT VALUES(STUDENT_ID.NEXTVAL,'Gauri',21);
   INSERT INTO STUDENT VALUES(STUDENT_ID.NEXTVAL,'Lilly',22);
   INSERT INTO STUDENT VALUES(STUDENT_ID.NEXTVAL,'Rose',18);



Returning the table and creating SYNONYM:

2. SELECT * FROM STUDENT;

```
SQL> SELECT * FROM STUDENT;

   ROLL_NO NAME                                               AGE
---------- -------------------------------------------------- ----------
       151 Gauri                                              21
       152 Lilly                                              22
       153 Rose                                               18

SQL> CREATE SYNONYM STUDENT_6201 for STUDENT;

Synonym created.
```

Returning the SYNONYM table and inserting value to the original table and then returning the original table:

3. CREATE SYNONYM STUDENT_6201 for STUDENT;
SELECT * FROM STUDENT_6201;
INSERT INTO STUDENT VALUES(STUDENT_ID.NEXTVAL,'Chelsea',24);
SELECT * FROM STUDENT;

```
Run SQL Command Line
Synonym created.

SQL> SELECT * FROM STUDENT_6201;

   ROLL_NO NAME                                               AGE
---------- -------------------------------------------------- ----------
       151 Gauri                                              21
       152 Lilly                                              22
       153 Rose                                               18

SQL> INSERT INTO STUDENT VALUES(STUDENT_ID.NEXTVAL,'Chelsea',24);

1 row created.

SQL> SELECT * FROM STUDENT;

   ROLL_NO NAME                                               AGE
---------- -------------------------------------------------- ----------
       151 Gauri                                              21
       152 Lilly                                              22
       153 Rose                                               18
       154 Chelsea                                            24
```

Creating a VIEW then returning it then UPDATING the VIEW:

4. CREATE VIEW STUDENT_VW AS SELECT Roll_No, Age FROM Student WHERE Age >= 20;
SELECT * FROM STUDENT_VW;
UPDATE STUDENT_VW SET Age = 28 WHERE Roll_No = 152;

```
SQL> CREATE VIEW STUDENT_VW AS SELECT Roll_No, Age FROM Student WHERE Age >= 20;

View created.

SQL> SELECT * FROM STUDENT_VW;

   ROLL_NO        AGE
---------- ----------
       151         21
       152         22
       154         24

SQL> UPDATE STUDENT_VW SET Age = 28 WHERE Roll_No = 152;

1 row updated.
```

Returning the updated VIEW, creating another VIEW from the previous VIEW, returning the new VIEW and then inserting values into the new VIEW and returning the updated view:

5. SELECT * FROM STUDENT_VW;
CREATE VIEW STUDENT_VEW AS SELECT Roll_No, Age FROM STUDENT_VW;
SELECT * FROM STUDENT_VEW;
INSERT INTO STUDENT_VEW VALUES (STUDENT_ID.NEXTVAL, 26);
SELECT * FROM STUDENT_VEW;

```
Run SQL Command Line

SQL> SELECT * FROM STUDENT_VW;

   ROLL_NO        AGE
---------- ----------
       151         21
       152         28
       154         24

SQL> CREATE VIEW  STUDENT_VEW AS SELECT Roll_No, Age FROM  STUDENT_VW;

View created.

SQL> SELECT * FROM STUDENT_VEW;

   ROLL_NO        AGE
---------- ----------
       151         21
       152         28
       154         24

SQL> INSERT INTO STUDENT_VEW VALUES (STUDENT_ID.NEXTVAL, 26);

1 row created.

SQL> SELECT * FROM STUDENT_VEW;

   ROLL_NO        AGE
---------- ----------
       151         21
       152         28
       154         24
       155         26
```

Returning the updated table and then DROPPING the view:

6. SELECT * FROM STUDENT;
DROP VIEW STUDENT_VW;

```
SQL> SELECT * FROM STUDENT;

   ROLL_NO NAME                                               AGE
---------- -------------------------------------------------- ----------
       151 Gauri                                              21
       152 Lilly                                              28
       153 Rose                                               18
       154 Chelsea                                            24
       155                                                    26

SQL> DROP VIEW STUDENT_VW;

View dropped.

SQL>
```

4. Result/Output/Writing Summary:

- Successfully implemented Sequence, Synonym and Views.
- Successfully understood the functioning and importance of the above mentioned.
- Successfully implemented operations on the above mentioned.
- Successfully understood the working of Sequence, Synonym and Views .

5. Learning outcomes (What I have learnt):

- How to implement Sequence, Synonym and Views on SQL Command Line.
- How to update Views and initialize Sequence.
- How to implement operations on Sequence, Synonym and Views and return their outcomes.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

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