NYU | TANDON SCHOOL OF ENGINEERING

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:

a) Procedure code (Oracle or MySQL)

b) 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;

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;


SELECT employee_id, first_name,last_name,hire_date,department_id, salary
FROM ap_employee
WHERE department_id=60;

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;
```

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

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

Submit

i) Suggested column(s) for index(es) and type of the index(es)
ii) DDL code of the index(es) created.
iii) Screenshot of execution plan
iv) Explanation about which index(es) are used and which are not, and reason for it