OPIM 5272 — Business Process Modeling and Data Management
Fall 2021, University of Connecticut

Homework 4

Each problem requires you to compose a query which will produce a given output. The queries that you design should not work only for the data that exists in the table—the design of the query should be independent of the data that is in the table, but work according to how the database is set up. As an example, suppose I ask you to write a query that displays the first name of every employee in the Oracle HR Schema whose manager is not specified in the database. A example of a correct answer would be:

```
SELECT first_name
FROM employees
WHERE manager_id IS NULL;
```

An example of a query where you would receive no points is:

```
SELECT first_name
FROM employees
WHERE first_name = 'Steven';
```

Although the output is correct, it depends on the data in the table, and would not work if a different employee was added who also had a NULL manager_id. This should also work if any of the data in the tables is ever changed. If you have any questions regarding this please let me know, but it should be clear what queries are unacceptable, even if the output that they provide is correct.

There are 15 problems in the homework, and each is worth 6 and 2/3 points.

Unless otherwise specified please assume that the queries should be based off of the Oracle HR Schema.

Problem 1

For this problem, you are to compose a script that will accomplish the following tasks. Create a table called dep_active that will contain all of the data in the departments table, including only those instances for which there is at least one employee in the employees table that is in that department (i.e., has the associated department ID). Append a column that represents the average salary of all employees in the employees table that are in that department.

Problem 2

For this problem, you are to compose a script that will accomplish the following tasks. For the table created in Problem 1, set the department id as the primary key. Then, set the department_name as required and unique. Finally, set the location_ID as an optional foreign key attribute.

Problem 3

For this problem, you are to compose a script that will accomplish the following tasks. First, delete the table that you created in Problem 1. Next, the script should copy all of the data in the employees table into a new table that is called copy_employees. Then, add a primary key constraint called copy_emp_pk which enforces that employee_id is the primary key for the newly created table. Finally, update the manager ID to 100 for every employee whose department ID is 50 or 80 in the copy_employee table.

Problem 4

Write an SQL query that will display for every country in the countries table, the number of employees and their corresponding average salary in the employees table that are in a department whose location is in that country. The output should contain three columns: the country name, the number of employees, and the average salary for those employees. Round the average salaries to the nearest integer. There should be an output for each country, even if there is no employee in that country.

Problem 5

Write an SQL query that will display the total salary of all employees in the employees table broken down by job title of each job id. The output should contain two columns: job title and total salary. Only include those employees that were hired in the 1990s. There should be an output for every job, regardless of if there is an employee that matches the criteria.

Problem 6

Write an SQL query that will display three columns. The first two columns will be the department id and manager id for each department in the departments table. The third column will be the manager's last name. Only keep records from departments which have at least two employees.

Problem 7

Write an SQL query to find the employees who are not in the same department as their manager. For each of these employees, output the employee's email and his/her manager's email.

Problem 8

Write an SQL query that will output three columns that reports the salary and job grade for every employee. There should be an output for all jobs ever held by each employee in the employees table, both as reported in the employees table and in the job history table. The first column in the output should be the employee's last name. The second column should be the salary. The third columns should be the job grade. If there is a job in the job history table that doesn't correspond to any employee in the employees table, you should still produce an output, and report NULL as the last name.

Problem 9

Write an SQL query to display the highest salary and the lowest salary for a given department name and job title, for employees who were hired later than the earlist hire date among all IT_PROG. Order the output by department name and job title in alphabetical order.

Problem 10

Write an SQL query that outputs three columns. The first column should output the last name of each employee in the employees table that is the manager for some other employee, the second column is the corresponding department name of the employee from the departments table, and the third column should be the number of employees that this person manages. Arrange your results in descending order of the number of employees managed.

Problem 11

Write an SQL query that will display two columns based only on the employees table. The first column will display the department IDs for the employees listed in this table. The second column should display the maximum salary of all employees with that corresponding department ID. Count towards this maximum only those employees whose hire date is before Jan 1st, 1995. Display an output only for those departments that have a maximum salary of over $10,000$. Order the outputs in descreasing order of department ID.

Problem 12

Create a view that provides the first name, last name, and name of the department in which the employee currently works for each employee in the employees table. Make sure that underlying data in the base table is not editable.

Problem 13

Write an SQL query to find the employees who are the first one(s) to join the company in each year using the employees table. For each of these employees, output his/her employee id and the year in which he/she was hired. For example, if two employees joined the company in 2000 and three employees joined the company in 2001, your report should the employee who was the first one (out of the two) who joined in 2000 and the employee who was the first one who joined in 2001.

Problem 14

Create a query to display the last name and salary of those managers (i.e., those employees in the employees table who are the manager of at least one other employee) who are earning more than the maximum salary of ALL employees who are not managers.

Problem 15

Write a query that displays the number of department managers (i.e., those employees that are managers of some department in the departments table) that have each job grade_level. Output a row for every grade_level in the job_grades table, independent of whether there are any department managers with the corresponding grade_level.