CSC 352 / 452: Database Programming

assignment #1 (60 Points)

**CSC 352/452-501: Due on Sunday, 7/26/2015 at 11:59PM**

**CSC 352/452-510: Due on Monday, 7/27/2015 at 11:59PM**

Unless prior arrangements are made, homework turned in late but within 24 hours of the due time will be graded at 75% credit, homework turned in between 24 and 48 hours will be graded at 50% credit, and homework turned in later than 48 hours will not be accepted.

**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 (5 points, 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,

ADDRESS 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) Create table EMPLOYEE as described below.

CREATE TABLE employee

( EMPLOYEE\_ID NUMBER(4) PRIMARY KEY,

EMPLOYEE\_NAME VARCHAR2(20) NOT NULL,

JOB VARCHAR2(50) NOT NULL,

MANAGER\_ID NUMBER(4),

HIRE\_DATE DATE NOT NULL,

SALARY NUMBER(9, 2) NOT NULL,

COMMISSION NUMBER(9, 2),

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

6) 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', 'PUBLIC 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', 'ANALYST', 7596, '03-DEC-01', 3000, NULL, 20);

INSERT INTO employee

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

INSERT INTO employee

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

INSERT INTO employee

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

INSERT INTO employee

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

INSERT INTO employee

VALUES(7952, 'LANCASTER', 'SALESMAN', 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;

**Part II (35 points)**

**Your SQL statements can only reference the DEPARTMENT table and/or EMPLOYEE table. You are not allowed to create/access other tables/views.**

1) (CSC 352 - 20 points | CSC 452 – 10 points)

Write a SQL SELECT statement to display all jobs, the minimum **total pay** (salary + commission) for each job, and the total number of employees in each job. You must display the minimum total pay with a dollar ($) sign, a comma, and two decimal places (e.g., $1,234.56). (You will lose 5 points if you fail to do so.) Sort your output in ascending order by job. (Submitting more than one SQL statement will receive 0 points.)

The output of your statement must match the following:

JOB MINIMUM TOTAL PAY TOTAL NUMBER OF EMPLOYEES

-------------------------------------------------- ----------------- -------------------------

ANALYST $2,400.00 3

DATABASE ADMINISTRATOR $2,900.00 1

PRESIDENT $5,000.00 1

PROGRAMMER $1,000.00 3

PUBLIC ACCOUNTANT $2,500.00 2

SALESMAN $2,150.00 3

TBA $2,500.00 1

VICE PRESIDENT $4,000.00 2

2) (CSC 352 - 20 points | CSC 452 – 10 points)

Write a SQL SELECT statement to display name, hire date, and salary for all employees along with their managers’ names 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 “\*\*\*\*\*\*” and the manager’s hire date is shown as 01-JAN-3000 in your output. You must display the salary with a dollar ($) sign, a comma, and two decimal places (e.g., $1,234.56). Sort your output in ascending order by employee name.

(Submitting more than one SQL statement will receive 0 points.)

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

2) NVL(TO\_CHAR(column\_x, 'DD-MON-YYYY'), '01-JAN-3000')

You cannot use hard-coded employee names (e.g., WHERE employee\_name = 'KING') in your programs.

The output of your statement must match the following:

EMPLOYEE NAME EMPLOYEE SALARY EMPLOYEE HIRE DATE MANAGER NAME MANAGER HIRE DATE

-------------------- --------------- ------------------ -------------------- -----------------

ADAMS $2,000.00 15-JAN-2003 SMITH 20-DEC-2001

BREWSTER $2,500.00 22-AUG-2013 \*\*\*\*\*\* 01-JAN-3000

CLARK $4,000.00 12-JUN-2001 KING 20-NOV-2001

FISHER $3,000.00 06-DEC-2001 CLARK 12-JUN-2001

JACKSON $2,500.00 25-FEB-2005 FISHER 06-DEC-2001

JONES $3,000.00 05-APR-2001 JOST 04-MAY-2001

JOST $4,500.00 04-MAY-2001 KING 20-NOV-2001

KING $5,000.00 20-NOV-2001 \*\*\*\*\*\* 01-JAN-3000

LANCASTER $2,000.00 06-DEC-2006 FISHER 06-DEC-2001

LEE $2,400.00 04-SEP-2006 WILSON 03-DEC-2001

MILLER $1,000.00 25-JAN-2002 ADAMS 15-JAN-2003

SCOTT $2,500.00 15-JAN-2003 SMITH 20-DEC-2001

SMITH $2,900.00 20-DEC-2001 JOST 04-MAY-2001

STEEL $2,500.00 08-MAR-2003 JONES 05-APR-2001

WILSON $3,000.00 03-DEC-2001 JOST 04-MAY-2001

WOLFE $2,500.00 15-FEB-2002 WILSON 03-DEC-2001

3) (CSC 452 only – 20 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, and hire date. Any employee who does not belong to any department is excluded from your output. Sort your output in ascending order by department name and then employee name. (Submitting more than one SQL statement will receive 0 points.)

Hint: A subquery may be needed in your SELECT statement.

The output of your statement must match the following:

DEPARTMENT ID DEPARTMENT NAME EMPLOYEE ID EMPLOYEE NAME JOB HIRE\_DATE

------------- -------------------- ----------- -------------------- -------------------------------------------------- ---------

10 ACCOUNTING 7886 STEEL PUBLIC ACCOUNTANT 08-MAR-03

50 EXECUTIVE 7839 KING PRESIDENT 20-NOV-01

40 IT 7876 ADAMS PROGRAMMER 15-JAN-03

40 IT 7788 SCOTT PROGRAMMER 15-JAN-03

20 RESEARCH 7944 LEE ANALYST 04-SEP-06

30 SALES 7952 LANCASTER SALESMAN 06-DEC-06

**Part III (20 points)**

The table GROCERY\_PRICE shows some grocery prices for the year 2012 and estimates of grocery prices for the year 2022. Create and populate the table GROCERY\_PRICE as described below.

CREATE TABLE grocery\_price

( GROCERY\_ID VARCHAR2(30) PRIMARY KEY,

PRICE\_IN\_2012 NUMBER(6, 2),

PRICE\_IN\_2022 NUMBER(6, 2));

INSERT INTO grocery\_price VALUES('E\_001', 0.62, 1.78);

INSERT INTO grocery\_price VALUES('B\_001', 0.80, 3.28);

INSERT INTO grocery\_price VALUES('B\_002', 2.72, 7.36);

INSERT INTO grocery\_price VALUES('M\_004', 2.70, 5.65);

INSERT INTO grocery\_price VALUES('T\_006', 5.70, 6.65);

INSERT INTO grocery\_price VALUES('R\_003', 4.00, 16.40);

INSERT INTO grocery\_price VALUES('G\_010', 8.00, 32.80);

COMMIT;

**Your SQL statement can only reference the GROCERY\_PRICE table. You are not allowed to create/access other tables/views. You will get a zero point if you use a different table/column name.**

(20 Points) Write a SQL SELECT statement that finds the item(s) having **the highest percentage** of price increase. You must display each price with a dollar ($) sign and two decimal places (e.g., **$0.80**). Sort your output in ascending order by GROCERY\_ID. (Submitting more than one SQL statement will receive 0 points.)

Hard coding (e.g., WHERE …… = **310, rownum < 4**) is not allowed in your programs.

The output of your statement must match the following:

GROCERY\_ID PRICE\_IN\_2012 PRICE\_IN\_2022 HIGHEST\_PERCENTAGE\_INCREASE

------------------------------ ------------- ------------- -----------------------------------------

B\_001 $0.80 $3.28 310%

G\_010 $8.00 $32.80 310%

R\_003 $4.00 $16.40 310%

Hints: 1) ((PRICE\_IN\_2022 – PRICE\_IN\_2012) / PRICE\_IN\_2012) \* 100

2) A subquery may be needed in your SELECT statement. The example on page 26 may help you.

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

Example: your\_name\_hw1.txt

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

Part II

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

1)

SELECT … FROM …;

2)

SELECT … FROM …;

……

**Optional Question**

**Just for fun (no credit, no extra credit, no need to submit, just for if you are a curious person and like database programming).**

Write a SQL SELECT statement to display names and salaries for all employees and indicate the amount of their salaries through asterisks. Each asterisk signifies a hundred dollars. Sort the data in descending order of salary.

EMPLOYEE\_AND\_SALARY

--------------------------------------------------------------------------------------

KING ($5000) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

JOST ($4500) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CLARK ($4000) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

JONES ($3000) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WILSON ($3000) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FISHER ($3000) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SMITH ($2900) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WOLFE ($2500) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

BREWSTER ($2500) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SCOTT ($2500) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

JACKSON ($2500) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STEEL ($2500) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LEE ($2400) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ADAMS ($2000) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LANCASTER ($2000) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MILLER ($1000) \*\*\*\*\*\*\*\*\*\*