Experiment-9

Student Name: Gauri Prabhakar	**UID:** 18BCS6201
Branch: 18AITAIML-2	**Section/Group:** B
Semester: 7	**Date of Performance:** 17h November, 2021
Subject Name: Advanced Database Management Lab	**Subject Code:** CSP - 434

1. Aim/Overview of the practical:

To implement a Case Study explaining the need for converting tables to fourth and fifth Normal forms.

2. Task to be done:

To implement a Case Study explaining the need for converting tables to fourth and fifth Normal forms.

3. What is 4NF and 5NF?

A table is said to be in 4NF iff:

- It is in BCNF.
- It is independent of many to one relationships and multi-valued attributes.

A table is said to be in 5NF iff:

- It is in 4NF.
- It is independent of any join dependency.
- It is broken into as many tables as possible.

4. Why 4NF?

- 4NF helps reduce redundancy.
- Helps avoid contradiction.

5. Why 5NF?

- 5NF helps reduce redundancy.
- Helps avoid update anomalies.

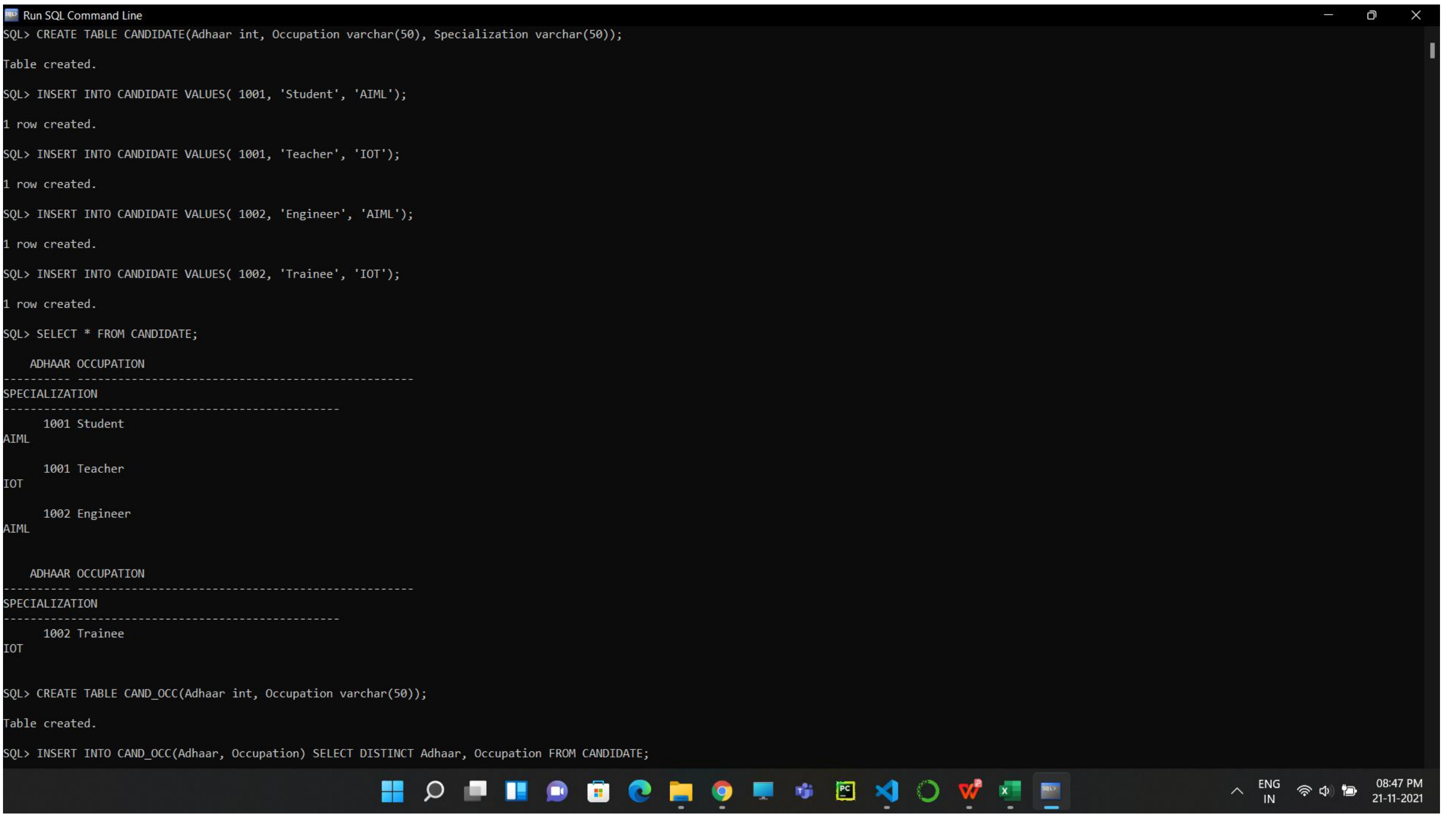
6. **Steps to be followed:**

4NF:

Creating a table CANDIDATE which does not follow 4NF norms and then returning it, then creating another table CAND_OCC to eliminate multi-valued attributes:

1. CREATE TABLE CANDIDATE(Adhaar int, Occupation varchar(50), Specialization varchar(50));
   INSERT INTO CANDIDATE VALUES(1001, 'Student', 'AIML');
   INSERT INTO CANDIDATE VALUES(1001, 'Teacher', 'IOT');
   INSERT INTO CANDIDATE VALUES(1002, 'Engineer', 'AIML');
   INSERT INTO CANDIDATE VALUES(1002, 'Trainee', 'IOT');
   SELECT * FROM CANDIDATE;

   CREATE TABLE CAND_OCC(Adhaar int, Occupation varchar(50));
   INSERT INTO CAND_OCC(Adhaar, Occupation) SELECT DISTINCT Adhaar, Occupation FROM CANDIDATE;

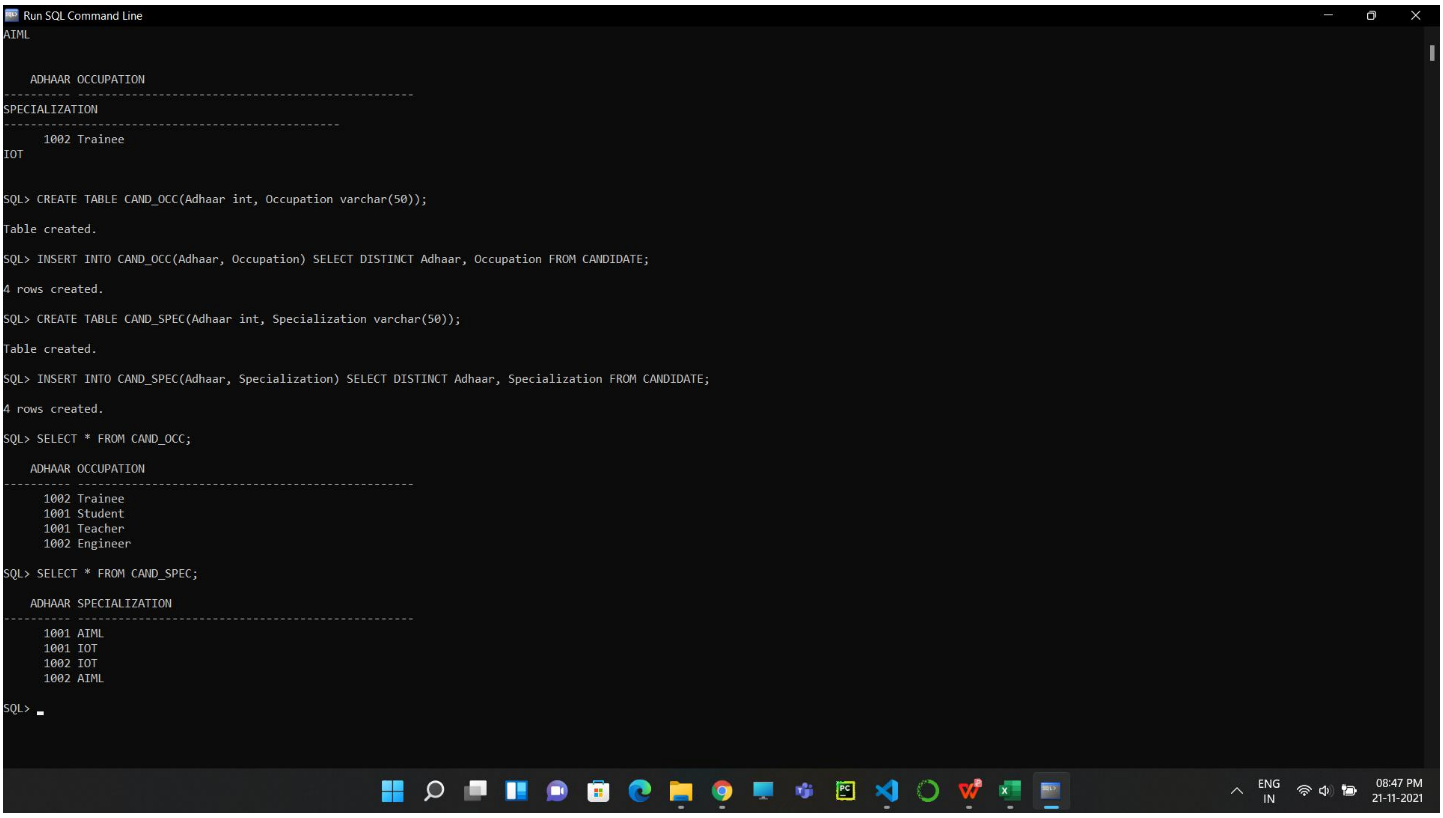


```
SQL> CREATE TABLE CANDIDATE(Adhaar int, Occupation varchar(50), Specialization varchar(50));

Table created.

SQL> INSERT INTO CANDIDATE VALUES( 1001, 'Student', 'AIML');

1 row created.

SQL> INSERT INTO CANDIDATE VALUES( 1001, 'Teacher', 'IOT');

1 row created.

SQL> INSERT INTO CANDIDATE VALUES( 1002, 'Engineer', 'AIML');

1 row created.

SQL> INSERT INTO CANDIDATE VALUES( 1002, 'Trainee', 'IOT');

1 row created.

SQL> SELECT * FROM CANDIDATE;

    ADHAAR OCCUPATION
---------- --------------------------------------------------
SPECIALIZATION
--------------------------------------------------
      1001 Student
AIML

      1001 Teacher
IOT

      1002 Engineer
AIML


    ADHAAR OCCUPATION
---------- --------------------------------------------------
SPECIALIZATION
--------------------------------------------------
      1002 Trainee
IOT


SQL> CREATE TABLE CAND_OCC(Adhaar int, Occupation varchar(50));

Table created.

SQL> INSERT INTO CAND_OCC(Adhaar, Occupation) SELECT DISTINCT Adhaar, Occupation FROM CANDIDATE;
```

Creating another table CAND_SPEC to eliminate multi-valued attributes then returning both the tables:

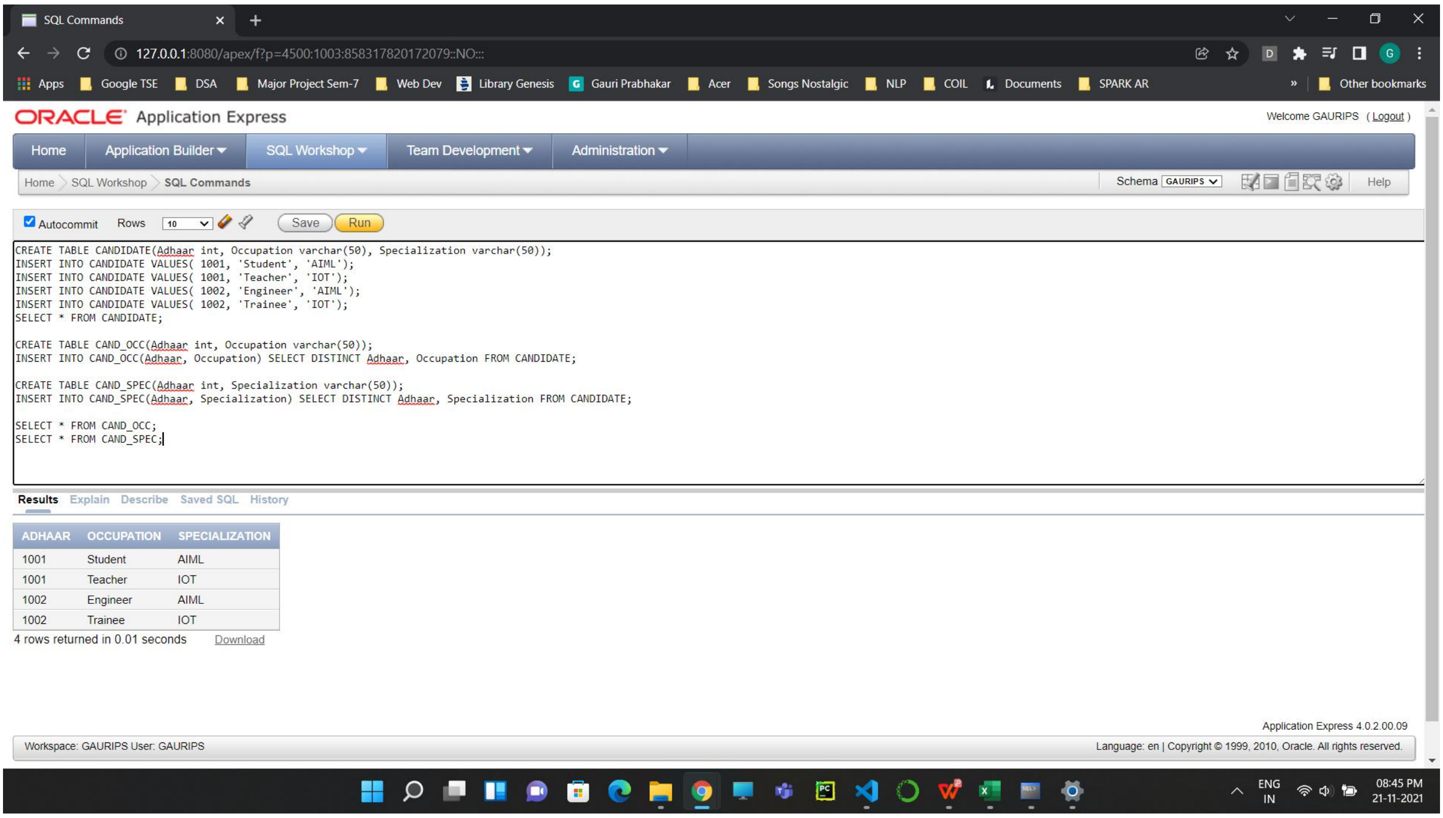
2. CREATE TABLE CAND_SPEC(Adhaar int, Specialization varchar(50));
INSERT INTO CAND_SPEC(Adhaar, Specialization) SELECT DISTINCT Adhaar, Specialization FROM CANDIDATE;

SELECT * FROM CAND_OCC;
SELECT * FROM CAND_SPEC;



```
AIML


    ADHAAR OCCUPATION
---------- --------------------------------------------------
SPECIALIZATION
--------------------------------------------------
      1002 Trainee
IOT


SQL> CREATE TABLE CAND_OCC(Adhaar int, Occupation varchar(50));

Table created.

SQL> INSERT INTO CAND_OCC(Adhaar, Occupation) SELECT DISTINCT Adhaar, Occupation FROM CANDIDATE;

4 rows created.

SQL> CREATE TABLE CAND_SPEC(Adhaar int, Specialization varchar(50));

Table created.

SQL> INSERT INTO CAND_SPEC(Adhaar, Specialization) SELECT DISTINCT Adhaar, Specialization FROM CANDIDATE;

4 rows created.

SQL> SELECT * FROM CAND_OCC;

    ADHAAR OCCUPATION
---------- --------------------------------------------------
      1002 Trainee
      1001 Student
      1001 Teacher
      1002 Engineer

SQL> SELECT * FROM CAND_SPEC;

    ADHAAR SPECIALIZATION
---------- --------------------------------------------------
      1001 AIML
      1001 IOT
      1002 IOT
      1002 AIML

SQL> _
```

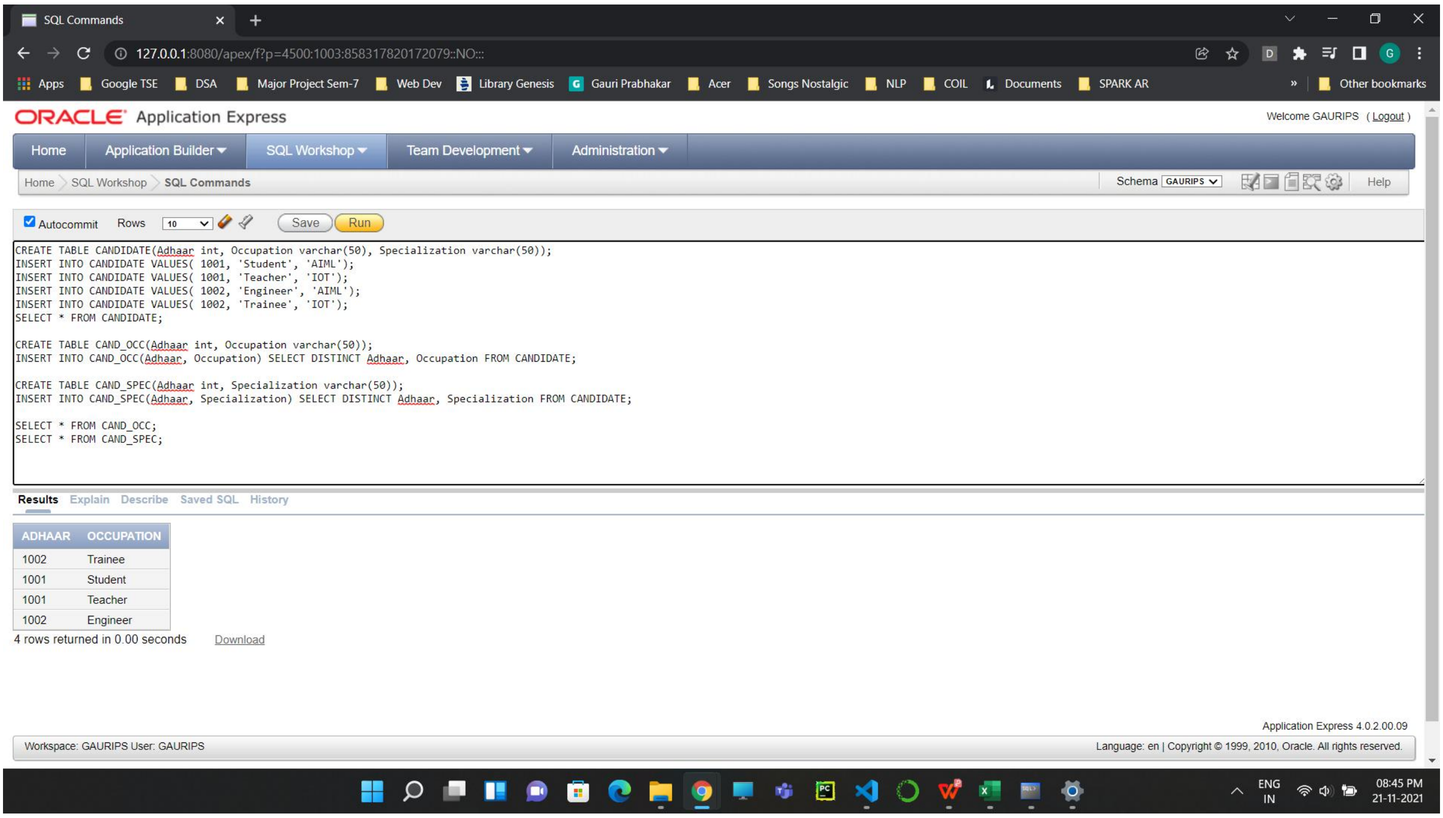
Output:


SQL Commands
127.0.0.1:8080/apex/f?p=4500:1003:858317820172079::NO:::
ORACLE Application Express
Welcome GAURIPS (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > SQL Workshop > SQL Commands
Schema GAURIPS
Help
☑ Autocommit Rows 10
Save
Run
Results Explain Describe Saved SQL History
Download
Application Express 4.0.2.00.09
Workspace: GAURIPS User: GAURIPS
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.
08:45 PM 21-11-2021

```
CREATE TABLE CANDIDATE(Adhaar int, Occupation varchar(50), Specialization varchar(50));
INSERT INTO CANDIDATE VALUES( 1001, 'Student', 'AIML');
INSERT INTO CANDIDATE VALUES( 1001, 'Teacher', 'IOT');
INSERT INTO CANDIDATE VALUES( 1002, 'Engineer', 'AIML');
INSERT INTO CANDIDATE VALUES( 1002, 'Trainee', 'IOT');
SELECT * FROM CANDIDATE;

CREATE TABLE CAND_OCC(Adhaar int, Occupation varchar(50));
INSERT INTO CAND_OCC(Adhaar, Occupation) SELECT DISTINCT Adhaar, Occupation FROM CANDIDATE;

CREATE TABLE CAND_SPEC(Adhaar int, Specialization varchar(50));
INSERT INTO CAND_SPEC(Adhaar, Specialization) SELECT DISTINCT Adhaar, Specialization FROM CANDIDATE;

SELECT * FROM CAND_OCC;
SELECT * FROM CAND_SPEC;
```

ADHAAR	OCCUPATION	SPECIALIZATION
1001	Student	AIML
1001	Teacher	IOT
1002	Engineer	AIML
1002	Trainee	IOT

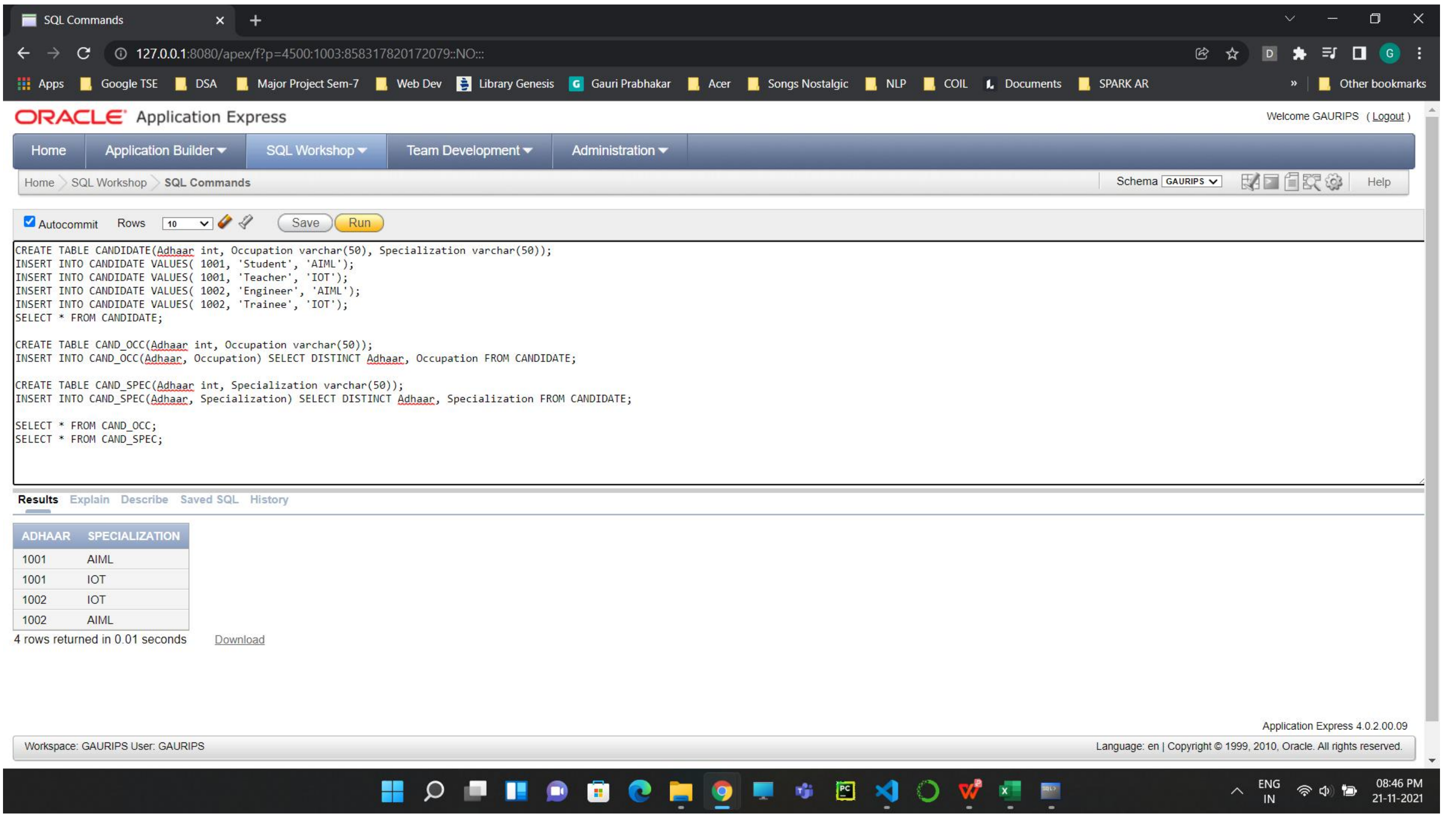
4 rows returned in 0.01 seconds


SQL Commands
127.0.0.1:8080/apex/f?p=4500:1003:858317820172079::NO:::
ORACLE Application Express
Welcome GAURIPS (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > SQL Workshop > SQL Commands
Schema GAURIPS
Help
☑ Autocommit Rows 10
Save
Run
Results Explain Describe Saved SQL History
Download
Application Express 4.0.2.00.09
Workspace: GAURIPS User: GAURIPS
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.
08:45 PM 21-11-2021

```
CREATE TABLE CANDIDATE(Adhaar int, Occupation varchar(50), Specialization varchar(50));
INSERT INTO CANDIDATE VALUES( 1001, 'Student', 'AIML');
INSERT INTO CANDIDATE VALUES( 1001, 'Teacher', 'IOT');
INSERT INTO CANDIDATE VALUES( 1002, 'Engineer', 'AIML');
INSERT INTO CANDIDATE VALUES( 1002, 'Trainee', 'IOT');
SELECT * FROM CANDIDATE;

CREATE TABLE CAND_OCC(Adhaar int, Occupation varchar(50));
INSERT INTO CAND_OCC(Adhaar, Occupation) SELECT DISTINCT Adhaar, Occupation FROM CANDIDATE;

CREATE TABLE CAND_SPEC(Adhaar int, Specialization varchar(50));
INSERT INTO CAND_SPEC(Adhaar, Specialization) SELECT DISTINCT Adhaar, Specialization FROM CANDIDATE;

SELECT * FROM CAND_OCC;
SELECT * FROM CAND_SPEC;
```

ADHAAR	OCCUPATION
1002	Trainee
1001	Student
1001	Teacher
1002	Engineer

4 rows returned in 0.00 seconds



☑ Autocommit Rows 10

```
CREATE TABLE CANDIDATE(Adhaar int, Occupation varchar(50), Specialization varchar(50));
INSERT INTO CANDIDATE VALUES( 1001, 'Student', 'AIML');
INSERT INTO CANDIDATE VALUES( 1001, 'Teacher', 'IOT');
INSERT INTO CANDIDATE VALUES( 1002, 'Engineer', 'AIML');
INSERT INTO CANDIDATE VALUES( 1002, 'Trainee', 'IOT');
SELECT * FROM CANDIDATE;

CREATE TABLE CAND_OCC(Adhaar int, Occupation varchar(50));
INSERT INTO CAND_OCC(Adhaar, Occupation) SELECT DISTINCT Adhaar, Occupation FROM CANDIDATE;

CREATE TABLE CAND_SPEC(Adhaar int, Specialization varchar(50));
INSERT INTO CAND_SPEC(Adhaar, Specialization) SELECT DISTINCT Adhaar, Specialization FROM CANDIDATE;

SELECT * FROM CAND_OCC;
SELECT * FROM CAND_SPEC;
```

ADHAAR	SPECIALIZATION
1001	AIML
1001	IOT
1002	IOT
1002	AIML

4 rows returned in 0.01 seconds

Workspace: GAURIPS User: GAURIPS

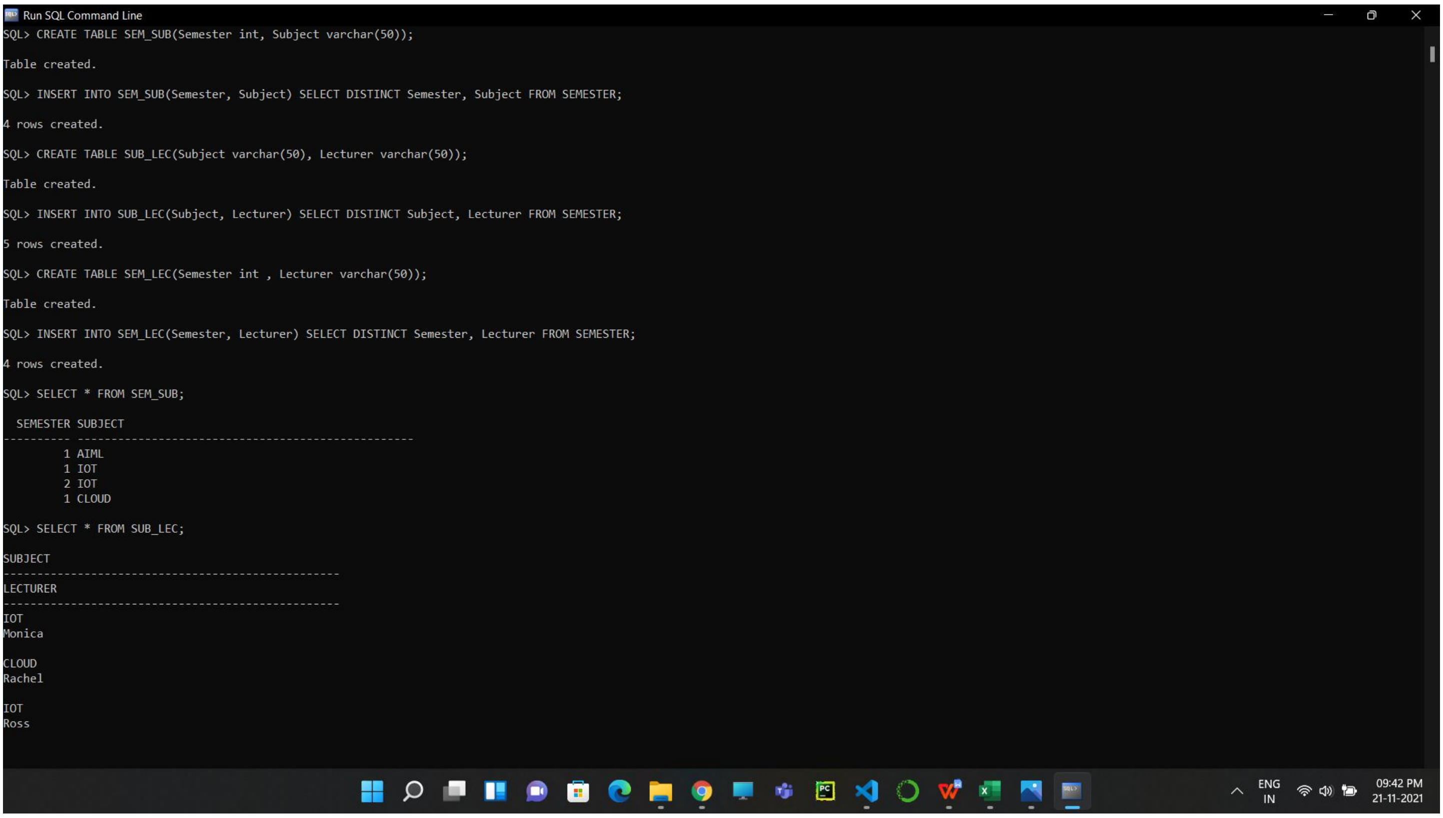
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.

Now the table is in 4NF.

5NF:

Creating table SEMESTER which does not follow 5NF and then returning it:

3. CREATE TABLE SEMESTER(Subject varchar(50), Lecturer varchar(50), Semester int);
   INSERT INTO SEMESTER VALUES('AIML', 'Chandler', 1);
   INSERT INTO SEMESTER VALUES('AIML', 'Ross', 1);
   INSERT INTO SEMESTER VALUES('IOT', 'Ross', 1);
   INSERT INTO SEMESTER VALUES('IOT', 'Monica', 2);
   INSERT INTO SEMESTER VALUES('CLOUD', 'Rachel', 1);
   SELECT * FROM SEMESTER;



```
SQL> CREATE TABLE SEM_SUB(Semester int, Subject varchar(50));

Table created.

SQL> INSERT INTO SEM_SUB(Semester, Subject) SELECT DISTINCT Semester, Subject FROM SEMESTER;

4 rows created.

SQL> CREATE TABLE SUB_LEC(Subject varchar(50), Lecturer varchar(50));

Table created.

SQL> INSERT INTO SUB_LEC(Subject, Lecturer) SELECT DISTINCT Subject, Lecturer FROM SEMESTER;

5 rows created.

SQL> CREATE TABLE SEM_LEC(Semester int , Lecturer varchar(50));

Table created.

SQL> INSERT INTO SEM_LEC(Semester, Lecturer) SELECT DISTINCT Semester, Lecturer FROM SEMESTER;

4 rows created.

SQL> SELECT * FROM SEM_SUB;

  SEMESTER SUBJECT
---------- --------------------------------------------------
         1 AIML
         1 IOT
         2 IOT
         1 CLOUD

SQL> SELECT * FROM SUB_LEC;

SUBJECT
--------------------------------------------------
LECTURER
--------------------------------------------------
IOT
Monica

CLOUD
Rachel

IOT
Ross
```

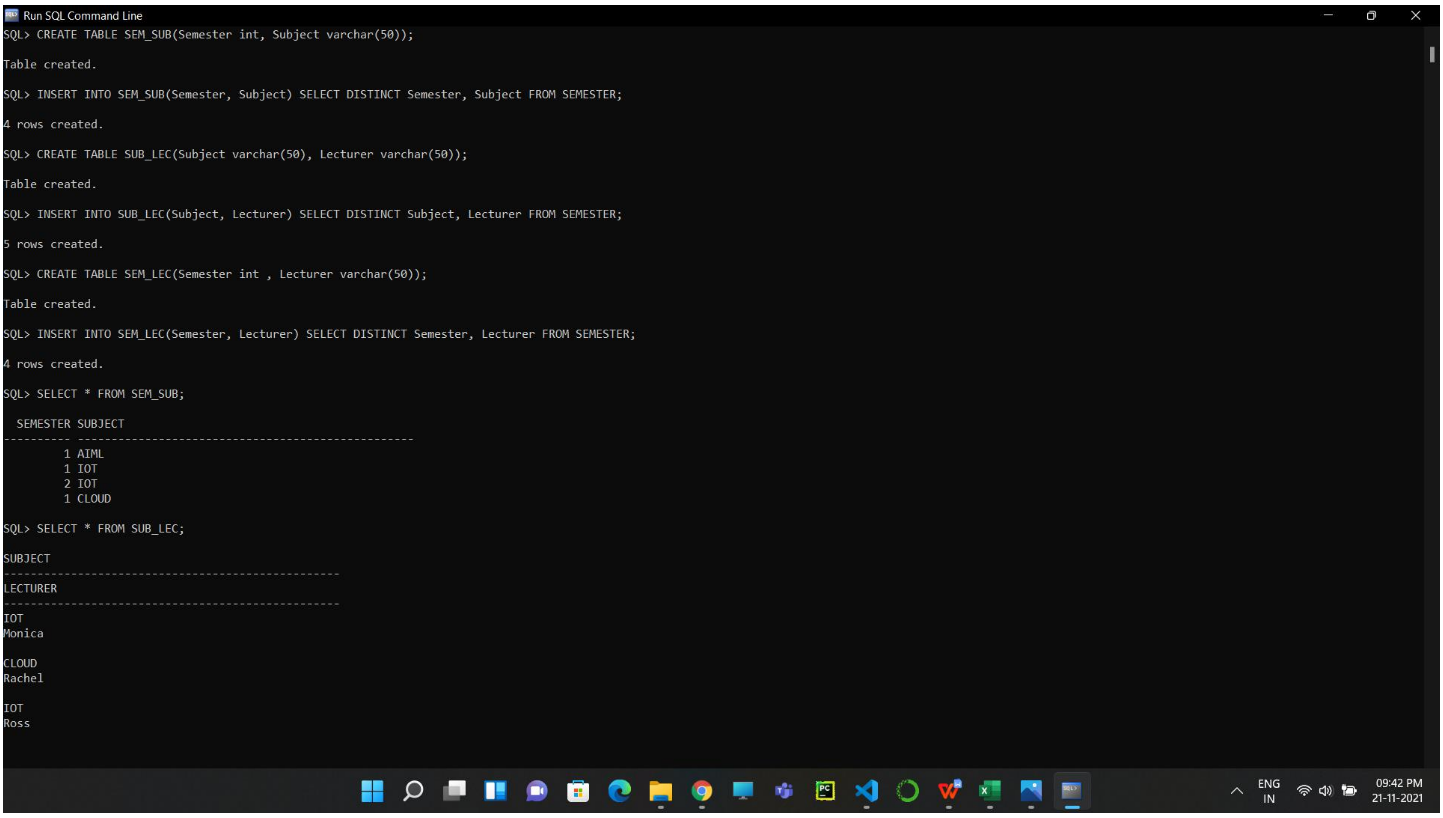
Creating 3 tables, SEM_SUB, SUB_LEC and SEM_LEC to convert the above table to 5NF and then returning them:

4. CREATE TABLE SEM_SUB(Semester int, Subject varchar(50));
   INSERT INTO SEM_SUB(Semester, Subject) SELECT DISTINCT Semester, Subject FROM SEMESTER;

   CREATE TABLE SUB_LEC(Subject varchar(50), Lecturer varchar(50));
   INSERT INTO SUB_LEC(Subject, Lecturer) SELECT DISTINCT Subject, Lecturer FROM SEMESTER;

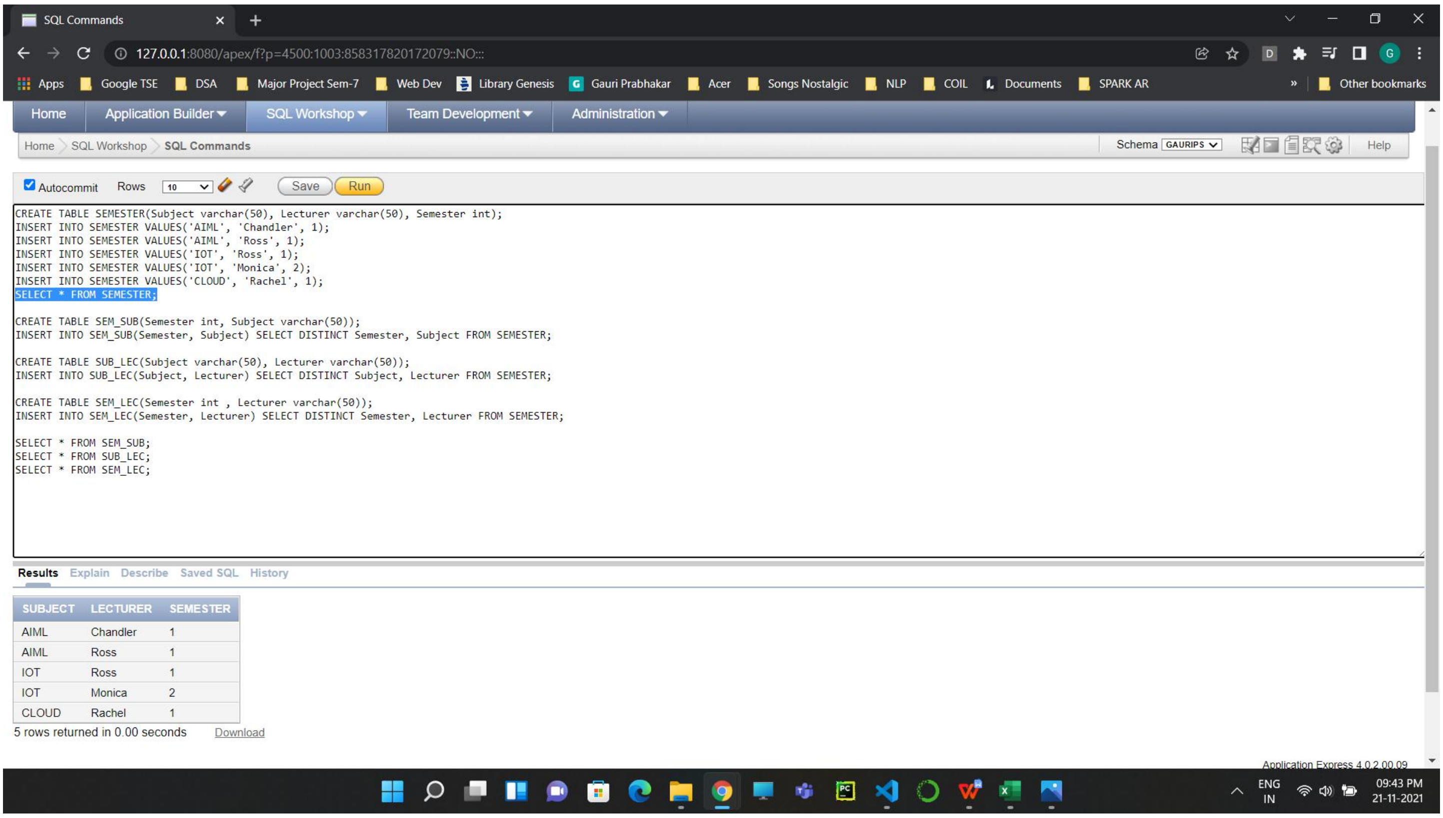
   CREATE TABLE SEM_LEC(Semester int , Lecturer varchar(50));
   INSERT INTO SEM_LEC(Semester, Lecturer) SELECT DISTINCT Semester, Lecturer FROM SEMESTER;

   SELECT * FROM SEM_SUB;
   SELECT * FROM SUB_LEC;
   SELECT * FROM SEM_LEC;



```
SQL> CREATE TABLE SEM_SUB(Semester int, Subject varchar(50));

Table created.

SQL> INSERT INTO SEM_SUB(Semester, Subject) SELECT DISTINCT Semester, Subject FROM SEMESTER;

4 rows created.

SQL> CREATE TABLE SUB_LEC(Subject varchar(50), Lecturer varchar(50));

Table created.

SQL> INSERT INTO SUB_LEC(Subject, Lecturer) SELECT DISTINCT Subject, Lecturer FROM SEMESTER;

5 rows created.

SQL> CREATE TABLE SEM_LEC(Semester int , Lecturer varchar(50));

Table created.

SQL> INSERT INTO SEM_LEC(Semester, Lecturer) SELECT DISTINCT Semester, Lecturer FROM SEMESTER;

4 rows created.

SQL> SELECT * FROM SEM_SUB;

  SEMESTER SUBJECT
---------- --------------------------------------------------
         1 AIML
         1 IOT
         2 IOT
         1 CLOUD

SQL> SELECT * FROM SUB_LEC;

SUBJECT
--------------------------------------------------
LECTURER
--------------------------------------------------
IOT
Monica

CLOUD
Rachel

IOT
Ross
```

Output:

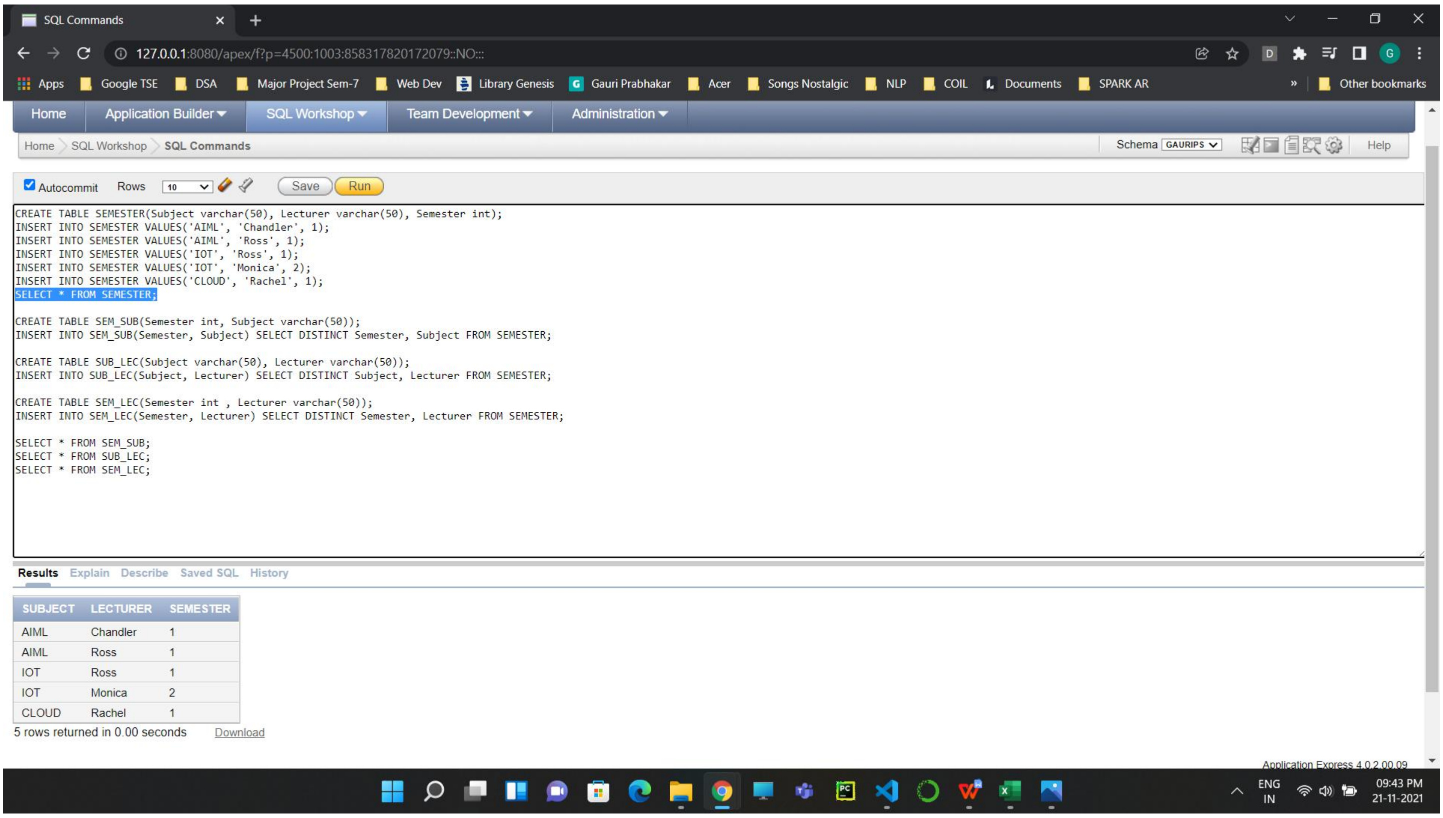


☑ Autocommit Rows 10 Save Run

```
CREATE TABLE SEMESTER(Subject varchar(50), Lecturer varchar(50), Semester int);
INSERT INTO SEMESTER VALUES('AIML', 'Chandler', 1);
INSERT INTO SEMESTER VALUES('AIML', 'Ross', 1);
INSERT INTO SEMESTER VALUES('IOT', 'Ross', 1);
INSERT INTO SEMESTER VALUES('IOT', 'Monica', 2);
INSERT INTO SEMESTER VALUES('CLOUD', 'Rachel', 1);
SELECT * FROM SEMESTER;

CREATE TABLE SEM_SUB(Semester int, Subject varchar(50));
INSERT INTO SEM_SUB(Semester, Subject) SELECT DISTINCT Semester, Subject FROM SEMESTER;

CREATE TABLE SUB_LEC(Subject varchar(50), Lecturer varchar(50));
INSERT INTO SUB_LEC(Subject, Lecturer) SELECT DISTINCT Subject, Lecturer FROM SEMESTER;

CREATE TABLE SEM_LEC(Semester int , Lecturer varchar(50));
INSERT INTO SEM_LEC(Semester, Lecturer) SELECT DISTINCT Semester, Lecturer FROM SEMESTER;

SELECT * FROM SEM_SUB;
SELECT * FROM SUB_LEC;
SELECT * FROM SEM_LEC;
```

SUBJECT	LECTURER	SEMESTER
AIML	Chandler	1
AIML	Ross	1
IOT	Ross	1
IOT	Monica	2
CLOUD	Rachel	1

5 rows returned in 0.00 seconds Download

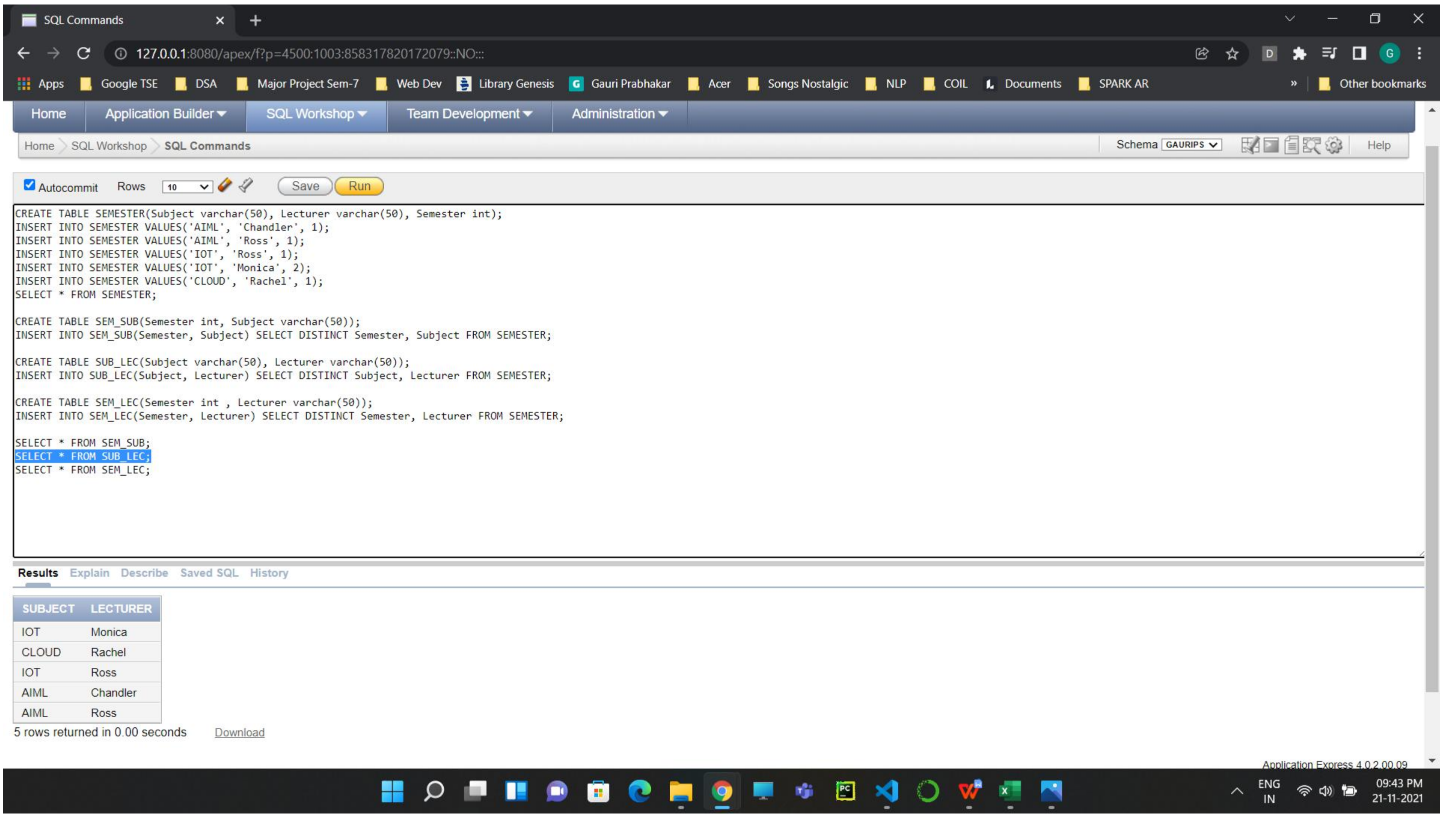


☑ Autocommit Rows 10 Save Run

```
CREATE TABLE SEMESTER(Subject varchar(50), Lecturer varchar(50), Semester int);
INSERT INTO SEMESTER VALUES('AIML', 'Chandler', 1);
INSERT INTO SEMESTER VALUES('AIML', 'Ross', 1);
INSERT INTO SEMESTER VALUES('IOT', 'Ross', 1);
INSERT INTO SEMESTER VALUES('IOT', 'Monica', 2);
INSERT INTO SEMESTER VALUES('CLOUD', 'Rachel', 1);
SELECT * FROM SEMESTER;

CREATE TABLE SEM_SUB(Semester int, Subject varchar(50));
INSERT INTO SEM_SUB(Semester, Subject) SELECT DISTINCT Semester, Subject FROM SEMESTER;

CREATE TABLE SUB_LEC(Subject varchar(50), Lecturer varchar(50));
INSERT INTO SUB_LEC(Subject, Lecturer) SELECT DISTINCT Subject, Lecturer FROM SEMESTER;

CREATE TABLE SEM_LEC(Semester int , Lecturer varchar(50));
INSERT INTO SEM_LEC(Semester, Lecturer) SELECT DISTINCT Semester, Lecturer FROM SEMESTER;

SELECT * FROM SEM_SUB;
SELECT * FROM SUB_LEC;
SELECT * FROM SEM_LEC;
```

SUBJECT	LECTURER	SEMESTER
AIML	Chandler	1
AIML	Ross	1
IOT	Ross	1
IOT	Monica	2
CLOUD	Rachel	1

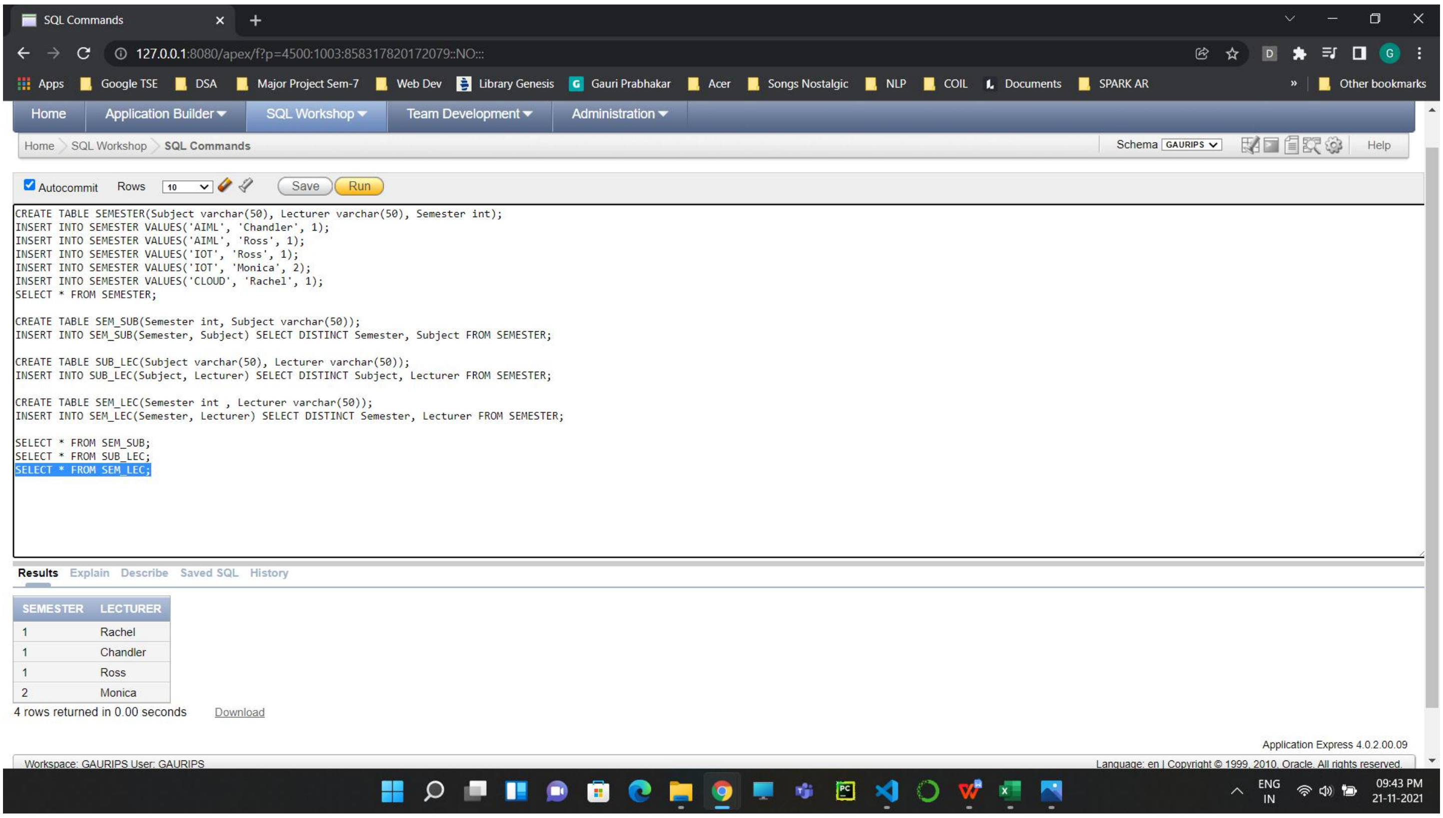
5 rows returned in 0.00 seconds Download


SQL Commands
127.0.0.1:8080/apex/f?p=4500:1003:858317820172079::NO:::
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > SQL Workshop > SQL Commands
Schema GAURIPS
Help
☑ Autocommit Rows 10
Save
Run
Results Explain Describe Saved SQL History
Download
Application Express 4.0.2.00.09
09:43 PM
21-11-2021

```
CREATE TABLE SEMESTER(Subject varchar(50), Lecturer varchar(50), Semester int);
INSERT INTO SEMESTER VALUES('AIML', 'Chandler', 1);
INSERT INTO SEMESTER VALUES('AIML', 'Ross', 1);
INSERT INTO SEMESTER VALUES('IOT', 'Ross', 1);
INSERT INTO SEMESTER VALUES('IOT', 'Monica', 2);
INSERT INTO SEMESTER VALUES('CLOUD', 'Rachel', 1);
SELECT * FROM SEMESTER;

CREATE TABLE SEM_SUB(Semester int, Subject varchar(50));
INSERT INTO SEM_SUB(Semester, Subject) SELECT DISTINCT Semester, Subject FROM SEMESTER;

CREATE TABLE SUB_LEC(Subject varchar(50), Lecturer varchar(50));
INSERT INTO SUB_LEC(Subject, Lecturer) SELECT DISTINCT Subject, Lecturer FROM SEMESTER;

CREATE TABLE SEM_LEC(Semester int , Lecturer varchar(50));
INSERT INTO SEM_LEC(Semester, Lecturer) SELECT DISTINCT Semester, Lecturer FROM SEMESTER;

SELECT * FROM SEM_SUB;
SELECT * FROM SUB_LEC;
SELECT * FROM SEM_LEC;
```

SUBJECT	LECTURER
IOT	Monica
CLOUD	Rachel
IOT	Ross
AIML	Chandler
AIML	Ross

5 rows returned in 0.00 seconds


SQL Commands
127.0.0.1:8080/apex/f?p=4500:1003:858317820172079::NO:::
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > SQL Workshop > SQL Commands
Schema GAURIPS
Help
☑ Autocommit Rows 10
Save
Run
Results Explain Describe Saved SQL History
Download
Application Express 4.0.2.00.09
Workspace: GAURIPS User: GAURIPS
Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.
09:43 PM
21-11-2021

```
CREATE TABLE SEMESTER(Subject varchar(50), Lecturer varchar(50), Semester int);
INSERT INTO SEMESTER VALUES('AIML', 'Chandler', 1);
INSERT INTO SEMESTER VALUES('AIML', 'Ross', 1);
INSERT INTO SEMESTER VALUES('IOT', 'Ross', 1);
INSERT INTO SEMESTER VALUES('IOT', 'Monica', 2);
INSERT INTO SEMESTER VALUES('CLOUD', 'Rachel', 1);
SELECT * FROM SEMESTER;

CREATE TABLE SEM_SUB(Semester int, Subject varchar(50));
INSERT INTO SEM_SUB(Semester, Subject) SELECT DISTINCT Semester, Subject FROM SEMESTER;

CREATE TABLE SUB_LEC(Subject varchar(50), Lecturer varchar(50));
INSERT INTO SUB_LEC(Subject, Lecturer) SELECT DISTINCT Subject, Lecturer FROM SEMESTER;

CREATE TABLE SEM_LEC(Semester int , Lecturer varchar(50));
INSERT INTO SEM_LEC(Semester, Lecturer) SELECT DISTINCT Semester, Lecturer FROM SEMESTER;

SELECT * FROM SEM_SUB;
SELECT * FROM SUB_LEC;
SELECT * FROM SEM_LEC;
```

SEMESTER	LECTURER
1	Rachel
1	Chandler
1	Ross
2	Monica

4 rows returned in 0.00 seconds

Now the table is in 5NF.

7. **Result/Output/Writing Summary:**

- Successfully implemented table conversion to 4NF.
- Successfully implemented table conversion to 5NF.
- Successfully implemented operations for NORMALIZATION.
- Successfully understood the functioning and importance of the above mentioned.

8. **Learning outcomes (What I have learnt):**

- How to implement table conversion to 4NF on SQL Command Line.
- How to implement table conversion to 5NF on SQL Command Line.
- How to implement NORMALIZATION on a table.

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.			