CSC 352 / 452: Database Programming

assignment #1 (60 Points)

**CSC 352/452-501: Due on Sunday, 7/23/2017 at 11:59PM**

**CSC 352/452-510: Due on Monday, 7/24/2017 at 11:59PM**

Unless prior arrangements are made, homework turned in late will not be accepted. However, homework turned in within 24 hours late will be graded at 50% credit.

**Please note that only TEXT files will be accepted. All other file types (e.g., DOC, DOCX, RTF, PDF, JPG, or ZIP) will be rejected. In D2L, only the most recent submission is kept.**

**Part I (No need to submit)**

1) Download and install Oracle SQL Developer on your machine.

2) Create a connection from your SQL Developer to our database server.

3) Create table DEPARTMENT as described below.

CREATE TABLE department

( DEPARTMENT\_ID NUMBER(4) PRIMARY KEY,

DEPARTMENT\_NAME VARCHAR2(20) NOT NULL UNIQUE,

LOCATION VARCHAR2(20) NOT NULL);

4) Populate the DEPARTMENT table.

INSERT INTO department VALUES(10, 'ACCOUNTING', 'NEW YORK');

INSERT INTO department VALUES(20, 'RESEARCH', 'DALLAS');

INSERT INTO department VALUES(30, 'SALES', 'CHICAGO');

INSERT INTO department VALUES(40, 'IT', 'DALLAS');

INSERT INTO department VALUES(50, 'EXECUTIVE', 'NEW YORK');

INSERT INTO department VALUES(60, 'MARKETING', 'CHICAGO');

COMMIT;

5) Please make sure that there are 6 rows in your DEPARTMENT table.

SELECT \* FROM department;

6) Create table EMPLOYEE as described below.

CREATE TABLE employee

( EMPLOYEE\_ID NUMBER(4) PRIMARY KEY,

EMPLOYEE\_NAME VARCHAR2(20) NOT NULL,

JOB\_TITLE VARCHAR2(50) NOT NULL,

MANAGER\_ID NUMBER(4)

REFERENCES employee(EMPLOYEE\_ID) ON DELETE SET NULL,

HIRE\_DATE DATE NOT NULL,

SALARY NUMBER(9, 2) NOT NULL,

COMMISSION NUMBER(9, 2),

DEPARTMENT\_ID NUMBER(4) REFERENCES department(DEPARTMENT\_ID));

7) Populate the EMPLOYEE table.

INSERT INTO employee

VALUES(7839, 'KING', 'PRESIDENT', NULL, '20-NOV-01', 5000, NULL, 50);

INSERT INTO employee

VALUES(7596, 'JOST', 'VICE PRESIDENT', 7839, '04-MAY-01', 4500, NULL, 50);

INSERT INTO employee

VALUES(7603, 'CLARK', 'VICE PRESIDENT', 7839, '12-JUN-01', 4000, NULL, 50);

INSERT INTO employee

VALUES(7566, 'JONES', 'CHIEF ACCOUNTANT', 7596, '05-APR-01', 3000, NULL, 10);

INSERT INTO employee

VALUES(7886, 'STEEL', 'PUBLIC ACCOUNTANT', 7566, '08-MAR-03', 2500, NULL, 10);

INSERT INTO employee

VALUES(7610, 'WILSON', 'BUSINESS ANALYST', 7596, '03-DEC-01', 3000, NULL, 20);

INSERT INTO employee

VALUES(7999, 'WOLFE', 'TEST ANALYST', 7610, '15-FEB-02', 2500, NULL, 20);

INSERT INTO employee

VALUES(7944, 'LEE', 'REPORTING ANALYST', 7610, '04-SEP-06', 2400, NULL, 20);

INSERT INTO employee

VALUES(7900, 'FISHER', 'SALES EXECUTIVE', 7603, '06-DEC-01', 3000, 500, 30);

INSERT INTO employee

VALUES(7921, 'JACKSON', 'SALES REPRESENTATIVE', 7900, '25-FEB-05', 2500, 400, 30);

INSERT INTO employee

VALUES(7952, 'LANCASTER', 'SALES CONSULTANT', 7900, '06-DEC-06', 2000, 150, 30);

INSERT INTO employee

VALUES(7910, 'SMITH', 'DATABASE ADMINISTRATOR', 7596, '20-DEC-01', 2900, NULL, 40);

INSERT INTO employee

VALUES(7788, 'SCOTT', 'PROGRAMMER', 7910, '15-JAN-03', 2500, NULL, 40);

INSERT INTO employee

VALUES(7876, 'ADAMS', 'PROGRAMMER', 7910, '15-JAN-03', 2000, NULL, 40);

INSERT INTO employee

VALUES(7934, 'MILLER', 'PROGRAMMER', 7876, '25-JAN-02', 1000, NULL, 40);

INSERT INTO employee

VALUES(8000, 'BREWSTER', 'TBA', NULL, '22-AUG-13', 2500, NULL, NULL);

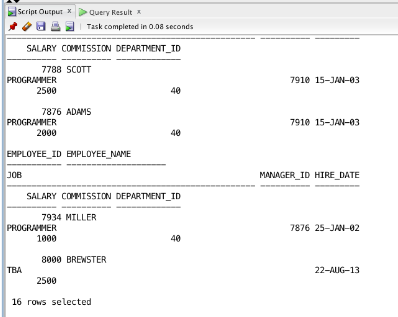
COMMIT;

8) Please make sure that there are 16 rows in your EMPLOYEE table.

SELECT \* FROM employee;

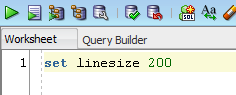
9) Q&A

Q: When I run the script and view the content on the Script Output screen, the text wraps. Do you know if there is a way to fix this?



A: Your SQL Developer does not set the linesize to unlimited. The users need to set the desired linesize by themselves. You may run

set linesize 200



Or download/use SQL Developer 3.2.09.30 from <https://ipdapp.cdm.depaul.edu/sqldeveloper/>

**Part II (60 points)**

**Your SQL statements can only reference the DEPARTMENT table and/or EMPLOYEE table. You are not allowed to create/access other tables/views. You will receive 0 points if you access different tables or columns (e.g., dept, emp, job, sal, deptid …).**

1) (CSC 352 - 20 points | CSC 452 – 15 points)

Write a SQL SELECT statement to display the employee ID, name, job title, and hire date for all employees along with their managers’ names, job titles, and hire dates. Make sure that employees without managers are included as well.

* If an employee does not have a manager, the manager’s name is shown as “------”, the manager’s job title is shown as “------”, and the manager’s hire date is shown as 31-DEC-9999 in your output.
* The hire date must be displayed in the **DD-MON-YYYY** format (4-digit year).
* You cannot use hard-coded employee names (e.g., WHERE employee\_name = 'KING') in your SELECT statement.
* Sort your output in ascending order by employee ID.
* You will receive 0 points if you submit more than one SELECT statement.

Hints: 1) You may need to use an OUTER JOIN and a SELF-JOIN.

2) NVL(TO\_CHAR(column\_x, 'DD-MON-YYYY'), '31-DEC-9999')

Your statement’s output should match the following format:

**EMPLOYEE ID EMPLOYEE NAME EMPLOYEE JOB TITLE EMPLOYEE HIRE DATE MANAGER NAME MANAGER JOB TITLE MANAGER HIRE DATE**

**----------- -------------------- -------------------------------------------------- ------------------ -------------------- -------------------------------------------------- -----------------**

**7566 JONES CHIEF ACCOUNTANT 05-APR-2001 JOST VICE PRESIDENT 04-MAY-2001**

**7596 JOST VICE PRESIDENT 04-MAY-2001 KING PRESIDENT 20-NOV-2001**

**7603 CLARK VICE PRESIDENT 12-JUN-2001 KING PRESIDENT 20-NOV-2001**

**7610 WILSON BUSINESS ANALYST 03-DEC-2001 JOST VICE PRESIDENT 04-MAY-2001**

**7788 SCOTT PROGRAMMER 15-JAN-2003 SMITH DATABASE ADMINISTRATOR 20-DEC-2001**

**7839 KING PRESIDENT 20-NOV-2001 ------ ------ 31-DEC-9999**

**……**

*(This question should take less than 20 minutes to solve.)*

2) (CSC 352 - 20 points | CSC 452 – 15 points)

Write a SQL SELECT statement to display 1) all department names, 2) the maximum **total pay** (salary + commission) for each department, 3) the minimum hire date for each department, and 4) the total number of employees in each department.

* Any employee who does not belong to any department is excluded from your output.
* You must display the maximum total pay with a dollar ($) sign, a comma, and two decimal places (e.g., $1,234.56). If a department does not have any employee, the maximum total pay is shown as $0.00.
* The minimum hire date must be displayed in the **DD-MON-YYYY** format (4-digit year). If a department does not have any employee, the minimum hire date is shown as 31-DEC-9999.
* Sort your output in ascending order by department name.
* You will receive 0 points if you submit more than one SELECT statement.

Hint: You may need to use an OUTER JOIN.

Your statement’s output should match the following format:

**DEPARTMENT NAME MAXIMUM TOTAL PAY MINIMUM HIRE DATE TOTAL NUMBER OF EMPLOYEES**

**-------------------- ----------------- ----------------- -------------------------**

**ACCOUNTING $3,000.00 05-APR-2001 2**

**EXECUTIVE $5,000.00 04-MAY-2001 3**

**IT $2,900.00 20-DEC-2001 4**

**MARKETING $0.00 31-DEC-9999 0**

**RESEARCH $3,000.00 03-DEC-2001 3**

**SALES $3,500.00 06-DEC-2001 3**

*(This question should take less than 20 minutes to solve.)*

3) (CSC 352 - 20 points | CSC 452 – 15 points)

Write aSQL SELECT statement to find out the **most recently** hired employees in **each department**. Your SELECT statement must display the department ID, department name, employee ID, employee name, job title, and hire date. Any employee who does not belong to any department is excluded from your output.

* The hire date must be displayed in the **DD-MON-YYYY** format (4-digit year).
* You cannot use hard-coded values (e.g., WHERE department\_id = 10) in your SELECT statement.
* Sort your output in ascending order by department name and then employee name.
* You will receive 0 points if you submit more than one SELECT statement.

Hint: A subquery may be needed in your SELECT statement (The examples on pages 27-28 may help

you).

Your statement’s output should match the following format:

**DEPARTMENT ID DEPARTMENT NAME EMPLOYEE ID EMPLOYEE NAME JOB TITLE HIRE DATE**

**------------- -------------------- ----------- -------------------- -------------------------------------------------- -----------**

**10 ACCOUNTING 7886 STEEL PUBLIC ACCOUNTANT 08-MAR-2003**

**……**

*(This question should take less than 20 minutes to solve.)*

4) (**CSC 452 only** – 15 points)

Write a SQL SELECT statement to display the **name** and location of all departments (except the departments located in **Dallas**) with the highest number of employees.

* **You cannot use join operations in your SELECT statement.**
* Hard coding, except the string 'DALLAS', is not allowed in your SELECT statement.
* Sort your output in ascending order by department name.
* You will receive 0 points if you submit more than one SELECT statement.

Hint:

|  |  |  |  |
| --- | --- | --- | --- |
| **Department Name** | **Location** | **# of Employees** | **Meet the search criteria?** |
| ACCOUNTING | NEW YORK | 2 | No |
| EXECUTIVE | NEW YORK | 3 | Yes |
| IT | DALLAS | 4 | No |
| MARKETING | CHICAGO | 0 | No |
| RESEARCH | DALLAS | 3 | No |
| SALES | CHICAGO | 3 | Yes |

Your statement’s output should match the following format:

**DEPARTMENT NAME LOCATION**

**-------------------- --------------------**

**EXECUTIVE NEW YORK**

**……**

*(This question should take less than 30 minutes to solve.)*

**Please submit a text file (your\_name\_hw1.txt) containing all the source codes (Part II) to D2L by the due date.**

Example: your\_name\_hw1.txt

================

Part II

================

1)

SELECT … FROM …;

2)

SELECT … FROM …;

……