****

**CS-GY 6083 - B, FALL 2021**

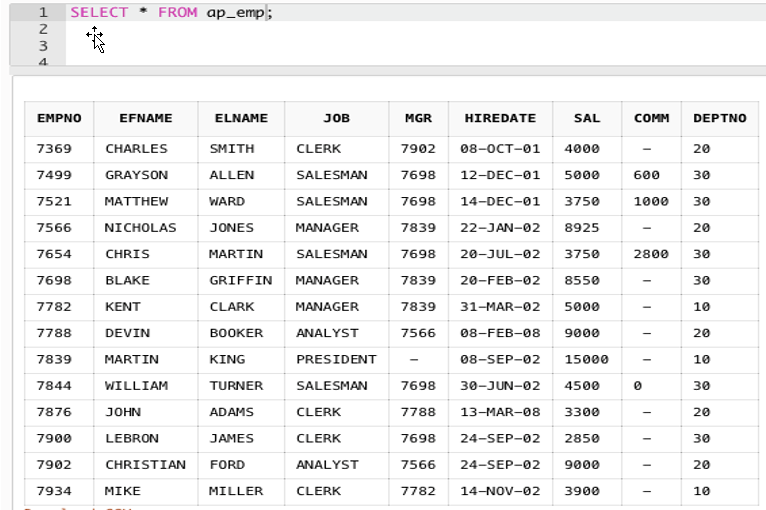
**Principles of Database Systems**

**Possible Solution Assignment: 4 [100 points]**

**Please submit your assignment to NYU Brightspace with a single PDF document attachment. Please mention Student ID, Name, Course, Section Number, and date of submission on the first page of your submission. This is an individual assignment and you should create your own work. All table names in your submission should have your prefix as your initial, e.g. AP\_EMP, where AP is the initial of the student.**

1. **To write a database procedure (Oracle or MySQL) [30 points]**

**Consider following employee table in your practice schema. The table should have your initial as prefix.**



**The HR department intend to give salary increment to all employees at every six months with specific base increment amount, for example $300. Each time HR department gives salary increment, the base increment amount is not same**

**and it varies. So, application team intend to write a database procedure, that takes base salary increment amount as input to the procedure, for example N.**

**The procedure should update everyone’s salary with following criteria,**

* **Everyone’s salary is increased by at least, the value N (e.g. 300 if N=300)**
* **Number of years of service >= 5, then salary is increased with 10% more of amount N (e.g. 330 if N=300)**
* **Number of years of service >=10, then salary is increased with 15% more of amount N (e.g. 345 if N=300)**
* **Number of years of service >=15 then salary is increased with 20% more of amount N (e.g. 360 if N=300)**

**Write a database procedure that do work as mentioned above. Your procedure name should have your initial as prefix, e.g. AP\_RAISE\_SAL.**

**Submit the procedure code, execute procedure for at least 4 salary change, and screenshot of table data after the salary changes.**

REM: Oracle procedure code

CREATE OR REPLACE PROCEDURE raise\_sal (sal\_raise NUMBER)

IS

duration number;

BEGIN

UPDATE xq\_emp

SET sal = sal + sal\_raise \* (CASE CEIL(months\_between(sysdate, hiredate)/12/5)

WHEN 0 THEN 1

WHEN 1 THEN 1

WHEN 2 THEN 1.1

WHEN 3 THEN 1.15

ELSE 1.2

END);

END raise\_sal;

**REM: MySQL procedure code**

**DELIMITER //**

**CREATE PROCEDURE raise\_sal (IN sal\_raise INT)**

**BEGIN**

**UPDATE xq\_emp**

**SET sal = sal + sal\_raise \* (CASE CEIL(timestampdiff(MONTH, hiredate, CURRENT\_DATE)/12/5)**

**WHEN 0 THEN 1**

**WHEN 1 THEN 1**

**WHEN 2 THEN 1.1**

**WHEN 3 THEN 1.15**

**ELSE 1.2**

**END);**

**END; //**

**DELIMITER ;**

1. **To write a database function (Oracle or MySQL) [30 points]**

**Consider the same employee table in question A.**

**Write a database function that takes Hiredate as input and calculate employment duration as of system date. The employment duration should be returned in format of string that looks as “12 YEARS 8 MONTHS 24 DAYS”. Your function name should have your initial as prefix, e.g. AP\_HIRE\_DURATION**

**Submit the function code, use the function and show data of employee number, hire date and hire duration using function. Arrange the result in order of employee number.**

**REM: Oracle function code**

**CREATE OR REPLACE FUNCTION Cal\_Hire\_Duration(hiredate IN DATE) RETURN VARCHAR2**

**AS**

**month\_gap NUMBER;**

**years NUMBER;**

**months NUMBER;**

**days NUMBER;**

**duration VARCHAR2(50);**

**BEGIN**

**month\_gap := trunc(months\_between(sysdate, hiredate));**

**years := trunc(month\_gap/12);**

**months := mod(month\_gap,12);**

**days := trunc(sysdate - add\_months(hiredate, month\_gap));**

**duration := years || ' YEARS ' || months || ' MONTHS ' || days || ' DAYS'; RETURN duration;**

**END;**

**/**

**REM: MySQL function code**

**DELIMITER $$**

**DROP FUNCTION IF EXISTS Cal\_Hire\_Duration $$ CREATE FUNCTION Cal\_Hire\_Duration(hiredate DATE) RETURNS VARCHAR(30)**

**DETERMINISTIC BEGIN**

**DECLARE month\_gap INT;**

**DECLARE years INT;**

**DECLARE months INT;**

**DECLARE days INT;**

**DECLARE duration VARCHAR(30);**

**SET month\_gap = truncate(timestampdiff(MONTH, hiredate, CURRENT\_DATE),0);**

**SET years = truncate(month\_gap/12,0);**

**SET months = mod(month\_gap,12);**

**SET days = datediff(CURRENT\_DATE, DATE\_ADD(hiredate, INTERVAL month\_gap MONTH));**

**SET duration = concat(years, ' YEARS ', months, ' MONTHS ', days, ' DAYS');**

**RETURN duration;**

**END;**

**$$**

**DELIMITER ;**

1. **Solution to BITMAP index (Oracle or MySQL) [40 points]**

**Create patient table, your table name should have your initial as prefix**

create table AP\_PATIENT

(patient\_id number(10),

gender char(1),

marital\_status char(1),

race char(1),

primary key (patient\_id));

**Populate data to your patient table, with exact data as follow.**

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10001, 'F','S','A');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10002, 'F','S','W');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10003, 'F','S','B');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10004, 'F','M','A');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10005, 'F','D','B');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10006, 'F','W','A');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10007, 'F','M','W');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10008, 'F','W','B');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10009, 'M','S','A');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10010, 'M','S','W');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10011, 'M','S','B');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10012, 'M','M','A');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10013, 'M','D','B');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10014, 'M','W','A');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10015, 'M','M','W');

insert into ap\_patient (patient\_id,gender,marital\_status,race) values (10016, 'M','W','B');

**In this dataset,**

**Gender are M (male), F (female)**

**Marital\_Status are S (single), M (married), D (divorced), and W (Widow or Widower)**

**Race are A (Asian), B (Black), W (White)**

1. **Identify columns suitable of creating bitmap indexes, and write bitmaps of each of such bitmap indexes. (10 points). Submit bitmaps of each distinct value of each of such bitmap indexes.**

**Ans:**

**The columns which are suitable for creating bitmap index are gender, marital\_status and race**

**Bitmap of gender:**

**For Male(M): 0000000011111111**

**For Female(F): 1111111100000000**

**Bitmap of marital\_status:**

**For Single(S): 1110000011100000**

**For Married(M): 0001001000010010**

**For Divorced(D): 0000100000001000**

**For Widow(W):0000010000000101**

**Bitmap of race:**

**For Asian(A): 1001010010010100**

**For Black(B): 0010100100101001**

**For White(W): 0100001001000010**

1. **Using bitmaps created in step i, solve the business question “List patient IDs of those patients who are Female and not Asian, and their marital status is either Single or Married”. Submit intermediate results and final result of bitmap. List Patient Id of the final result. [10 point]**

**Ans:**

**Intermediate results:**

**Marital status is either single or married:**

**Single OR married = 1110000011100000 OR 0001001000010010**

**= 1111001011110010**

**Not Asian = NOT 1001010010010100 = 0110101101101011**

**Final result: Female AND (NOT Asian) AND (Single OR Married)**

**= 1111111100000000 AND 0110101101101011 AND 1111001011110010**

**= 0110001000000000**

**Patient IDs: 10002, 10003, 1007**

1. **Write a SQL query that solves business question in Q2 [10 points]**

**SELECT patient\_id from ss\_patient where gender = 'F' AND (race != 'A' ) AND (marital\_status = 'S' OR marital\_status ='M');**



1. **Write DDL code to create bitmap indexes for which bitmaps are created in step i [10 points]**

**CREATE BIMPAP INDEX bi\_ind\_gender on AP\_PATIENT (gender);**

**CREATE BIMPAP INDEX bi\_ind\_matital\_status on AP\_PATIENT (marital\_status);**

**CREATE BIMPAP INDEX bi\_ind\_race on AP\_PATIENT (race);**