LAB 02

CSE 4308
Database Management Systems Lab

Hasin Mahtab Alvee
210042174
Department of CSE
B.Sc in Software Engineering

August 20, 2023

Contents

Introduction

In DBMS lab 02, we were tasked to create a table and insert the data of Instructors in the table as defined. Once all commands were done, we were told to run the .sql file as a script in the SQL Terminal to execute the code in file.

1 Create and Connecting User

To create a table and insert data, first we need to create an user and connect to the Oracle database through that user. We also need to provide the user with all privileges and then we connect to the user using username and password. To run the file as a script we just need to put @ sign and then specify the SQL file location.

```
CREATE USER C_210042174 IDENTIFIED BY cse4308;
GRANT ALL PRIVILEGES TO C_210042174;
CONNECT C_210042174/cse4308;

SHOW USER;
```

1.1 Difficulties

While creating the user, some minor inconvenience were faced, such as -

- The instructions specified using a non-numeric symbol at the beginning of username, if it starts with a numeric symbol, it doesn't create any user.

2 Creating the Table

Once the user is connected, we can create the structure of table and specify the constraints that we need to apply. We need to set the ID as a primary key, which means we don't need to add any NOT-NULL constraints to ID. We can add this to the Name, and we also must set the VARCHAR length. We also want to check for the salary before inserting it into the table whether it is more than 20000 or not.

```
CREATE TABLE INSTRUCTOR (
ID VARCHAR(20),
NAME VARCHAR(100) NOT NULL,
DEPT_NAME VARCHAR(100),
SALARY INT,
CONSTRAINT PK_ID PRIMARY KEY (ID),
```

```
CONSTRAINT SALARY_CHECK CHECK (SALARY > 20000)
);
```

2.1 Difficulties

While creating the table structure, some minor inconvenience were faced, such as -

- The ID being used as a Primary key, it does not need to be specified as a NOT-NULL column.
- The ID was changed to VARCHAR from INT, as we needed to show the leading zeros in the ID.

3 Inserting data into Table

Once we are all set, we can now insert the data to table. We use INSERT to add the data to table. Oracle does not take multi-line input so we must use single line Insert for every data.

```
INSERT INTO INSTRUCTOR VALUES ('10101','Srinivasan','Comp. Sci.',65000),
INSERT INTO INSTRUCTOR VALUES ('12121', 'Wu', 'Finance', 90000);
INSERT INTO INSTRUCTOR VALUES ('15151', 'Mozart', 'Music', 40000);
INSERT INTO INSTRUCTOR VALUES ('22222', 'Einstein', 'Physics', 95000);
INSERT INTO INSTRUCTOR VALUES ('32343', 'El Said', 'History', 60000);
INSERT INTO INSTRUCTOR VALUES ('00456', 'Gold', 'Physics', 87000);
INSERT INTO INSTRUCTOR VALUES ('45565', 'Katz', 'Comp. Sci.', 75000);
INSERT INTO INSTRUCTOR VALUES ('58583', 'Califieri', 'History', 62000);
INSERT INTO INSTRUCTOR VALUES ('76543', 'Singh', 'Finance', 80000);
INSERT INTO INSTRUCTOR VALUES ('76766', 'Crick', 'Biology', 72000);
INSERT INTO INSTRUCTOR VALUES ('03821', 'Brandt', 'Comp. Sci.', 92000);
INSERT INTO INSTRUCTOR VALUES ('98345', 'Kim', 'Elec. Eng.', 80000);
```

3.1 Difficulties

While inserting the data, some minor inconvenience were faced, such as -

- Oracle does not accept multi-line inserting, so we had to insert each data line by line.
- We had to make sure the ID were being inserted as STRING and not INT.

4 Showing the Data

Once the data is inserted, we can simply show the data however we want. Here, different filters are used to show different kinds of data from the table.

```
SELECT * FROM INSTRUCTOR;

SELECT ID, NAME FROM INSTRUCTOR;

SELECT NAME, DEPT_NAME FROM INSTRUCTOR WHERE SALARY > 70000;

SELECT NAME, DEPT_NAME FROM INSTRUCTOR WHERE SALARY >= 80000 AND SALARY
    <= 100000;

SELECT ID, NAME FROM INSTRUCTOR WHERE DEPT_NAME = 'Comp. Sci.';

SELECT NAME, SALARY FROM INSTRUCTOR WHERE DEPT_NAME = 'Finance';

SELECT ID, NAME FROM INSTRUCTOR WHERE DEPT_NAME = 'Comp. Sci.' OR SALARY
    > 75000;

SELECT DEPT_NAME FROM INSTRUCTOR;
```

4.1 Difficulties

While creating the table structure, some minor inconvenience were faced, such as -

- I had to make sure the salary was being inclusive for one of the filters.

5 Dropping the Table

Finally, we need to add the command to drop the table when we don't need it any longer. But we need to specify the constraints to be removed as well, otherwise the constraints remain there.

```
DROP TABLE INSTRUCTOR CASCADE CONSTRAINTS;
```