**Лабораторная работа № 8 СХЕМА HR PL/SQL Программы**

Выполните следующие PL/SQL Программы.  
Настройка SQL. Необходимо включить режим ECHO и вывести протокол лаб. работы в файл *<Фамилия студента>.txt*. Этот файл является отчетом о проделанной лаб. работе.

SQL> set echo on

SQL> spool c:\spool. txt

……………………………………..

Завершить протокол лаб. работы (команда spool off) и направить результаты преподавателю.

- Не забудьте включить режим !!!!!!!

SQL> **SET SERVEROUTPUT ON**

SQL> spool off

|  |  |
| --- | --- |
| Пример программы замещения (изменения) з/п работника  с таб.номером 120 на з/п работника 122 и наоборот.  Declare  V\_salary\_120 employees.salary%type;  Begin  Select salary into v\_salary\_120  From employees where employee\_id = 120;  Update employees set salary = ( select salary from employees where employee\_id = 122)  Where employee\_id = 120;  Update employees set salary = v\_salary\_120 Where employee\_id = 122;  Commit;  End;  ПРОВЕРЬТЕ РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ  SQL> Select salary v\_salary\_120  From employees where employee\_id = 120;  SQL> Select salary v\_salary\_122  From employees where employee\_id = 122;  / ------- ПРОГРАММА ---------- /  SQL> Select salary v\_salary\_120  From employees where employee\_id = 120;  SQL> Select salary v\_salary\_122  From employees where employee\_id = 122;  declare  v VARCHAR2(20);  begin  v:=get\_dept\_manager\_name (10);  dbms\_output.put\_line(v);  end;  Пример программы увеличения з/п работника с таб.номером 115 по следующей схеме: если стаж более, чем 5 лет з/п увеличивается на 10%, больше 10 лет - на 20%, во всех остальных случаях на 5%.  declare  v\_exp number(2);  v\_hike number(5,2);  begin  select floor((sysdate-hire\_date) / 365 ) into v\_exp  from employees  where employee\_id = 115;    v\_hike := 1.05;    case  when v\_exp > 10 then  v\_hike := 1.20;  when v\_exp > 5 then  v\_hike := 1.10;  end case;    update employees set salary = salary \* v\_hike  where employee\_id = 115;  end;  ПРОВЕРЬТЕ РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ  SQL> Select salary v\_salary\_115  From employees where employee\_id = 115;  / ------- ПРОГРАММА ---------- /  SQL> Select salary v\_salary\_115  From employees where employee\_id = 115;  SQL> ROLLBACK;  Пример программы изменения комиссии у работника  с таб.номером 150 по следующей схеме: если з/п больше 10000, то комиссия 0.4%, если з/п меньше 10000, а стаж больше 10 лет, то комиссия 0.35%, если з/п меньше 3000, то комиссия 0.25%,  во всех остальных случаях комиссия 0,15%.  declare  v\_salary employees.salary%type;  v\_exp number(2);  v\_cp number(5,2);  begin  select v\_salary, floor ( (sysdate-hire\_date)/365) into v\_salary, v\_exp  from employees  where employee\_id = 150;    if v\_salary > 10000 then  v\_cp := 0.4;  elsif v\_exp > 10 then  v\_cp := 0.35;  elsif v\_salary < 3000 then  v\_cp := 0.25;  else  v\_cp := 0.15;    end if;    update employees set commission\_pct = v\_cp  where employee\_id = 150;  end;  ПРОВЕРЬТЕ РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ  Пример программы поиска работника и соответствующего отдела по руководителю с таб.номером 103  declare  v\_name employees.first\_name%type;  v\_deptname departments.department\_name%type;  begin  select first\_name , department\_name into v\_name, v\_deptname  from employees join departments using (department\_id)  where employee\_id = ( select manager\_id from employees where employee\_id = 103);    dbms\_output.put\_line(v\_name);  dbms\_output.put\_line(v\_deptname);    end;  ПРОВЕРЬТЕ РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ  Пример программы поиска "пропавшего" работника  declare  v\_min number(3);  v\_max number(3);  v\_c number(1);  begin  select min(employee\_id), max(employee\_id) into v\_min, v\_max  from employees;  for i in v\_min + 1 .. v\_max - 1  loop  select count(\*) into v\_c  from employees  where employee\_id = i;    if v\_c = 0 then  dbms\_output.put\_line(i);  end if;  end loop;    end;  Пример программы отображения года, в котором максимальное количество сотрудников поступило на работу (и сколько поступило в каждом месяце в этом году)  declare  v\_year number(4);  v\_c number(2);  begin  select to\_char(hire\_date,'yyyy') into v\_year  from employees  group by to\_char(hire\_date,'yyyy')  having count(\*) =  ( select max( count(\*))  from employees  group by to\_char(hire\_date,'yyyy'));    dbms\_output.put\_line('Year : ' || v\_year);  for month in 1 .. 12  loop  select count(\*) into v\_c  from employees  where to\_char(hire\_date,'mm') = month and to\_char(hire\_date,'yyyy') = v\_year;    dbms\_output.put\_line('Month : ' || to\_char(month) || ' Employees : ' || to\_char(v\_c));  end loop;  end;  Пример программы изменения оклада сотрудника 130 на оклад сотрудника с именем «Джо». Если Джо не найден, то расчитываем среднюю зарплату всех сотрудников. Если найдено более одного сотрудника с именем «Джо», то берем наименьшую зарплату сотрудников с именем Джо.  declare  v\_salary employees.salary%type;  begin  select salary into v\_salary  from employees where first\_name = 'Joe';    update employees set salary = v\_salary  where employee\_id = 130;    exception  when no\_data\_found then  update employees set salary = (select avg(salary) from employees)  where employee\_id = 130;  end;  Показать должность и имя сотрудника, который работает первый день.  declare  cursor jobscur is select job\_id, job\_title from jobs;  v\_name employees.first\_name%type;  begin  for jobrec in jobscur  loop  select first\_name into v\_name  from employees  where hire\_date = ( select min(hire\_date) from employees where job\_id = jobrec.job\_id)  and job\_id = jobrec.job\_id;    dbms\_output.put\_line( jobrec.job\_title || '-' || v\_name);  end loop;  end;  Отобразите c 5-го по 10-го сотрудников из таблицы «Сотрудники».  declare  cursor empcur is  select employee\_id, first\_name  from employees;    begin  for emprec in empcur  loop  if empcur%rowcount > 4 then  dbms\_output.put\_line( emprec.first\_name);  exit when empcur%rowcount > 10;  end if;  end loop;    end;  Обновление зарплаты сотрудника в зависимости от отдела и процента комиссии. Если отдел - 40, увеличьте зарплату на 10%. Если отдел - 70, то на 15%, если комиссия больше 0,3%, то 5%, в противном случае 10%.  declare  cursor empcur is  select employee\_id, department\_id, commission\_pct  from employees;    v\_hike number(2);  begin  for emprec in empcur  loop  if emprec.department\_id = 40 then  v\_hike := 10;  elsif emprec.department\_id = 70 then  v\_hike := 15;  elsif emprec.commission\_pct > 0.30 then  v\_hike := 5;  else  v\_hike := 10;  end if;  update employees set salary = salary + salary \* v\_hike/100  where employee\_id = emprec.employee\_id;    end loop;  end;  Создайте функцию, которая принимает идентификатор отдела и возвращает имя руководителя отдела.  create or replace function get\_dept\_manager\_name(deptid number)  return varchar is  v\_name employees.first\_name%type;  begin  select first\_name into v\_name  from employees  where employee\_id = ( select manager\_id from departments where department\_id = deptid);  return v\_name;  end;  ПРОВЕРЬТЕ РЕЗУЛЬТАТ РАБОТЫ ФУНКЦИИ  SQL> set serveroutput on  declare  v VARCHAR2(20);  begin  v:=get\_dept\_manager\_name (10);  dbms\_output.put\_line(v);  end;  Создайте функцию, которая принимает идентификатор сотрудника и возвращает количество заданий, выполненных сотрудником в прошлом.  create or replace function get\_no\_of\_jobs\_done(empid number)  return number is  v\_count number(2);  begin  select count(\*) into v\_count  from job\_history  where employee\_id = empid;  return v\_count;  end;  ПРОВЕРЬТЕ РЕЗУЛЬТАТ РАБОТЫ ФУНКЦИИ  Создайте процедуру, которая принимает идентификатор отдела и заменяет идентификатор менеджера отдела на сотрудника отдела с самой высокой зарплатой.  create or replace procedure change\_dept\_manager(deptid number)  is  v\_empid employees.employee\_id%type;  begin  select employee\_id into v\_empid  from employees  where salary = ( select max(salary) from employees where department\_id = deptid)  and department\_id = deptid;  update departments set manager\_id = v\_empid  where department\_id = deptid;  dbms\_output.put\_line(v\_empid);  end;  ПРОВЕРЬТЕ РЕЗУЛЬТАТ РАБОТЫ ПРОЦЕДУРЫ  SQL> set serveroutput on  SQL>select salary salary\_max, employee\_id v\_empid  from employees  where salary = (select max(salary) from employees  where department\_id = 10) and department\_id = 10;  declare  id number(4);  begin  id:=10;  change\_dept\_manager(id);  end;  Создайте функцию, которая принимает идентификатор менеджера и возвращает имена сотрудников, которые отчитываются перед этим менеджером. Имена должны быть возвращены в виде строки с запятыми, разделяющими имена.  create or replace function get\_employees\_for\_manager(manager number)  return varchar2  is  v\_employees varchar2(1000) := '';  cursor empcur is  select first\_name from employees  where manager\_id = manager;  begin  for emprec in empcur  loop  v\_employees := v\_employees || ',' || emprec.first\_name;  end loop;  -- remove extra , at the beginning  return ltrim(v\_employees,',');  end;  ПРОВЕРЬТЕ РЕЗУЛЬТАТ РАБОТЫ ФУНКЦИИ  Убедитесь, что в таблицу EMPLOYEES нельзя вносить изменения до 6 часов утра и после 10 часов вечера в течение дня.  create or replace trigger trg\_employees\_time\_check  before update or insert or delete  on employees  for each row  begin  if to\_char(sysdate,'hh24') < 6 or to\_char(sysdate,'hh24') > 10 then  raise\_application\_error(-20111,'Sorry! No change can be made before 6 AM and after 10 PM');  end if;  end;  ПРОВЕРЬТЕ РЕЗУЛЬТАТ РАБОТЫ ТРИГГЕРА  Создайте триггер, чтобы заработная плата работника не уменьшалась.  create or replace trigger trg\_employees\_salary\_check  before update  on employees  for each row  begin  if :old.salary > :new.salary then  raise\_application\_error(-20111,'Sorry! Salary can not be decreased!');  end if;  end;  ПРОВЕРЬТЕ РЕЗУЛЬТАТ РАБОТЫ ТРИГГЕРА  SQL>update employees set salary = 10  where department\_id = 10;  Создайте триггер, чтобы сотрудник и менеджер принадлежали к одному отделу.  Примечание: этот триггер должен прочитать строку, которая изменяется, что вызывает проблему с мутацией.  create or replace trigger trg\_log\_job\_change  after update of job\_id  on employees  for each row  declare  v\_enddate date;  v\_startdate date;  begin  -- find out whether the employee has any row in job\_history table  select max(end\_date) into v\_enddate  from job\_history  where employee\_id = :old.employee\_id;  if v\_enddate is null then  v\_startdate := :old.hire\_date;  else  v\_startdate := v\_enddate + 1;  end if;  insert into job\_history values (:old.employee\_id, v\_startdate, sysdate, :old.job\_id, :old.department\_id);  end;   ПРОВЕРЬТЕ РЕЗУЛЬТАТ РАБОТЫ ТРИГГЕРА | CREATE FUNCTION a\_function () RETURNS void AS $$  DECLARE  v\_salary\_120 real;  BEGIN  Select salary  From employees into v\_salary\_120 where employee\_id = 120;  Update employees set salary = ( select salary from employees where employee\_id = 122)  Where employee\_id = 120;  Update employees set salary = v\_salary\_120 Where employee\_id = 122;  END;  $$ LANGUAGE plpgsql;  SELECT a\_function() AS output;  Select salary From employees where employee\_id = 120 or employee\_id = 122;  CREATE or replace FUNCTION b\_function () RETURNS void AS $$  DECLARE  v\_exp integer;  v\_hike real;  BEGIN  select EXTRACT(DAY FROM (NOW() - hire\_date) / 365 ) into v\_exp  from employees where employee\_id = 115;    v\_hike := 1.05;    case  when v\_exp > 10 then  v\_hike := 1.20;  when v\_exp > 5 then  v\_hike := 1.10;  end case;    update employees set salary = salary \* v\_hike  where employee\_id = 115;  END;  $$ LANGUAGE plpgsql;  SELECT b\_function() AS output;  Select salary From employees where employee\_id = 115; |