**CSC 450/550 Database Management Systems**

**Assignment 2 (125 points)**

**Problem 1) SQL: Storing Employee Information, Warm-up (16 points)** This question is on an employee salaries database.

* 1. First, use the Import Data utility in Oracle SQL Developer to load the small sample database from the *StoringEmployeeInformation\_Data.csv* file (right click over Tables, choose Import Data from the drop-down menu). Import it in Oracle as an *Employees* table. The structure of the data in the csv file includes the following attributes: Full Name, Current annual salary, Gross pay received, Overtime pay, Department (This is the abbreviation of department name; for example BOA stands for Board of Appeals,…), Division, Assignment Category (Parttime-Regular, Fulltime-Regular), Position Title (Supervisor, Manager, Social worker, Library assistant,…) and Date First Hired. During the import steps, make sure you set the columns to appropriate types – pay special attention to the date column, and choose the appropriate date format.

As a sanity check that you have the correct table, running the command *select count(\*) from Employees;* should return 40.

Next, write SQL queries to do the following:

* 1. Query 1: Return the names of all part-time employees. *Hint:* you have to look at the Assignment Category column.
  2. Query 2: Return the total number of departments in the database.

**Turn in:**

* (5 points) The SQL query for Query 1.

select Full\_Name

from A2\_EMPLOYEES\_SAMPLE

where ASSIGNMENT\_CATEGORY like 'Parttime-Regular';

* (3 points) The output of running Query 1 in Oracle on the sample database.

Berlin Susan S

Lane Myles B.

Marroquin Carlos

Mars Frandi M.

* (5 points) The SQL query for Query 2.

select count(DISTINCT department)

from A2\_EMPLOYEES\_SAMPLE;

* (3 points) The output of running Query 2 in Oracle on the sample database.

13

**Problem 2) SQL: Storing Employee Information, Larger CSV file (23 points)**

* 1. A bigger raw comma separated value (csv) file is given on Blackboard: *employee-sample.csv*. It has a similar structure to the sample in Problem 1. Import it in Oracle. Again, as a sanity check, your correct table should have 1871 records. Next, write queries to do the following:
  2. Query 3: List all people who work in the same department as “Adcock Sr Gerald W”- himself excluded.
  3. Query 4: List all the departments and total number of employees working in that department.

**Turn in:**

* (10 points) The SQL query for Query 3.

Select Full\_Name

from A2\_EMPLOYEES

where DEPARTMENT like (

Select DEPARTMENT

From A2\_EMPLOYEES

where FULL\_NAME like 'Adcock Sr Gerald W' );

* (3 points) The output of running Query 3 on the csv file after loading it in Oracle.

Colbert James M

Forney Bryan W.

Fouant Zelinda Joan

Adcock Sr Gerald W

Johnson Bobby R

Johnson Jared

Johnson Richard

Jones David E

Harper Todd S

Plucinski Karen J

Kline Victoria

Klinger Dieter

Mack Patricia D

Malcom Betty L.

Miller Judith M

Milo Deborah L

Moeis Deri N

Pappachan Shoge

Tang Karen N.

Tao Pea-Fen W.

Taormino Timothy P

Tarquinio Michael C

Varkey Samir J

Vellaiappan Balamurugan

Vernoy Gregory L

Rogers Richard L

Segal Harash N

Selbst Barbara A

Sessa Louis J

Shang Catherine L

* (7 points) The SQL query for Query 4.

Select department, count(department) as num\_employees

from A2\_EMPLOYEES

group by DEPARTMENT;

* (3 points) The output of running Query 4 on the csv file after loading it in Oracle.

CAT 17

HCA 17

OHR 16

DLC 77

SHF 40

CEC 13

CCL 21

OMB 2

DHS 2

OCP 5

OLO 1

DTS 30

CUS 8

LIB 70

BOE 6

FRS 283

COR 102

DPS 28

BOA 1

MCERP 3

HHS 316

DGS 97

FIN 28

PIO 10

OIG 2

CEX 4

REC 24

POL 352

DOT 251

DEP 33

DED 12

**Problem 3) Life without SQL. (24 points)** Finally, write code in either Java or C++ to do both queries above (Query 3 and Query 4) on the csv data file directly.

**Turn in:**

* (10 points) Your Java/C++ code for doing Query 3 on the raw csv file directly.
* (10 points) Your Java/C++ code for doing Query 4 on the raw csv file directly.
* (4 points) A *readme* file to explain how to run your code.

**Problem 4) SQL: The School DB** **(62 points)**

The schema of the database for this problem is provided below (keys are underlined, field types are omitted):

dept(dname, numphds)

course(cno, cname, dname)

prof(pname, dname)

section(cno, dname, sectno, pname)

student(sid, sname, sex, age, gpa)

major(sid, dname)

enroll(sid, cno, dname, sectno, grade)

A script is provided too, to create and populate these tables (file *UniversityDB.sql*).

In this problem, you will only deal with the querying part of SQL. This means you are NOT allowed to tamper with (change the contents of) the database, i.e., CREATE, INSERT, DELETE, ALTER, UPDATE etc.

Write SQL queries that answer the questions below (one query per question). The query answers must not contain duplicates, but you should use the SQL keyword distinct only when necessary. For this question, creation of temporary tables is NOT allowed, i.e., for each question you have to write exactly one SQL statement (possible using nested SQL).

* 1. (7 points) Find the names and gpas of the students who are enrolled in 351.

Note: there might be multiple courses sharing the same number, offered by different departments; students enrolled in all 351 courses should be selected, regardless of the department offering it.

select distinct student.SNAME,student.GPA

from enroll

join STUDENT on STUDENT.SID = enroll.SID

where cno = 351;

Brandt 4

Ellison 3

Sanchez 3

Bourikas 3.75

Shankar 3.5

Allen 4

* 1. (8 points) Find the name of the oldest student(s).

select sname

from STUDENT

where age = (

select max(age)

from student);

Tanaka

* 1. (10 points) Find the names and majors of students who are taking one of the World History courses.

select distinct student.SNAME,major.DNAME

from enroll

inner join STUDENT on STUDENT.SID = enroll.SID

inner join course on course.CNO = enroll.CNO

inner join major on major.SID = STUDENT.SID

where course.CNAME like 'World History%';

Shankar Comp. Sci.

Walton History

Brandt History

Ellison History

Bourikas Elec. Eng.

Allen History

Sanchez Music

* 1. (8 points) Find the names of students who are enrolled in a course from both the Computer Sciences and Electrical Engineering departments.

select sname

from student

where sid in (

select sid

from enroll

where dname like 'Comp. Sci.')

and sid in (

select sid

from enroll

where dname like 'Elec. Eng.');

Brown

* 1. (10 points) How many students have more than one major? (*Hint*: requires a nested query)

select count(\*)

from student

where sid in (

select sid

from major

group by sid

having count(dname) > 1);

2

* 1. (8 points) Find the name(s) of the oldest male student. (*Hint*: requires a nested query)

select sname

from student

where sex like 'M' and age = (

select max(age)

from student

where sex like 'M');

Levy

* 1. (11 points) Find the ids(s) and name(s) of the students who are not taking any course offered by any of the department(s) of which they are a major. (Hint: requires a nested query)

select sname,sid

from student

where sid not in (

select enroll.sid

from enroll

inner join major on major.sid = enroll.sid

where major.dname = enroll.dname);

Snow 70557

Bourikas 98765

**Turn in, for each query:**

* (max points – 2 points) The SQL query.
* (2 points) The output of running the query on the university database after creating it in Oracle.