**NetID: jc12300**

**Name: Jie Cheng**

**Course: CSGY-6083-Principles of Database Systems**

**Section: B**

**Date of submission: Dec 5, 2023**



**CS-GY 6083 - B, FALL 2023**

Principles of Database Systems

Assignment: 4 [100 points]

**Please submit your assignment on NYU Brightspace course site with a single PDF document attachment. Please mention Student ID, Name, Course, Section Number, and date of submission on first page of your submission. Each table in your submission of SQLs and their results should have your initial as prefix, e.g., AP\_EMPLOYEE etc. You can use either Oracle or MySQL for this assignment.**

Q1) To write a database procedure (Oracle or MySQL) [60 points]

The HR department intend to give salary increment to employees of specific department when requested by their department director. Different directors have different criteria about salary increment. For an example, some directors ask for base increment as 5% of average salary of their department, and some may ask for base increment as 7% or 10% of average salary. So, the base increment percent of avg. salary is determined by the department director. However, following criteria remains same for all departments.

The new salary is calculated by the formula, New Salary = S + N% of A+ S\*Y%

S= original salary

N%= base increment percent of department’s avg. salary (e.g 5, 7, 10 etc.) A= average salary of the department

Y=Square root of number of years employee’s working as of Dec. 31st, 2022.

Write a database procedure that takes two input variables department number and base N percentage of avg salary. Apply salary increment criteria as detailed above. Your procedure name should have your initial as prefix, e.g. AP\_RAISE\_SAL.

Use the table and its data attached to the assignment. Submit:

1. **Procedure code (Oracle or MySQL)**

CREATE OR REPLACE PROCEDURE JC\_RAISE\_SAL (

p\_department\_id JC\_EMPLOYEE.DEPARTMENT\_ID%TYPE,

p\_base\_increment\_percent NUMBER

) AS

v\_avg\_salary NUMBER;

v\_increment\_amount NUMBER;

v\_years\_worked NUMBER;

v\_new\_salary NUMBER;

BEGIN

-- Calculate the average salary for the specified department

SELECT AVG(SALARY)

INTO v\_avg\_salary

FROM JC\_EMPLOYEE

WHERE DEPARTMENT\_ID = p\_department\_id;

-- Loop through employees in the specified department

FOR emp\_rec IN (SELECT EMPLOYEE\_ID, SALARY, HIRE\_DATE FROM JC\_EMPLOYEE WHERE DEPARTMENT\_ID = p\_department\_id)

LOOP

-- Calculate the number of years worked as of Dec. 31st, 2022

v\_years\_worked := TRUNC(MONTHS\_BETWEEN(TO\_DATE('31-DEC-2022', 'DD-MON-YYYY'), emp\_rec.HIRE\_DATE) / 12);

-- Calculate the increment amount based on the given criteria

v\_increment\_amount := (p\_base\_increment\_percent / 100) \* v\_avg\_salary + emp\_rec.SALARY \* SQRT(v\_years\_worked) / 100;

-- Calculate the new salary

v\_new\_salary := emp\_rec.SALARY + v\_increment\_amount;

-- Update the employee's salary in the database

UPDATE JC\_EMPLOYEE

SET SALARY = v\_new\_salary

WHERE EMPLOYEE\_ID = emp\_rec.EMPLOYEE\_ID;

END LOOP;

-- Commit the changes

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Salary increment applied successfully.');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No employees found for the specified department.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occurred: ' || SQLERRM);

END JC\_RAISE\_SAL;

/

1. **If you are using Oracle, provide result of following,**

SELECT employee\_id, first\_name,last\_name,hire\_date,department\_id, salary FROM ap\_employee

WHERE department\_id=90;

图形用户界面, 表格

描述已自动生成

execute ap\_raise\_sal (90, 5); -- 90 is the department\_id and 5 is base increment of avg. salary SELECT employee\_id, first\_name,last\_name,hire\_date,department\_id, salary

FROM ap\_employee WHERE department\_id=90;

手机屏幕的截图

描述已自动生成

SELECT employee\_id, first\_name,last\_name,hire\_date,department\_id, salary FROM ap\_employee

WHERE department\_id=60;

屏幕上有字

描述已自动生成

execute ap\_raise\_sal (60, 5); -- 60 is the department\_id and 5 is base increment of avg. salary SELECT employee\_id, first\_name,last\_name,hire\_date,department\_id, salary

FROM ap\_employee

WHERE department\_id=60;

手机的屏幕截图

描述已自动生成

If you are using MySQL, provide result of following,

SELECT employee\_id, first\_name,last\_name,hire\_date,department\_id, salary FROM ap\_employee

WHERE department\_id=90;

call ap\_raise\_sal (90, 5); -- 90 is the department\_id and 5 is base increment of avg. salary SELECT employee\_id, first\_name,last\_name,hire\_date,department\_id, salary

FROM ap\_employee

WHERE department\_id=90;

SELECT employee\_id, first\_name,last\_name,hire\_date,department\_id, salary FROM ap\_employee

WHERE department\_id=60;

call ap\_raise\_sal (60, 5); -- 60 is the department\_id and 5 is base increment of avg. salary SELECT employee\_id, first\_name,last\_name,hire\_date,department\_id, salary

FROM ap\_employee WHERE department\_id=60;

Q2) Indexes [40 points]

Consider following queries to the same employee table that used in Q1. select \* from ap\_employee

where substr(last\_name,1,1)='A'and JOB\_ID='SA\_REP'

order by last\_name;

select upper(first\_name), upper(last\_name), department\_id, salary

from ap\_employee a where a.salary>(select avg(salary) from ap\_employee b

where b.department\_id=department\_id);

For each of the above query do following

1. **Suggest which column(s) are suitable for indexes and what type of index should be created.**
2. **Create index(es) as suggested in step a**
3. **Create the query execution plan.**

Submit

1. **Suggested column(s) for index(es) and type of the index(es)**

For query 1

An index on the last\_name column would be suitable for optimizing the WHERE clause and the ORDER BY clause. The type of the index is function based index.

For query 2

An index on the department\_id column would be suitable for optimizing the subquery. The type of the index is bitmap index.

1. **DDL code of the index(es) created.**

For query 1

CREATE INDEX idx\_last\_name ON ap\_employee(last\_name);

For query 2

CREATE INDEX idx\_department\_id ON ap\_employee(department\_id);

1. **Screenshot of execution plan**

For query 1

文本

描述已自动生成

For query 2

文本

描述已自动生成

1. **Explanation about which index(es) are used and which are not, and reason for it**

For query 1

last name is used because it is used in ORDER BY clause.

For query 2

department id is used because it is used frequently in WHERE clause. Salary is not used because it might be frequently updated.