

Download đề bài tại:

<http://www.mediafire.com/?3hh975tcdvici7z>

Download cơ sở dữ liệu tại: <http://www.mediafire.com/?amchtkg9ilx1wk1>

Sau đó import dữ liệu vào oracle.

TRUY VẤN ORACLE

1. Liệt kê tên (last_name) và lương (salary) của những nhân viên có lương lớn hơn 12000\$.

```
SELECT LAST_NAME, SALARY  
FROM employees;
```

2. Liệt kê tên và lương của những nhân viên có lương thấp hơn 5000\$ hoặc lớn hơn 12000\$.

```
SELECT FIRST_NAME, LAST_NAME, SALARY  
FROM EMPLOYEES  
WHERE SALARY<5000 OR SALARY>12000;
```

3. Cho biết thông tin tên nhân viên (last_name), mã công việc (job_id), ngày thuê (hire_date) của những nhân viên được thuê từ ngày 20/02/1998 đến ngày 1/05/1998.

Thông tin được hiển thị tăng dần theo ngày thuê.

```
SELECT last_name,job_id,hire_date  
FROM employees  
WHERE hire_date BETWEEN '20/FEB/1998' AND '1/MAY/1998'  
ORDER BY hire_date ASC;
```

4. Liệt kê danh sách nhân viên làm việc cho phòng 20 và 50. Thông tin hiển thị gồm:

last_name, department_id, trong đó tên nhân viên được sắp xếp theo thứ tự alphabe.

```
SELECT LAST_NAME, DEPARTMENT_ID  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID IN(20,50);
```

5. Liệt kê danh sách nhân viên được thuê năm 1994.

```
SELECT *  
FROM EMPLOYEES  
WHERE TO_CHAR(HIRE_DATE, 'YY')='05';
```

6. Liệt kê tên nhân viên (last_name), mã công việc (job_id) của những nhân viên không có người quản lý.

```
SELECT LAST_NAME, FIRST_NAME, JOB_ID
FROM EMPLOYEES
WHERE MANAGER_ID IS NULL;
```

7. Cho biết thông tin tất cả nhân viên được hưởng hoa hồng (commission_pct), kết quả được sắp xếp giảm dần theo lương và hoa hồng.

```
SELECT FIRST_NAME, LAST_NAME, COMMISSION_PCT
FROM EMPLOYEES
WHERE COMMISSION_PCT IS NOT NULL;
```

8. Liệt kê danh sách nhân viên mà có kí tự thứ 3 trong tên là "a".

```
SELECT *
FROM EMPLOYEES
WHERE FIRST_NAME LIKE ('__a%');
```

9. Liệt kê danh sách nhân viên mà trong tên có chứa một chữ "a" và một chữ "e".

```
SELECT employee_id, first_name, last_name
FROM employees
WHERE first_name LIKE ('%a%e%');
```

10. Cho biết tên (last_name), mã công việc (job_id), lương (salary) của những nhân viên làm "Sales representative" hoặc "Stock clert" và có mức lương khác 2500\$, 3500\$, 7000\$.

--CACH 1

```
SELECT A.last_name, A.job_id, A.salary
FROM EMPLOYEES A INNER JOIN JOBS b
ON A.job_id =b.job_id
WHERE job_title IN ('Sales Representative' , 'Stock Clerk')
AND salary NOT IN (2500,3500,7000);
```

--cach 2

```
SELECT A.last_name, A.JOB_ID, A.SALARY
FROM EMPLOYEES A, JOBS B
WHERE B.JOB_TITLE IN ('Sales Representative', 'Stick clerk')
AND A.SALARY NOT IN(2500,3500,7000);
```

--CACH 3

```
SELECT A.LAST_NAME, A.JOB_ID, A.SALARY
FROM EMPLOYEES A
```

```

WHERE A.JOB_ID IN
      (SELECT B.JOB_ID
       FROM JOBS B
       WHERE JOB_TITLE IN('Sales Representative','Stick
cleark'))
AND A.SALARY NOT IN(2500, 3500, 7000);
--CACH 3.1
SELECT A.LAST_NAME, A.JOB_ID, A.SALARY
FROM EMPLOYEES A
WHERE A.SALARY NOT IN(2500, 3500, 7000)
AND A.JOB_ID IN
      (SELECT B.JOB_ID
       FROM JOBS B
       WHERE JOB_TITLE IN('Sales Representative','Stick
cleark'));
--CACH 3.2
SELECT A.LAST_NAME, A.JOB_ID, A.SALARY
FROM EMPLOYEES A
WHERE A.JOB_ID IN
      (SELECT B.JOB_ID
       FROM JOBS B
       WHERE JOB_TITLE IN('Sales Representative','Stick
cleark')
      AND A.SALARY NOT IN(2500, 3500, 7000));

```

11. Cho biết mã nhân viên (`employee_id`), tên nhân viên (`last_name`), lương sau khi tăng thêm 15% so với lương ban đầu, được làm tròn đến hàng đơn vị và đặt lại tên cột là “New Salary”.

```

SELECT EMPLOYEE_ID, LAST_NAME, ROUND(SALARY+((SALARY*15)/100),1) AS
"NEW SALARY"
FROM EMPLOYEES;

```

12. Cho biết tên nhân viên, chiều dài tương ứng của tên đối với những nhân viên có kí tự bắt đầu trong tên là “J”, “A”, “L”, “M”. Kết quả hiển thị tăng dần theo tên, kí tự đầu của tên viết hoa, các kí tự còn lại viết thường. (dùng hàm `INITCAP`, `LENGTH`, `SUBSTR`)

```

SELECT INITCAP(FIRST_NAME), LENGTH(FIRST_NAME)
FROM EMPLOYEES
WHERE SUBSTR(FIRST_NAME,1,1)IN('J','A','L','M')
ORDER BY FIRST_NAME;

```

13. Liệt kê danh sách nhân viên, khoảng thời gian (tính theo tháng) mà nhân

viên đã làm việc trong công ty cho đến nay. Kết quả sắp xếp tăng dần theo số lượng tháng làm việc. (dùng hàm MONTHS_BETWEEN)

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME,
MONTHS_BETWEEN(SYSDATE, HIRE_DATE) AS KHOANG_TG
FROM EMPLOYEES;
```

13.1 TUONG TU CAU 13 NHUNG LAM TRON HANG CHUC

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME,
ROUND(MONTHS_BETWEEN(SYSDATE, HIRE_DATE), 2) AS KHOANG_TG
FROM EMPLOYEES;
```

14. Thực hiện câu truy vấn cho kết quả theo định dạng sau :
<last_name> earns <salary> monthly but wants <3*salary> .
Cột được hiển thị có tên "Dream Salaries"

```
SELECT concat(concat(concat(concat(last_name, ' earns '), salary),
'monthly but wants'), (salary*3)) as Dream_Salaries
FROM EMPLOYEES;
```

15. Liệt kê tên nhân viên, mức hoa hồng nhân viên đó nhận được. Trường hợp nhân viên nào không được hưởng hoa hồng thì hiển thị "No commission". (dùng hàm NVL)

```
SELECT FIRST_NAME, NVL(TO_CHAR(COMMISSION_PCT, '.9'), 'NO COMMISSION')
FROM EMPLOYEES;
```

16. Thực hiện câu truy vấn cho kết quả như sau: (dùng hàm DECODE hoặc CASE...)

```
JOB_ID GRADE
AD_PRES A
ST_MAN B
IT_PROG C
SA_REP D
ST_CLERK E
Không thuộc 0
```

```
select distinct job_id, case job_id when 'AD_PRES' then 'A'
                                when 'ST_MAN' then 'B'
                                when 'IT_PROG' then 'C'
                                when 'SA_REP' then 'D'
                                when 'ST_CLERK' then 'E'
                                else '0' end "GRADE"
```

```
FROM jobs ORDER BY GRADE ;
```

17. Cho biết tên nhân viên, mã phòng, tên phòng của những nhân viên làm việc ở thành phố Toronto.

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, CITY
FROM EMPLOYEES A INNER JOIN
  (SELECT DEPARTMENT_ID, LOCATION_ID FROM DEPARTMENTS) B
  ON A.DEPARTMENT_ID=B.DEPARTMENT_ID INNER JOIN
  (SELECT LOCATION_ID, CITY
   FROM LOCATIONS WHERE CITY='Toronto')C
  ON B.LOCATION_ID=C.LOCATION_ID;
```

18. Liệt kê thông tin nhân viên cùng với người quản lý của nhân viên đó. Kết quả hiển thị: mã nhân viên, tên nhân viên, mã người quản lý, tên người quản lý.

----Câu 18 cách 1

cách 1

```
select e1.employee_id,e1.last_name,e1.first_name, e1.manager_id,
e2.name_manager
from employees e1 join
  (select distinct employee_id, last_name, first_name as
name_manager
   from employees
   where employee_id in(select manager_id from
employees))e2
on e1.manager_id=e2.employee_id;
```

--Câu 18 cách 2

```
select e1.employee_id,e1.last_name,e1.first_name, e1.manager_id,
e2.name_manager
from employees e1 ,(select distinct employee_id,last_name ,first_name
as name_manager
   from employees
   where employee_id in(select manager_id from
employees))e2
where e1.manager_id=e2.employee_id;
```

19. Liệt kê danh sách những nhân viên làm việc cùng phòng.

```
select * from employees
order by department_id;
```


20. Liệt kê danh sách nhân viên được thuê sau nhân viên “Davies”.

```
select *
from EMPLOYEES where hire_date >
      (select hire_date from EMPLOYEES where last_name='Davies');
```

21. Liệt kê danh sách nhân viên được thuê vào làm trước người quản lý của họ.

```
select a.last_name,a.hire_date,b.name_manager,b.hire_date_manager
from (select last_name,hire_date,manager_id from EMPLOYEES) a
     inner join (select employee_id,last_name as
                 name_manager,hire_date as hire_date_manager
                from EMPLOYEES
                where employee_id in(select distinct manager_id from EMPLOYEES))b
on a.manager_id=b.employee_id
where a.hire_date<b.hire_date_manager;
```

22. Cho biết lương thấp nhất, lương cao nhất, lương trung bình, tổng lương của từng loại công việc.

```
SELECT A.DEPARTMENT_ID, C.DEPARTMENT_NAME ,MAX(SALARY), MIN(SALARY),
AVG(SALARY), SUM(SALARY)
FROM (SELECT DEPARTMENT_ID, SALARY FROM EMPLOYEES)A,
      (SELECT DEPARTMENT_ID, DEPARTMENT_NAME FROM DEPARTMENTS)C
WHERE A.DEPARTMENT_ID=C.DEPARTMENT_ID
AND A.DEPARTMENT_ID IS NOT NULL
GROUP BY A.DEPARTMENT_ID, C.DEPARTMENT_NAME;
```

23. Cho biết mã phòng, tên phòng, số lượng nhân viên của từng phòng ban.

```
SELECT a.DEPARTMENT_ID,b.DEPARTMENT_NAME, COUNT(*)
FROM (SELECT DEPARTMENT_ID FROM EMPLOYEES) a inner join
      (select DEPARTMENT_ID,DEPARTMENT_NAME FROM DEPARTMENTS) b
on a.DEPARTMENT_ID=b.DEPARTMENT_ID
group by a.DEPARTMENT_ID, b.DEPARTMENT_NAME
order by a.DEPARTMENT_ID;
```

24. Cho biết tổng số nhân viên, tổng nhân viên được thuê từng năm 1995, 1996, 1997, 1998.

```
SELECT to_char(hire_date, 'YYYY') as Nam, count(*)
from EMPLOYEES
where to_char(hire_date, 'YYYY') in ('1996', '1997', '1998', '1995')
group by to_char(hire_date, 'YYYY');
```

25. Liệt kê tên, ngày thuê của những nhân viên làm việc cùng phòng với nhân viên “Zlotkey”.

```
select last_name, hire_date from EMPLOYEES
where department_id = (select department_id
from EMPLOYEES
where last_name=INITCAP('zlotkey'));
```

26. Liệt kê tên nhân viên, mã phòng ban, mã công việc của những nhân viên làm việc cho phòng ban đặt tại vị trí (location_id) 1700.

```
select a.last_name, a.department_id, a.job_id, c.location_id
from EMPLOYEES a inner join (select department_id, location_id from
DEPARTMENTS) b on
a.department_id=b.department_id
inner join (select location_id from LOCATIONS where location_id=1700)c
on b.location_id=c.location_id;
```

27. Liệt kê danh sách nhân viên có người quản lý tên “King”.

```
select * from EMPLOYEES
where manager_id in (select employee_id from EMPLOYEES where
last_name=INITCAP('king'));
```

28. Liệt kê danh sách nhân viên có lương cao hơn mức lương trung bình và làm việc cùng phòng với nhân viên có tên kết thúc bởi “n”.

```
select * from EMPLOYEES
where department_id in(select department_id
from employees
where last_name like ('%n'))
and salary > (select avg(salary) from EMPLOYEES);
```


29. Liệt kê danh sách mã phòng ban, tên phòng ban có ít hơn 3 nhân viên.

```
select department_id,department_name
from DEPARTMENTS
where department_id in (select department_id from employees )
group by department_id
having count(*)<3);
```

30. Cho biết phòng ban nào có đông nhân viên nhất, phòng ban nào có ít nhân viên nhất.

```
select a.department_id,b.department_name,count(*)
from EMPLOYEES a inner join (select department_id,department_name
from DEPARTMENTS) b on a. department_id=b.department_id
group by a.department_id,b.department_name
having count(*) >= all (select count(*)
from EMPLOYEES
group by department_id) or count(*) <= all
(select count(*) from EMPLOYEES group by department_id);
```

31. Liệt kê danh sách nhân viên được thuê vào ngày có số lượng nhân viên được thuê đông nhất. (dùng hàm TO_CHAR(hire_date, "Day").

```
select first_name,last_name, a.Ngay
from (select first_name,last_name,to_char(hire_date,'day') Ngay
from EMPLOYEES) a inner join (select to_char(hire_date,'day') Ngay,
count(to_char(hire_date,'day'))
from EMPLOYEES
group by to_char(hire_date,'day'))
having count(to_char(hire_date,'day')) >= all
(select count(to_char(hire_date,'day'))
from EMPLOYEES
group by to_char(hire_date,'day')) b
on a.Ngay=b.Ngay;
```

32. Liệt kê thông tin 3 nhân viên có lương cao nhất.

```
select * from (select * from EMPLOYEES order by salary desc)
where rownum<4;
```

33. Liệt kê danh sách nhân viên đang làm việc ở tiểu bang “California”.

```
select *
from EMPLOYEES a inner join (select department_id, location_id from
DEPARTMENTS) b on a.department_id=b.department_id
inner join (select location_id from LOCATIONS
where state_province='California') c
on b.location_id=c.location_id;
```

34. Cập nhật tên của nhân viên có mã 3 thành “Drexler”.

```
UPDATE EMPLOYEES
SET FIRST_NAME='Drexler'
WHERE EMPLOYEE_ID=3;
```

35. Liệt kê danh sách nhân viên có mức lương thấp hơn mức lương trung bình của phòng ban mà nhân viên đó làm việc.

```
select first_name,last_name,salary
from EMPLOYEES a
inner join (select department_id,avg(salary) LuongTrungBinh
from EMPLOYEES group by department_id) b
on a.department_id=b.department_id
where salary<LuongTrungBinh;
```

36. Tăng thêm 100\$ cho những nhân viên có lương nhỏ hơn 900\$.

```
UPDATE EMPLOYEES
SET SALARY = SALARY + 100
WHERE SALARY < 900;
```

37. Xóa phòng ban 500.

```
DELETE FROM DEPARTMENTS
WHERE DEPARTMENT_ID=500;
```

38. Xóa phòng ban nào chưa có nhân viên.

```
DELETE
FROM DEPARTMENTS
WHERE DEPARTMENT_ID NOT IN (SELECT DEPARTMENT_ID FROM EMPLOYEES);
```

TẠO VIEW

39. Tạo view chứa thông tin của những quốc gia ở vùng Asia.

```
create or replace view cau39
as
  select country_name
  from countries
  where region_id in
      (
        select region_id
        from regions
        where region_name='Asia'
      );
```

40. Tạo view chứa danh sách nhân viên không có người quản lý.

```
create or replace view cau40
as
  select employee_id, first_name, last_name, manager_id
  from employees
  where manager_id is null;
```

41. Tạo view chứa danh sách phòng ban chưa có nhân viên.

```
create or replace view cau41
as
  select department_id
  from DEPARTMENTS
  minus
  select department_id
  from EMPLOYEES ;
```

42. Tạo view chứa mã nhân viên, tên nhân viên, tên phòng, mã công việc, số năm làm việc, lương của những nhân viên có mức lương lớn hơn mức lương trung bình của công ty.

```
create view as cau42
as
select e1.employee_id, e1.first_name,e1.job_id, salary,
a.department_name
from employees e1, departments a
where e1.department_id=a.department_id;
```

43. Liệt kê các mã phòng ban(department_id) không tồn tại trong bảng nhân viên(employees).

```
select department_id
from departments dept
where not exists (select null from employees emp
where emp.department_id = dept.department_id );
```

TẠO STORE PROCEDURE

43. Tạo thủ tục có tên là dept_info cho biết thông tin về phòng ban với tham số truyền vào là mã phòng ban.

```
create or replace procedure dept_info(v_department_id number,
ten out departments.department_name%type)
as
begin
select department_name into ten
from departments
where department_id=v_department_id;
dbms_output.put_line('Ten phong ban: '||ten);
exception when no_data_found
then dbms_output.put_line('Khong co phong ban');
end;

--Thuc thi
set serveroutput on
declare ten departments.department_name%type;
begin
DEPT_INFO(&v_department_id, ten);
end;
```

44. Tạo thủ tục có tên là add_job thêm một công việc mới với tham số truyền vào là mã công việc, tên công việc.

```
create or replace procedure add_job (v_macv JOBS.job_id%TYPE, v_tencv
JOBS.job_title%TYPE)
as
v_macv_temp JOBS.job_id%TYPE;
v_loi EXCEPTION;
begin
    select job_id into v_macv_temp
    from JOBS
    where job_id=v_macv;
    if v_macv_temp is not null then
        raise v_loi;
    end if;
exception when v_loi then
    dbms_output.put_line('Khong them duoc');
    when no_data_found then
    insert into JOBS (job_id,job_title) values (v_macv,v_tencv);
        dbms_output.put_line('Cong viec ' || v_tencv || ' da duoc
them.');
```

end;

```
--thuc thi
set serveroutput on
execute add_job('p_code','Lap Trinh Vien');
```

45. Tạo thủ tục có tên là update_comm cập nhật hoa hồng cho nhân viên tăng thêm 5% hoa hồng ban đầu, tham số truyền vào là mã nhân viên.

```
create or replace procedure update_comm2(manhanvien number)
as
begin
    update employees
    set commission_pct = commission_pct*1.05
    where employee_id=manhanvien;
    dbms_output.put_line('Cap nhat hoa hong cua ' || manhanvien || '
thanh cong!');
```

end;

```
--thuc thi
set serveroutput on
execute update_comm2(3);
```

46. Tạo thủ tục có tên là add_emp thêm một nhân viên mới với tất cả các giá trị là tham số truyền vào.

```
create or replace procedure add_emp(v_employee_id number
,first_name varchar2, last_name varchar2, email varchar2, phone_number
```

```

varchar2,hire_date employees.hire_date%type, job_id varchar2, salary
number, commission_pct number
, manager_id number, department_id employees.department_id%type)
as
  v_count number;
  v_loi exception;
begin
  select count(*) into v_count
  from employees
  where employee_id=v_employee_id;
  if v_count > 0 then raise v_loi;
  else
    insert into employees values(v_employee_id, first_name, last_name,
email
    , phone_number, hire_date, job_id, salary, commission_pct,
manager_id,
    department_id);
    dbms_output.put_line('Them thanh cong!');
  end if;
  exception when v_loi then
    dbms_output.put_line('Nhan vien da ton tai');
end;

```

47. Tạo thủ tục có tên là delete_emp xóa một nhân viên mới với mã nhân viên là tham số truyền vào.

```

create or replace procedure delete_employee(manhanvien number)
as
begin
  delete from employees where employee_id=manhanvien;
  dbms_output.put_line('Da xoa' || manhanvien || 'thanh cong!');
end;

```

48. Tạo thủ tục có tên find_emp tìm kiếm nhân viên có lương lớn hơn mức lương thấp nhất

```

create or replace procedure find_emp
as
cursor c_nhanvien is
select employee_id, first_name,last_name
from EMPLOYEES a
where salary>(select min_salary from jobs b where a.job_id=b.job_id)
and salary<(select max_salary from jobs c where a.job_id=c.job_id);
v_emp_id EMPLOYEES.employee_id%TYPE;
v_first_name EMPLOYEES.first_name%TYPE;
v_last_name EMPLOYEES.last_name%TYPE;

```

```

begin
  for r_nhanvien in c_nhanvien
  loop
    v_emp_id := r_nhanvien.employee_id;
    v_first_name := r_nhanvien.first_name;
    v_last_name:=r_nhanvien.last_name;
    dbms_output.put_line('Ma NV: ' || v_emp_id || ' TenNV: ' ||
v_first_name || ' ' || v_last_name);
  end loop;
end;
--Thuc thi
set serveroutput on

execute find_emp;

```

49. Tạo thủ tục có tên update_comm cập nhật lương của nhân viên với điều kiện nhân viên nào làm việc trên 2 năm thì tăng lương thêm 200\$, nhân viên làm việc trên 1 năm và dưới 2 năm thì tăng lương thêm 100\$, nhân viên nào làm việc đúng 1 năm thì tăng 50\$, còn lại không tăng.

```

create or replace procedure update_comm
as
v_sonam_lamviec varchar2(2);
cursor c_nhanvien is
select * from EMPLOYEES;
begin
  for r_nhanvien in c_nhanvien
  loop
    v_sonam_lamviec := to_char(sysdate, 'yyyy')-
to_char(r_nhanvien.hire_date, 'yyyy');
    if v_sonam_lamviec >= 2 then
      update EMPLOYEES
      set salary=salary+200
      where employee_id=r_nhanvien.employee_id;
      dbms_output.put_line('Nhan vien : ' ||
r_nhanvien.last_name || ' da duoc tang 200$');
    else
      update EMPLOYEES
      set salary=salary+100
      where employee_id=r_nhanvien.employee_id;
      dbms_output.put_line('Nhan vien : ' ||
r_nhanvien.last_name || ' da duoc tang 100$');
    end if;
  end loop;
end;

```

```
--thuc thi
set serveroutput on
execute update_comm;
```

TẠO FUNCTION

```
--51
create or replace function sum_salary(maphongban in number)
return number
as
  v_salary number;
begin
select sum(salary)
into v_salary
from employees
where department_id=maphongban;
return v_salary;
  exception
when no_data_found
then
return('Du lieu khong tim thay');
when others then
return('loi ham');
end;
--Test
set serveroutput on
  show error;
select column_name, data_type, data_length
from user_tab_columns
where table_name='EMPLOYEES';
--
SELECT DEPARTMENT_ID FROM DEPARTMENTS;
set serveroutput on
set verify off
execute dbms_output.put_line('Tong luong la: '||sum_salary(&maphong));
```

```
52. Tao ham co ten name_con cho biet ten quoc gia voi ma quoc gia la
tham so
--truyen vao
```



```

--Xem kieu du lieu

select column_name, data_type, data_length
from user_tab_columns
where table_name='COUNTRIES';
--52
create or replace function name_con
(maqq in countries.country_id%type)
return varchar2
is
tenqq varchar2(50);
begin
    select country_name
    into tenqq
    from countries
    where country_id=maqq;
    return tenqq;
exception
when no_data_found
then
return('Du lieu khong tim thay');
when others then
return('loi ham');
end;

--Thuc thi
SELECT COUNTRY_ID FROM COUNTRIES;
--Goi thuc thi 1
set serveroutput on
set verify off
declare quocgia varchar2(35);
begin
    quocgia:=name_con('AU');
    dbms_output.put_line(quocgia);
end;
--Goi thuc thi 2
--53
create or replace function annual_comp
(luong employees.salary%type,
hoahong employees.commission_pct%type)
return number
as
thunhap number;
begin
    thunhap:=luong*12+(hoahong*luong*12);

```

```
        return thunhap;
    exception
when no_data_found
then
return('Du lieu khong tim thay');
when others then
return('loi ham');
end;
--Goi ham
select salary, commission_pct from EMPLOYEES;
--
set serveroutput on
set verify off
declare ThuNhap number;
begin
    ThuNhap:=annual_comp(14200,0.4);
    dbms_output.put_line(ThuNhap);
end;
--54
Create or replace function avg_salary
(mapb employees.department_id%type)
return number
as
luongtb number;
begin
    select avg(salary)
    into luongtb
    from employees
    where department_id=mapb;
    return luongtb;
    exception
when no_data_found
then
return('Du lieu khong tim thay');
when others then
return('loi ham');
end;
--Goi thuc thi
set serveroutput on;
set verify off
declare LuongTB number;
begin
    LuongTB:=avg_salary(110);
    dbms_output.put_line(LuongTB);
end;
```

```
--55
```

```
Create or replace function Time_work(MaNhanVien
EMPLOYEES.EMPLOYEE_ID%TYPE)
return number
as
    tglamviec number;
begin
select round(months_between(to_date(sysdate, 'dd/mm/yyyy'),
    to_date(hire_date, 'dd/mm/yyyy')),1)
into tglamviec
from EMPLOYEES
where employee_id=MaNhanVien;
return tglamviec;
    exception when no_data_found then
return('Du lieu khong tim thay');
end;
--Goi thuc thi
set serveroutput on
declare tg number(22);
begin
tg:=Time_work(&manv);
    dbms_output.put_line('So thang lam viec cua nhan vien la '||tg);
end;

show error
--TEST
select*from employees;
--xem kieu du lieu cua cot
select column_name,data_type, data_length
from user_tab_columns
where table_name='EMPLOYEES'

select round(months_between
(to_date(sysdate, 'dd/mm/yyyy'),to_date(hire_date, 'dd/mm/yyyy')),2)
from employees
```

TẠO TRIGGER

60. Cài đặt ràng buộc toàn vẹn ngày thuê nhân viên phải nhỏ hơn hoặc bằng ngày hiện hành khi thêm mới hoặc cập nhật thông tin về nhân viên.

```
create or replace trigger tr_ngaythue
after insert or update
on EMPLOYEES
for each row
declare v_ngaythue EMPLOYEES.HIRE_DATE%TYPE;
begin
    if(v_ngaythue>sysdate) then
        raise_application_error(-20020, 'Ngày thuê không hợp lệ');
    end if;
end;
```

61. Cài đặt ràng buộc toàn vẹn min_salary luôn nhỏ hơn max_salary khi thêm mới hoặc cập nhật thông tin bảng công việc

```
create or replace trigger tr_luong
before insert or update
on jobs
for each row
begin
    if(:new.min_salary>:new.max_salary)then
        raise_application_error(-20022, 'Luong nhập vào không hợp lệ');
    end if;
end;
```

62. Cài đặt ràng buộc toàn vẹn ngày bắt đầu luôn nhỏ hơn hoặc bằng ngày kết thúc khi thêm mới hoặc cập nhật thông tin bảng job_history.

```
create or replace trigger tr_Ngay
before insert or update
on job_history
for each row
begin
    if(:new.start_date>:new.end_date) then
        raise_application_error(-20021, 'Ngày bắt đầu phải nhỏ hơn ngày
ket thuc');
    end if;
end;
```

63. Cài đặt ràng buộc toàn vẹn lương và hoa hồng của nhân viên phải tăng chứ không giảm khi cập nhật nhân viên.

```
create or replace trigger tr_Luong_HoaHong
before update
on employees
for each row
begin
    if (:new.salary < :old.salary) then
        raise_application_error(-20022, 'Luong cap nhat phai lon hon
luong hien tai');
    end if;
    if (:new.commission_pct < :old.commission_pct) then
        raise_application_error(-20022, 'Hoa hong cap nhat phai lon hon
luong hien tai');
    end if;
end;
```

TẠO PACKAGE

```

create or replace package emp_info
as
procedure salary_table(manv employees.employee_id%TYPE);
function sum_salary (mapb departments.department_id%TYPE)
return number;
end;

create or replace package body emp_info
as
procedure salary_table(manv employees.employee_id%TYPE)
as
luong employees.salary%TYPE;
begin
select salary into luong from employees where employee_id=manv;
dbms_output.put_line('luong cua nhan vien nay la:' || luong);
exception
when no_data_found then
dbms_output.put_line('ko tim thay nhan vien nay');
end salary_table;

function sum_salary(mapb departments.department_id%TYPE)
return number
as
tongluong number;
begin
select sum(salary) into tongluong from employees where
department_id=mapb;
return tongluong;
exception
when no_data_found then
dbms_output.put_line('ko tim thay phong ban');
end sum_salary;
end emp_info;

thuc thi
set serveroutput on;
declare
v_luong number;
v_tongluong number;
begin
emp_info.salary_table(198);
v_tongluong:=emp_info.sum_salary(110);

```

```

dbms_output.put_line(v_luong);
dbms_output.put_line(v_tongluong);
end;

```

58. Tạo package có tên là job_pack chứa 3 thủ tục tên là add_job để thêm công việc, update_job để cập nhật công việc, del_job để xóa công việc và 1 hàm có tên q_job để tìm kiếm tên công việc theo mã.

```

create or replace package job_pack
as
  procedure add_job
  (
    macv jobs.job_id%type,
    tencv jobs.job_title%type,
    luongthapnhat jobs.min_salary%type,
    luongcaonhat jobs.max_salary%type
  );
  procedure update_job
  (
    macv jobs.job_id%type,
    tencv jobs.job_title%type,
    luongthapnhat jobs.min_salary%type,
    luongcaonhat jobs.max_salary%type
  );
  procedure del_job
  (
    macv jobs.job_id%type
  );
end job_pack;
--phan than
create or replace package body job_pack
as
  procedure add_job
  (
    macv jobs.job_id%type,
    tencv jobs.job_title%type,
    luongthapnhat jobs.min_salary%type,
    luongcaonhat jobs.max_salary%type

```

```

)
is
begin
insert into JOBS
values (macv,tencv,luongthapnhat, luongcaonhat);
dbms_output.put_line('Cong viec '||tencv||' da
duoc them');
exception when no_data_found then
dbms_output.put_line('Khong tim thay cong viec');
end add_job;
procedure update_job
(
macv jobs.job_id%type,
tencv jobs.job_title%type,
luongthapnhat jobs.min_salary%type,
luongcaonhat jobs.max_salary%type
)
is
begin
update jobs set job_title=tencv,min_salary=luongthapnhat
, max_salary=luongcaonhat where job_id=macv;
dbms_output.put_line('CAP NHAT THANH CONG');
end update_job;
procedure del_job(macv jobs.job_id%type)
is
begin
delete from jobs where job_id=macv;
dbms_output.put_line('XOA THANH CONG');
end del_job;
end job_pack;

--Thuc thi
--add_job
begin
job_pack.add_job('ADMIN2','ADMINISTRATOR2',20000,60000);
end;
--update_job
begin
job_pack.update_job('ADMIN2','AA',21000,61000);

```



```

end;
--del_job cach goi 1
EXECUTE job_pack.del_job('ADMIN')
--del_job cach goi 2
begin
    job_pack.del_job('ADMIN3');
end;

```

59. Tạo package có tên emp_pack chứa một thủ tục tên new_emp thêm một nhân viên mới với tất cả các tham số truyền vào và một hàm tên valid_deptid kiểm tra mã phòng ban hợp lệ, khi mã phòng hợp lệ mới được phép thêm nhân viên.

```

create or replace package emp_pack
as
    procedure new_emp
    (
        MaNV employees.employee_id%type,
        TenNV employees.first_name%type,
        HoNV employees.last_name%type,
        Email employees.email%type,
        DienThoai employees.phone_number%type,
        NgayThue employees.hire_date%type,
        MaCV employees.job_id%type,
        Luong employees.salary%type,
        HoaHong employees.commission_pct%type,
        MaQuanLy employees.manager_id%type,
        MaPhong employees.department_id%type
    );
    function valid_deptid(i_department_id in number)
    return boolean;
end emp_pack;
--phan than
create or replace package body emp_pack
as

    procedure new_emp
    (
        MaNV employees.employee_id%type,
        TenNV employees.first_name%type,

```

```

HoNV employees.last_name%type,
Email employees.email%type,
DienThoai employees.phone_number%type,
NgayThue employees.hire_date%type,
MaCV employees.job_id%type,
Luong employees.salary%type,
HoaHong employees.commission_pct%type,
MaQuanLy employees.manager_id%type,
MaPhong employees.department_id%type
)
is
begin
    insert into employees values(MaNv, TenNV, HoNV, Email,
DienThoai,
    NgayThue, MaCV, Luong, HoaHong, MaQuanLy, MaPhong);
end new_emp;
--ket thuc proc new_emp
function valid_deptid(i_department_id in number)
return boolean
is
    v_id_dept number;
begin
    select count(*) into v_id_dept
    from departments
    where department_id=i_department_id;
    return 1=v_id_dept;
exception when others then
    return false;
end valid_deptid;--ket thuc proc valid_deptid
end emp_pack;

--goi thuc thi
set serveroutput on
begin
    if(emp_pack.valid_deptid(&i_department_id)) then
        emp_pack.new_emp(1, 'First',
'Last', 'first.last@oracle.com',
            '(123)123-1234', '18-JUN-02', 'IT_PROG', 900,00,
100,110);

```

```
        dbms_output.put_line('Them thanh cong');  
    else  
        dbms_output.put_line('Ma phong ban nay khong ton  
tai!');  
    end if;  
end;
```

QUẢN TRỊ NGƯỜI DÙNG

Quản trị người dùng.

64. Tạo không gian bảng (tablespace) có kích thước 100M.
65. Tạo không gian bảng tạm (temporary tablespace) có kích thước 50M.
66. Tạo rollback segment rolora để truy xuất đồng thời cho table space vừa tạo.
67. Tạo user có tên là tên sinh viên, mật khẩu do sinh viên tự đặt với tablespace và temporary tablespace vừa tạo.
68. Cấp quyền truy xuất tài nguyên (resource) cho user vừa tạo.
69. Cấp quyền cho phiên làm việc (session) cho user vừa tạo.
70. Cấp quyền tạo bảng (table) cho user vừa tạo.
71. Cấp quyền tạo khung nhìn (view) cho user vừa tạo.
72. Cấp quyền Select, Insert, Update, Delete trên bảng Employees cho user vừa tạo.
73. Cấp quyền Select, Insert, Update, Delete trên bảng Departments cho user vừa tạo.
74. Cấp quyền Select chỉ với các thuộc tính job_id, job_title trên bảng Jobs cho user vừa tạo.
75. Login vào csdl HR với user vừa tạo.
76. Truy vấn các bảng trong csdl HR và cho nhận xét.
77. Cho biết các user hiện có từ view dba_users.
78. Đăng nhập với quyền hệ thống và tạo user có tên là mã sinh viên, mật khẩu là tên sinh viên
 - a. Thay đổi mật khẩu của user.
 - b. Cấp quyền đăng nhập csdl
 - c. Truy xuất view v\$\$session để xem phiên làm việc.

d. Tao bảng phòng ban gồm 2 thuộc tính: mã phòng ban, tên phòng ban và cho nhận xét câu c.

e. Cấp phát hạn ngạch (quota) 20M cho user vừa tạo.

f. Tao lại bảng Phòng ban và cho nhận xét.

g. Cấp phát không gian giới hạn tablespace cho user vừa tạo

h. Cấp phát tài nguyên cho user

i. Cấp phát phiên làm việc cho user

j. Cấp quyền đăng nhập OEM

79. Truy xuất vào view hệ thống dba_profiles.

80. Tao profile giới hạn việc truy xuất tài nguyên và cấp cho user vừa tạo:

a. Thời gian kết nối 120s

b. Số lần thất bại khi cho phép kết nối là 3.

c. Thời gian chờ kết nối là 60s

d. Phiên làm việc cho mỗi user là 2

e. Gán profile cho user vừa tạo

f. Thay đổi thông số của profile với phiên làm việc cho mỗi user là 4.

g. Số lần thất bại khi cho phép kết nối là 3

h. Thiết lập thông số để profile có hiệu lực.

i. Xóa profile vừa tạo.

81. Oracle Database Resource Manager(ODRM) để tạo vùng treo, tao nhóm người dùng, tao kế hoạch tài nguyên và định hướng kế hoạch, kiểm tra tính hợp lệ của vùng treo và gởi vùng treo:

a. Tạo vùng treo.

b. Xóa vùng treo.

c. Kiểm tra nhóm người dùng nào đã tồn tại và cho nhận xét liên quan các nhóm người dùng.

d. Kiểm tra kế hoạch của mỗi CSDL từ dba_rsrc_plans

e. Tao 4 nhóm người dùng: cập nhật, tìm kiếm, thống kê, báo cáo

- f. Kiểm tra tính hợp lệ của vùng treo.
- g. Truy xuất view dba_users và cho nhận xét.
- h. Gán người dùng vừa tạo tới nhóm người dùng.
- i. Truy xuất view dba_users và cho nhận xét.
- j. Khởi tạo user cho nhóm người dùng
- k. Truy xuất view dba_users và cho nhận xét.
- l. Tạo kế hoạch tài nguyên
- m. Tạo định hướng kế hoạch và giới hạn tài nguyên cho từng người dùng (CPU_1 nhóm cập nhật là 60, CPU_1 nhóm thống kê 40, CPU_2 nhóm báo cáo là 30, nhóm khác là 100).
- n. Gỡ vùng treo.
- o. Truy xuất view dba_rsrc_plan_directives và cho nhận xét.
- p. Kích hoạt kế hoạch sử dụng tài nguyên
- q. Truy xuất view v\$rsrc_consumer_group để xem việc sử dụng tài nguyên giữa các nhóm
- r. Truy xuất view v\$rsrc_plan và cho nhận xét.

--64

```
create tablespace oracle
datafile'oracle.dbf' size 100m;
```

--65

```
create temporary tablespace oracle_2
tempfile'oracle2.dbf' size 50m;
```

--66

```
create rollback segment seg
tablespace oracle;
```

```
--67
create user sinhvien identified by sinhvien
default tablespace oracle
temporary tablespace oracle_2
password expire;

--68
grant resource to sinhvien;

--69
grant create session to sinhvien;

--70
grant create table to sinhvien;

--71
grant create view to sinhvien;

--72
grant insert, update, select, delete on hr.employees to
sinhvien;

--73
grant insert, select, update, delete on hr.departments to
sinhvien;

--74
grant update(job_id, job_title) on hr.jobs to sinhvien;

--75
```

```
conn hr/hr;
```

```
--76
```

```
Khong truy van duoc vi chua cap quyen truy van
```

```
--77
```

```
select username from dba_users;
```

```
--78
```

```
conn system/system;
```

```
grant create user to hr;
```

```
conn hr/hr;
```

```
create user masinhvien identified by masinhvien
```

```
password exprire;
```

```
--78bis xoa user masinhvien
```

```
conn system/system;
```

```
drop user masinhvien;
```

```
--78b
```

```
create create connect to masinhvien;
```

```
--78c
```

```
select username, status from v$session;
```

```
--78 d Khong tao duoc vi chua cap quyen
```

```
--78e
```

```
create user sinhvien identified by sinhvien
```

```
default tablespace oracle
```



```
temporary tablespace oracle_2
quota 100m on oracle;
--80 a, b, c, d
create profile giang limit
connect_time 120
failed_login_attempts 3
idle_time 60
sessions_per_user 2;
--80e
alter user sinhvien identified by sinhvien
profile giang;
--80h
thiet lap thong so de profile co hieu luc
grant create profile to sinhvien;
--80f,g
alter profile giang limit
sessions_per_user 4
failed_login_attempts 3
--80i
drop profile giang cascade;
--81 a.Tao vung treo
exec dbms_resource_manager.create_pending_area;
```

```
--81 b.xoa vung treo
exec dbms_resource_manager.clear_pending_area;

--81 e
exec dbms_resource_manager.create_pending_area;

exec
dbms_resource_manager.create_consumer_group('capnhat','nhom
cap nhat');

exec
dbms_resource_manager.create_consumer_group('timkiem','nhom
timkiem');

exec
dbms_resource_manager.create_consumer_group('thongke','nhom
thongke');

exec
dbms_resource_manager.create_consumer_group('baocao','nhom
baocao');

--81 f

--kiem tra tinh hop le cua vung treo
exec dbms_resource_manager.validate_pending_area

--81 h
grant capnhat to sinhvien;
grant timkiem to sinhvien;
grant thongke to sinhvien;
grant baocao to sinhvien;

--81 i
```

```
select username, account_status from dba_users
--81 h. Gán người dùng vừa tạo tới nhóm người dùng

conn system/system

exec
dbms_resource_manager_privs.grant_system_privilege('SYSTEM',
'ADMINISTER_RESOURCE_MANAGER',TRUE);

exec dbms_resource_manager.create_pending_area

exec dbms_resource_manager.create_consumer_group('sv','nhom
sv','ROUND-ROBIN'

--khong chay dc

exec
dbms_resource_manager_privs.grant_switch_consumer_group('sin
hvien','sv',true)

--81 l

exec dbms_resource_manager.create_pending_area

exec
dbms_resource_manager.create_plan('kehoachtainguyen','ke
hoach tai nguyen cho nguoi dung')

--81 m

exec
dbms_resource_manager.create_plan_directive('kehoachtainguye
n','capnhat',60,100,100,100,100,100,100)

exec
dbms_resource_manager.create_plan_directive('kehoachtainguye
n','thongke',40,100,100,100,100,100,100,100)
```

```
exec
dbms_resource_manager.create_plan_directive('kehoachtainguye
n', 'baocao', 30, 100, 100, 100, 100, 100, 100, 100)

--81 n gui vung treo

exec dbms_resource_manager.create_pending_area
exec dbms_resource_manager.submit_pending_area

--81 o

select plan, cpu_p1, cpu_p2, cpu_p3 from
dba_rsrc_plan_directives

--Tao role

create role capnhat;
create role timkiem;
create role thongke;
create role baocao;

--cap quyen

grant capnhat to sinhvien;

--Xoa role

revoke capnhat from sinhvien;
```

THE END