CS352 - DATABASE SYSTEMS

Homework #2 - Due: Monday, February 13, at the start of class

Emphasis: Basic SQL

Introduction to SQL Homework Problems

As noted in the syllabus, most of the homework sets in the course will include some problems designed to help you develop facility with Structured Query Language (SQL).

Many of the problems will involve a database called `sample` that is distributed with DB2. The tables in this database correspond to the attached ER diagram. The commands used to create this database and the complete contents of each table are listed in the Appendix of the Birchall book. *Spend some time familiarizing yourself with the structure of this database before attempting the homework problems that pertain to it!* You can do these problems either on the workstations or by installing a personal copy of DB2-Express-C (the free version of DB2 IBM now makes available.) See the discussion below regarding each of these approaches.

You will be expected to dig out the information needed to answer these problems from the following sources. Since they are .pdf files, you will probably want to download them onto your computer for easy access. (Some suggestions as to where to look may be included in the homework) Generally speaking, you will find the Birchall book to be the best place to start; however, he does not cover all of the features of SQL.) They are accessible from the Blackboard site for the course.

Birchall, Graeme. *DB2 LUW V9.7 Cookbook*.

IBM. *IBM DB2 Universal Database SQL Reference* (2 volumes)

For these problems, you will be expected to turn in a printout of the query (neatly formatted using multiple lines and indentation) and the results generated by executing the query, created by selecting an appropriate portion of the screen and printing it out. For example, if the question were "what is all the information stored in the org table", the following printout would suffice. *Manually verify the output printed by db2 against the table contents in Birchall to be sure you've gotten the right answer - this becomes increasingly important as the assigned queries become more complex!*

```
select *
    from org;

DEPTNUMB DEPTNAME       MANAGER DIVISION   LOCATION
-------- -------------- ------- ---------- -------------
      10 Head Office        160 Corporate  New York
      15 New England         50 Eastern    Boston
      20 Mid Atlantic        10 Eastern    Washington
      38 South Atlantic      30 Eastern    Atlanta
      42 Great Lakes        100 Midwest    Chicago
      51 Plains             140 Midwest    Dallas
      66 Pacific            270 Western    San Francisco
      84 Mountain           290 Western    Denver
```

Abbreviated Usernames for DB2

DB2 limits user id's to 8 characters, and in one case only 7 characters are used. Since your linux username is usually of the form *firstname.lastname*, you will be given a special abbreviated alias for work with DB2 - just your firstname. This alias will access the same linux account as your full username (and you can use either when logging in to linux), but you must use the abbreviated alias whenever you interact with DB2 (Your password is initially the same for this new account as for your regular account; but if you change passwords you must do so for each account separately.)

Running db2 on Your Own Computer

These problems are designed to be done by accessing db2 from one of the workstations. Of course, this can be done either from 244 or by connecting to a workstation `ssh`, since db2's SQL interface is text-based. If you wish, you might try installing db2 on your own computer and doing the problems that way. I haven't checked the details out, so while the following procedure should work, it is not guaranteed.

1. Obtain a copy of the free express edition of db2 from `https://www.ibm.com/developerworks/downloads/im/db2express` and install it on your computer following IBM's directions.
2. To access the any of the databases used for the problems you will need to do the following once on your computer:
   a. Start db2 from the db2 administrative account on your computer.
   b. Issue the following command once
      `catalog tcpip node joshua remote joshua.cs.gordon.edu server 50000/tcpip`
   c. Issue the following command for each database you want to access. (You don't have to do all databases up front - you can do this whenever you need to start using a database:
      `catalog database` *whatever* `at node joshua.`
   d. Be sure to use your shott Linux username and password when connecting to a database.

Starting Working with the sample Database using db2

1. Login to a workstation using your regular username.
2. Be sure you have run the db2 profile either by having included it in your `.bash_profile` file or by executing it interactively:

   `. ~db2clien/sqllib/db2profile` (Note the "." at the start of the line!)

3. Start db2 using the following command

   `db2 -t`
   (Enter your regular Linux password when prompted for it)

   (Note: -t turns off db2's default terminator feature, which requires you to end each command with a semicolon. If you omit this, db2 will terminate a command whenever you press return, which is usually not what you want)

4. Issue the following commands at the db2 => prompt to connect to the sample database.

```
connect to sample user your short username ;
set schema sample;
```

Notes on Editing Commands

DB2 does not do command-line editing; however, you can edit the previous command you entered by using the command

`edit editor` *name of your favorite editor*;

(The default editor db2 uses is a real pain to use; so you're much better off specifying an editor that you like!)

When you exit the editor, db2 will display the edited command, and will ask you whether you want to execute it.

Be sure to terminate the db2 Session when you finish by issuing the following command:

```
terminate;
```

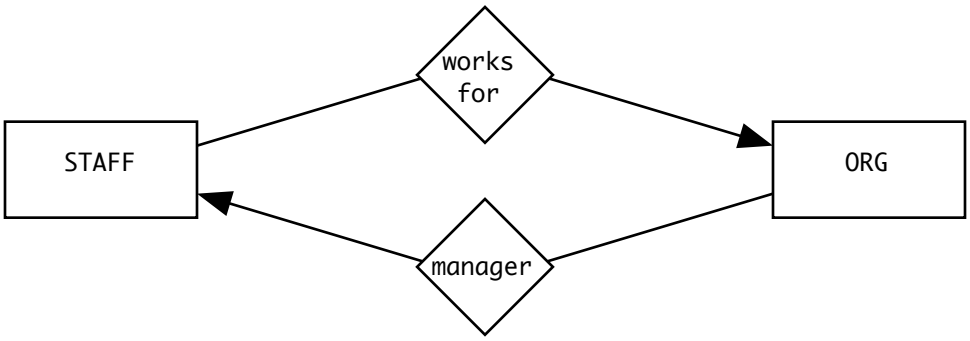
Structure of the `sample` Database

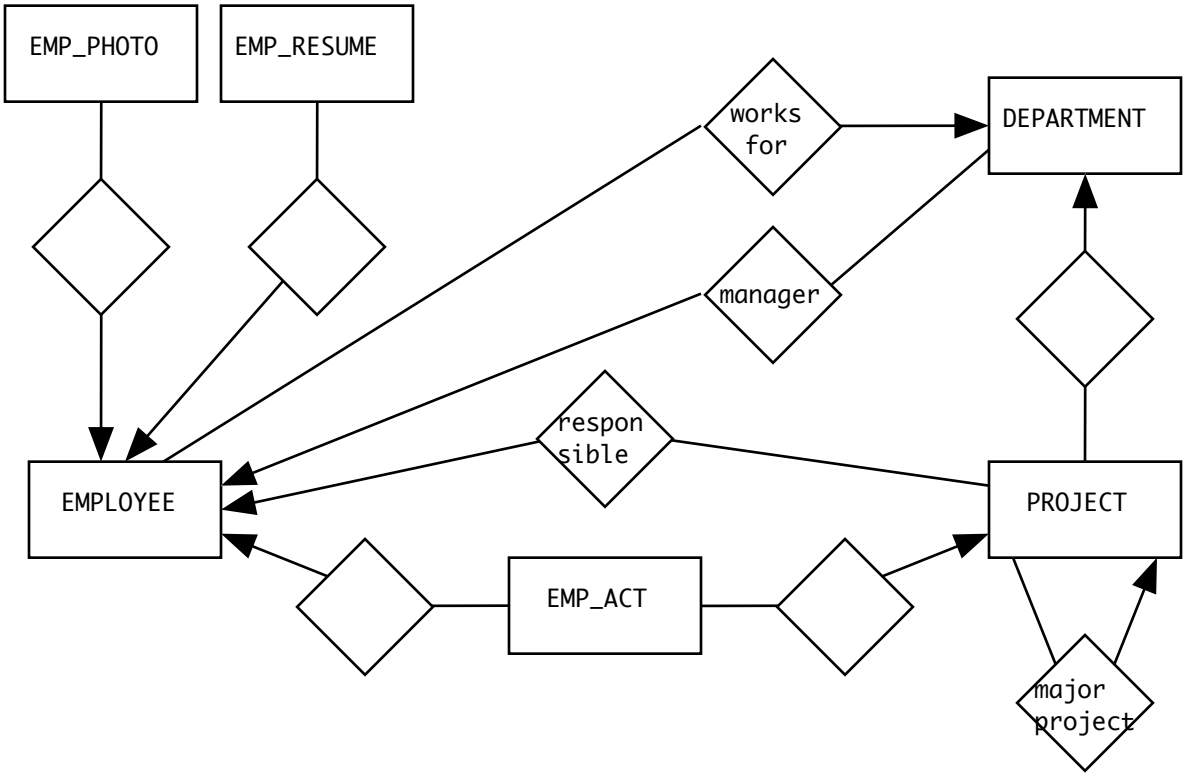
As you can see from the diagrams on the following page, the `sample` database consists of two groups of tables containing somewhat similar information, plus several independent tables. This is not meant to be an example of good database design, but rather as a basis for SQL examples. (Most of the examples in the Birchall book use this database, as does some of the documentation published by IBM).

One group of tables (`org` and `staff`) records information about people working for various departments within a company. The "works for" relationship is represented by each entry in the `staff` table containing a foreign key (`dept`) referring to the primary key (`deptnumb`) of the appropriate row in the `org` table. The "manager" relationship is represented by each entry in the `org` table containing containing a foreign key (`manager`) referring to the primary key (`id`) of the appropriate row in the `staff` table.

Another group of tables has a similar structure involving the `department` and `employee` tables (though the field names are different). In addition, this group of tables includes one that records information about projects (`project`) - each of which is associated with a specific department, and is the responsibility of a particular employee. Moreover, a project may be a part of a larger project. (This information is represented by the foreign keys `deptno`, `respemp` and `majproj`). For each project, the `emp_act` table records information about various activities - each performed by a particular employee. Note that this must be an entity, not a relationship, because a given employee may be involved in several activities for a given project. Finally, for some of the employees, their photo and/or resume is stored in the database as a binary large object (`blob`) or character large object (`blob`).

There are also three standalone tables not connected to any other tables - in fact, two don't initially contain any data. (They are for practice with `insert`).





CL_SCHED

IN_TRAY

SALES

For the following queries, it may be helpful to refer to the Birchall chapter entitled "Introduction to SQL" - especially the sections on "SQL Predicates" and "CASE Expressions"; the chapter on "Column functions" the chapter on "Order by, Group by, and Having" and the chapter on "Joins".

1. What is the average salary for all employees (from employee)? (This should be a single number).

2. What is the average salary by department for all employees in that department? (This should be a table with one row for each department containing the department code and the average salary for that department)

3. What are the full names (first, middle, last) and hire date of all employees hired between January 1, 1970 and December 31, 1979 ordered in descending order of last name?

4. What are the last names of all employees whose last name ends with 'SON' , listed in ordinary alphabetical order of last name? (Hint: use LIKE)

5. What are the full names (first, middle, last) and department name of all employees (from employee), ordered in alphabetical order of department name. Note that this query asks for the name of the department, not the department code so joining two tables is required. The column headings should be FIRST, M, LAST, DEPARTMENT.

   Do the join using "on"

6. Repeat the above problem, but do the join using "where"

7. The above query does not print out any rows for the DEVELOPMENT CENTER, because no employees are listed as working for that department. Formulate a query that includes a row for this department with nulls (printed as -) for the employee information) (**Use on for the join**)

8. List the full names (first, middle, last) and gender of all employees The gender should be printed as 'MALE' or 'FEMALE'. Use the form of the CASE statement that implicitly tests for equality (i.e. your code should not contain something like sex = 'M'). Do not use "else" - explicitly check the two possible values. The column headings should be FIRST, M, LAST, GENDER.

9. What are the full names (first, middle, last) and day of week hired on for all employees. (from employee)? When printing the date hired, show both the day of the week and the calendar
date. The column headings should be FIRST, M, LAST, DAY, DATE'.

   The beginning of the output should look something like the following example:

```
FIRST        M LAST            DAY        DATE
------------ - --------------- ---------- ----------
CHRISTINE    I HAAS            Friday     01/01/1965
```

   Note: the function that converts a date to a day of the week produces an output that is 100 columns wide! Use the following function to truncate it to a more manageable size:
   `varchar(function that calculates day of week, 10)`

10. List the full names (first, middle, last) and education level (column labelled EDUCATION) of all employees. The education level should be printed as 'GRADUATE' if 18 or greater; 'COLLEGE' if 16 or greater; and 'HIGH SCHOOL' if less than 16. The column headings should be FIRST, M, LAST, EDUCATION.