January 2025 CSE216: Database Sessional

Online Assignment on PL/SQL

Subsection: B1+B2

Time: 60 minutes **Marks: 10+10+10**

Question 1.

First create a copy of the **EMPLOYEES** table in the HR schema, named **EMPLOYEES_COPY.** Do all the work in **EMPLOYEES_COPY** table.

Write a function named **Exchange_Employees** that should take two **manager_ids** as parameters and exchange the lowest-paying employees under each manager. If there is a tie (i.e., multiple employees have the same salary), select any one of them. Note that when employees are exchanged, their jobs will remain the same, but one employee will join the department of the other employee under that employee's manager. The salaries of the exchanged employees will be updated. The new salary for each employee will be increased by **50%** of the difference between their original salaries. Print the employee's information before and after the exchange and handle appropriate exceptions.

To copy a table, execute the following script-

```
CREATE TABLE employees_copy AS
SELECT * FROM employees;
COMMIT;
```

Before `EXCHANGE_EMPLOYEES(100,145)`:

```
Employee Information:
Employee ID: 124  Name: Kevin Mourgos
Email: KMOURGOS  Phone Number: 650.123.5234
Hire Date: 16-NOV-07
Job ID: ST_MAN  Salary: 5800
Commission Percentage:
Manager ID: 100  Department ID: 50
Employee Information:
Employee ID: 155  Name: Oliver Tuvault
Email: OTUVAULT  Phone Number: 011.44.1344.486508
Hire Date: 23-NOV-07
Job ID: SA_REP  Salary: 7000
Commission Percentage: .15
Manager ID: 145  Department ID: 80
```

After `EXCHANGE_EMPLOYEES(100,145)`:

```
Employee Information:
Employee ID: 124  Name: Kevin Mourgos
Email: KMOURGOS  Phone Number: 650.123.5234
Hire Date: 16-NOV-07
Job ID: ST_MAN  Salary: 6400
Commission Percentage:
Manager ID: 145  Department ID: 80
Employee Information:
Employee ID: 155  Name: Oliver Tuvault
Email: OTUVAULT  Phone Number: 011.44.1344.486508
Hire Date: 23-NOV-07
Job ID: SA_REP  Salary: 7600
Commission Percentage: .15
Manager ID: 100  Department ID: 50
```

Question 2.

Write a PL/SQL procedure named **LOCATION_SALARY_REPORT** that performs the following tasks:

- For each location (city), compute:
  - The number of employees working in that city
  - The average salary of those employees (rounded to 2 decimal places)
  - The job title of the highest-paid employee in that city
- Rank the cities based on:
  - Ascending order of the number of employees
  - Descending order of average salary (used as a tie-breaker)
- Print the following information for each location:
  - Rank
  - City Name
  - Number of Employees
  - Average Salary
  - Highest Paying Job Title

Make sure to handle exceptions with appropriate messages.

Question 3.

Create a trigger that activates when an employee is transferred to a new department (i.e., when an UPDATE operation on the department_id is performed in the Employee table).

1 new table: **Transfers**
(Fields:
employee_id,
employee_working_instead_of_him,
new_department,
current date
)

Conditions:

1. If that employee had a manager,
   a. An employee with the closest salary to him/her under the same manager will work instead of him/her. The new salary for that **work-in-place** employee will be equal to previous salary of that **work-in-place** employee + 0.5* salary of the transferred employee.
   b. His/her new manager in the new department should be the manager with the closest number of subordinates to his/her previous manager.

No changes in the Job table and the Job_history table are necessary for your ease.