Database Management Systems Lab
Lab 2
CSE 4308

A Z Hasnain Kabir
200042102
Software Engineering

4 September 2022

Introduction

SQL stands for Structured Query Language. It is used to manage data stored in a Relational Database Management System. It is useful in handling structured data. Structured data incorporates relations among entities and variables. In this lab we are supposed to familiarise ourselves with Oracle XE and understand how to create database and tables.

Task 1

Task: Create a user with user_name = ¡student_id¿ and password = cse4308 and grant necessary privileges to that user.
Analysis of the problem: We need to write some codes in our SQL command line to create a user.

Code:

```
CREATE USER Nibir IDENTIFIED BY nibir ;
```

Explanation of solution: We direct our database system Oracle XE to create a user called Nibir which is identified by nibir.

Findings: We have to log in to our system to make sure the user is created.

Task 2

Task: Write SQL statement to create a table 'INSTRUCTOR' which has 4 attributes. The attributes are:

- ID
- NAME
- DEPT_NAME
- Salary

Analysis of the problem: CREATE syntax in SQL will generate a table according to our given attributes and attribute types.

Code:

```
CREATE table INSTRUCTOR(
    ID INT NOT NULL,
    NAME VARCHAR2(20),
    DEPT_NAME VARCHAR2(20),
    SALARY INT
);
```

Explanation of solution: The CREATE syntax generates a table called instructor for us and the syntaxes in the brackets define the names of the attributes that should be present in that table.

Task 3

Task: Write SQL statements to insert given records into 'INSTRUCTOR' table:
Analysis of the problem: INSERT INTO syntax will insert rows inside previously created SQL table. We need to provide the values we want to insert under each attribute of the table.

Code:

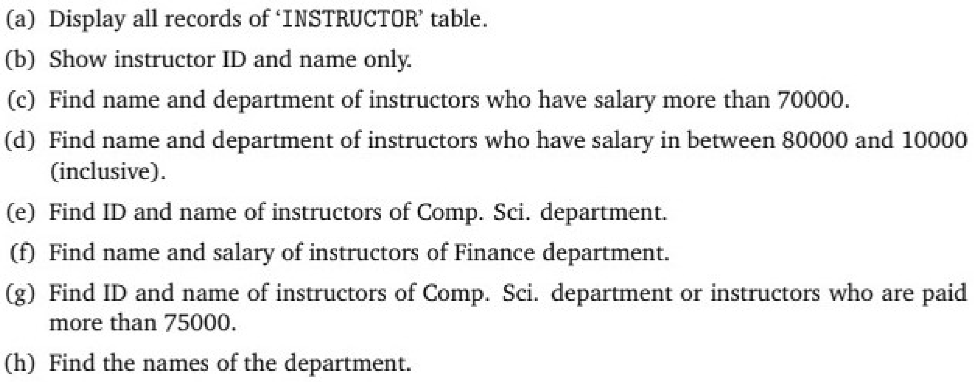
```
INSERT INTO INSTRUCTOR VALUES (10101, 'Srinivasan', 'Comp. Sci.', 65000);
INSERT INTO INSTRUCTOR VALUES (12121, 'Wu', 'Finance', 90000);
INSERT INTO INSTRUCTOR VALUES (15151, 'Mozart', 'Music', 40000);
INSERT INTO INSTRUCTOR VALUES (22222, 'Einstein', 'Physics', 95000);
INSERT INTO INSTRUCTOR VALUES (32343, 'El Said', 'History', 60000);
INSERT INTO INSTRUCTOR VALUES (33456, '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 (83821, 'Brandt', 'Comp. Sci.', 92000);
INSERT INTO INSTRUCTOR VALUES (98345, 'Kim', 'Elec. Eng.', 80000);
```

Explanation of solution: As we can see in the above code, we are inserting rows of information inside our previously created table. The syntaxes here are self explanatory and therefore requires little explanation.

Findings: We need to make sure we don't keep the value of ID empty for each row we are inserting.

Task 4

Task: Write SQL statements to perform given queries:



(a) Display all records of 'INSTRUCTOR' table.

(b) Show instructor ID and name only.

(c) Find name and department of instructors who have salary more than 70000.

(d) Find name and department of instructors who have salary in between 80000 and 10000 (inclusive).

(e) Find ID and name of instructors of Comp. Sci. department.

(f) Find name and salary of instructors of Finance department.

(g) Find ID and name of instructors of Comp. Sci. department or instructors who are paid more than 75000.

(h) Find the names of the department.

Analysis of the problem: We can use SELECT command for the given queries to find or display anything inside our table

(a)

Code:

```
SELECT * FROM INSTRUCTOR;
```

(b)

Code:

```
SELECT ID, NAME FROM INSTRUCTOR;
```

(c)

Code:

```
SELECT NAME, DEPT_NAME FROM INSTRUCTOR WHERE SALARY > 70000;
```

(d)

Code:

```
SELECT NAME, DEPT_NAME FROM INSTRUCTOR WHERE SALARY > 10000 AND SALARY <80000;
```

(e)

Code:

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

(f)

Code:

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

(g)

Code:

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

(h)

Code:

```
SELECT DISTINCT(DEPT_NAME) FROM INSTRUCTOR;
```

Explanation of solution: The SELECT command selects any information needed on the table that is specified using FROM command. WHERE command can determine which statements we are selecting and * command selects all records from any table.

Findings: We can select multiple attributes from each table and the statement after a WHERE command can work like a boolean expression.