Name: Urooba Gohar

Roll No: 22P-9216

Section: BSCS-6A

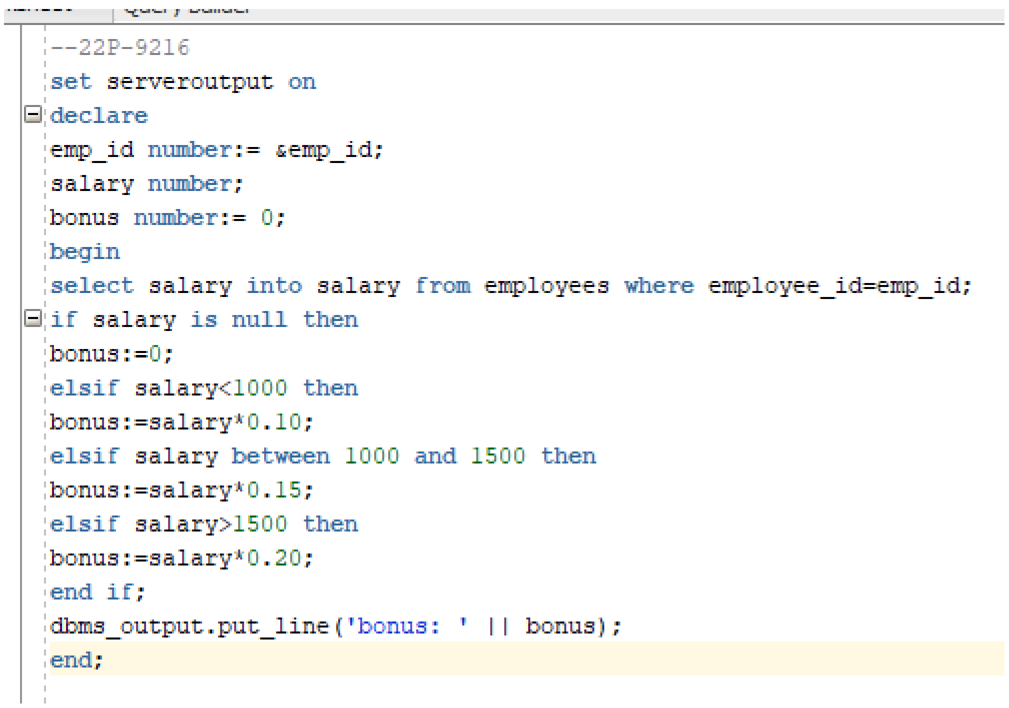
Database Labtask 7

Question 1:

Create a PL/SQL block that computes and prints the bonus amount for a given Employee based on the employee's salary. Accept the employee number as user input with a SQL*Plus substitution Variable.

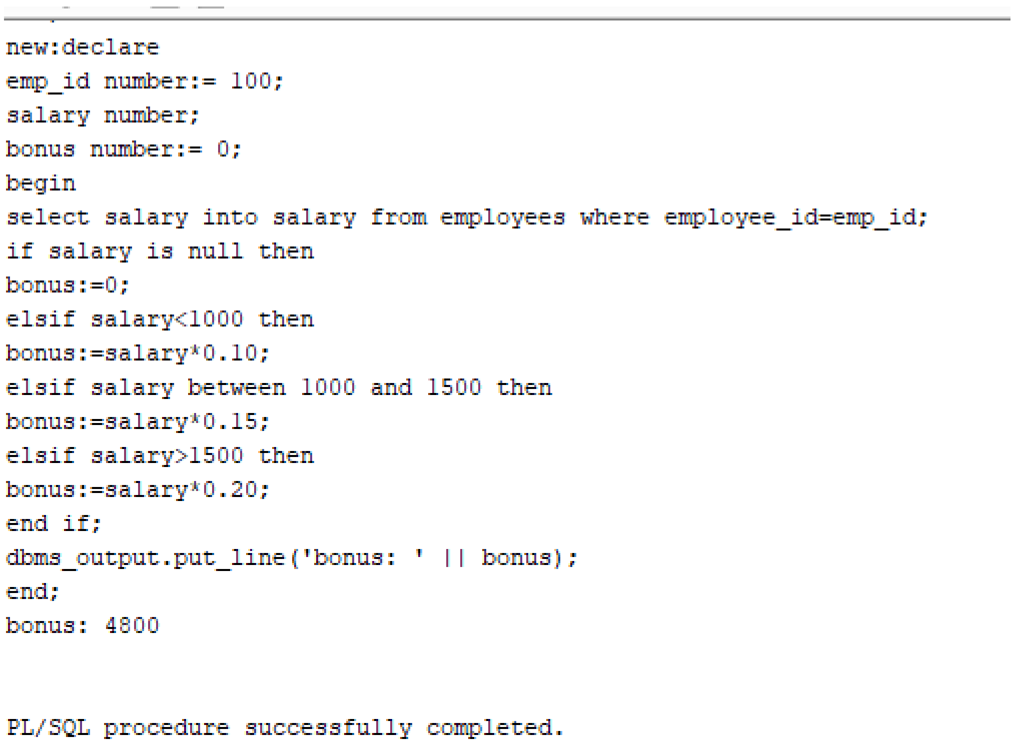
a. If the employee's salary is less than 1,000, set the bonus amount for the Employee to 10% of the salary.
b. If the employee's salary is between 1,000 and 1,500, set the bonus amount for the employee to 15% of the salary.
c. If the employee's salary exceeds 1,500, set the bonus amount for the employee to 20% of the salary.
d. If the employee's salary is NULL, set the bonus amount for the employee to 0.

Answer:



```
--22P-9216
set serveroutput on
declare
emp_id number:= &emp_id;
salary number;
bonus number:= 0;
begin
select salary into salary from employees where employee_id=emp_id;
if salary is null then
bonus:=0;
elsif salary<1000 then
bonus:=salary*0.10;
elsif salary between 1000 and 1500 then
bonus:=salary*0.15;
elsif salary>1500 then
bonus:=salary*0.20;
end if;
dbms_output.put_line('bonus: ' || bonus);
end;
```

Output:

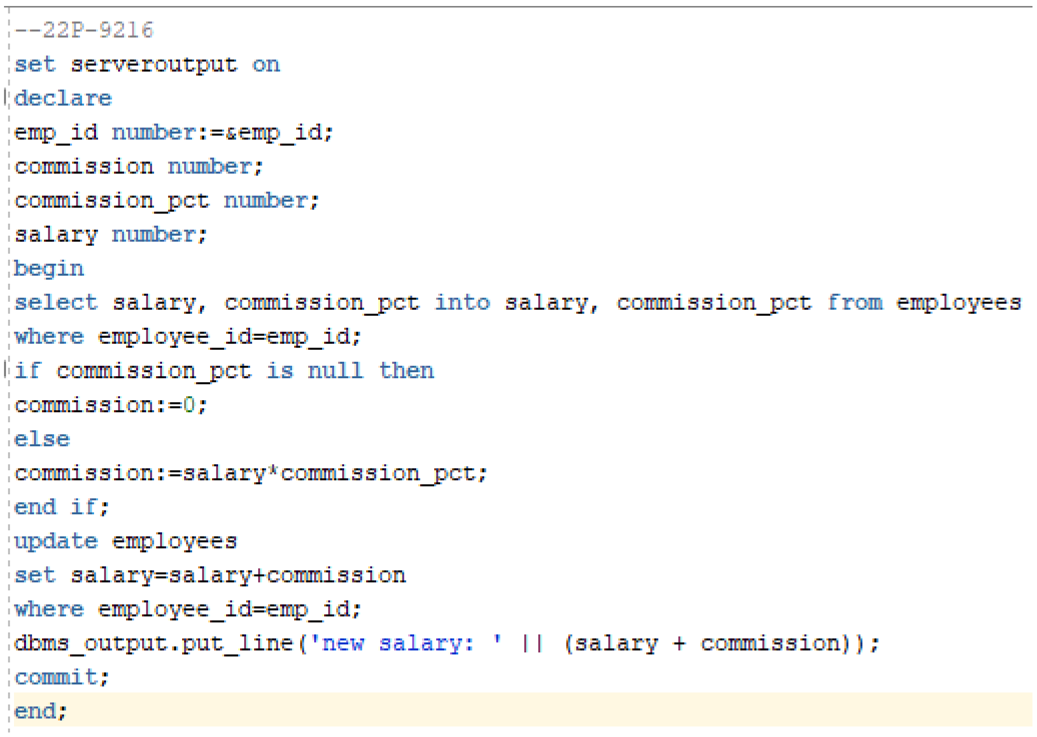


```
new:declare
emp_id number:= 100;
salary number;
bonus number:= 0;
begin
select salary into salary from employees where employee_id=emp_id;
if salary is null then
bonus:=0;
elsif salary<1000 then
bonus:=salary*0.10;
elsif salary between 1000 and 1500 then
bonus:=salary*0.15;
elsif salary>1500 then
bonus:=salary*0.20;
end if;
dbms_output.put_line('bonus: ' || bonus);
end;
bonus: 4800


PL/SQL procedure successfully completed.
```

Question 2:

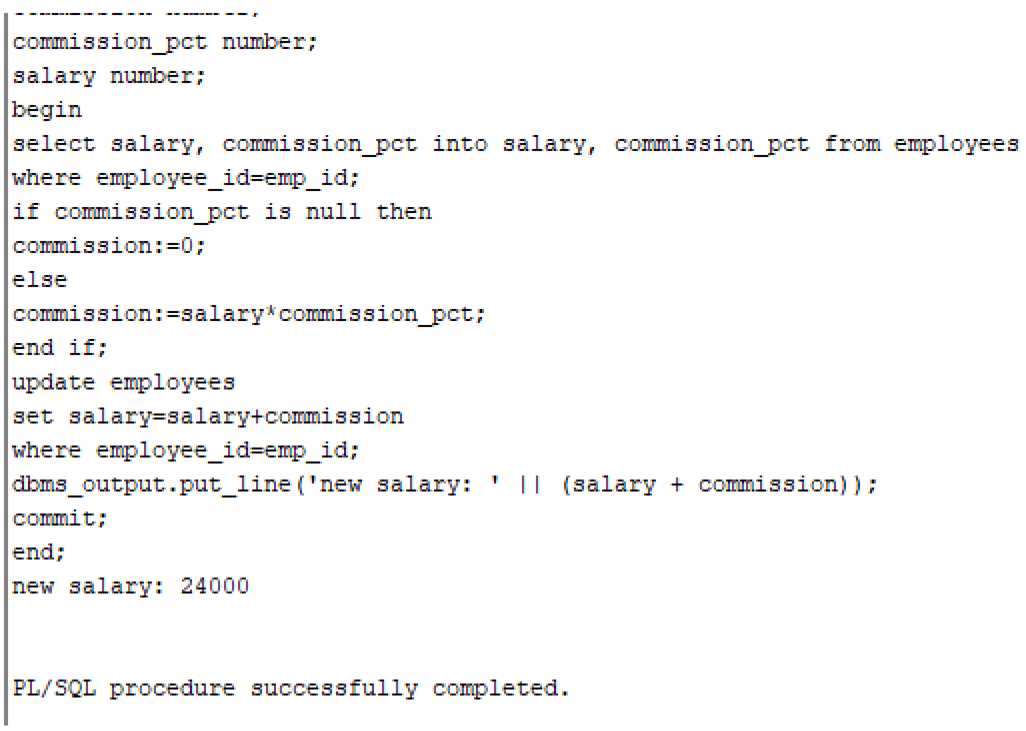
Write a pl/sql block in sql that ask a user for employee id than it checks its commission if commission is null than it updates salary of that employee by adding commission into salary.

Answer:



```
--22P-9216
set serveroutput on
declare
emp_id number:=&emp_id;
commission number;
commission_pct number;
salary number;
begin
select salary, commission_pct into salary, commission_pct from employees
where employee_id=emp_id;
if commission_pct is null then
commission:=0;
else
commission:=salary*commission_pct;
end if;
update employees
set salary=salary+commission
where employee_id=emp_id;
dbms_output.put_line('new salary: ' || (salary + commission));
commit;
end;
```

Output:

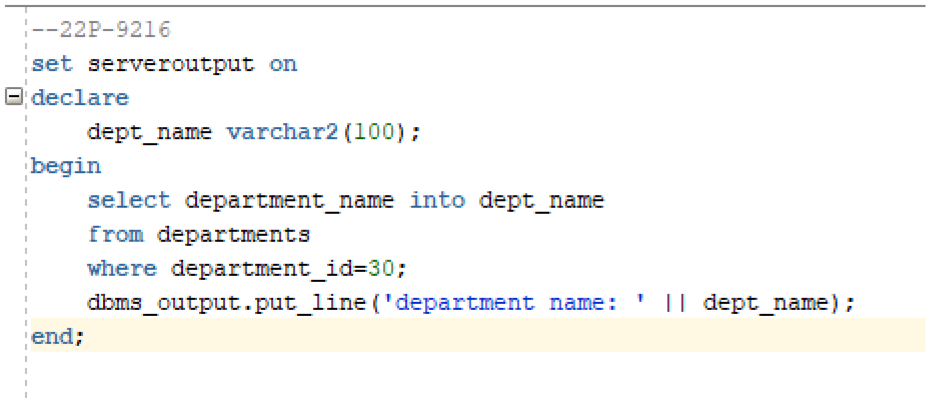


```
commission_pct number;
salary number;
begin
select salary, commission_pct into salary, commission_pct from employees
where employee_id=emp_id;
if commission_pct is null then
commission:=0;
else
commission:=salary*commission_pct;
end if;
update employees
set salary=salary+commission
where employee_id=emp_id;
dbms_output.put_line('new salary: ' || (salary + commission));
commit;
end;
new salary: 24000


PL/SQL procedure successfully completed.
```

Question 3:

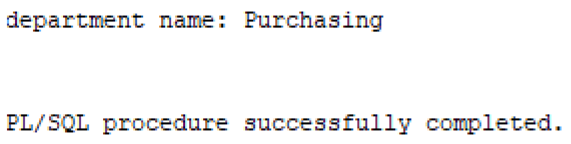
Write a PL/SQL block to obtain the department name of the employee who works for deptno 30.

Answer:



```
--22P-9216
set serveroutput on
declare
    dept_name varchar2(100);
begin
    select department_name into dept_name
    from departments
    where department_id=30;
    dbms_output.put_line('department name: ' || dept_name);
end;
```

Output:



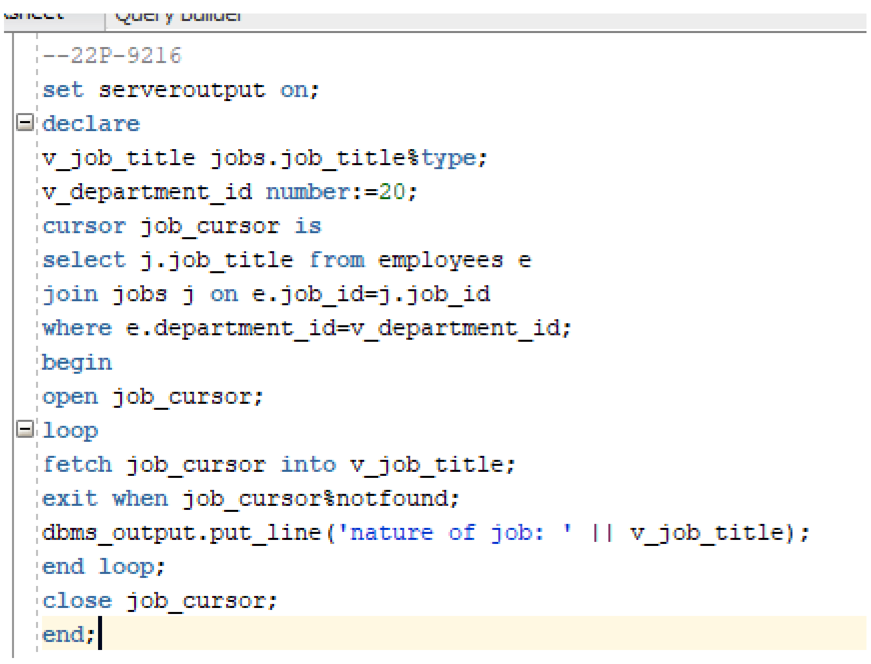
```
department name: Purchasing


PL/SQL procedure successfully completed.
```

____

Question 4:

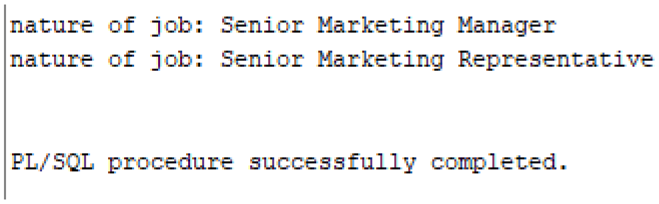
Write a PL /SQL block to find the nature of job of the employee whose deptno is 20(to be passed as an argument).

Answer:



```
--22P-9216
set serveroutput on;
declare
v_job_title jobs.job_title%type;
v_department_id number:=20;
cursor job_cursor is
select j.job_title from employees e
join jobs j on e.job_id=j.job_id
where e.department_id=v_department_id;
begin
open job_cursor;
loop
fetch job_cursor into v_job_title;
exit when job_cursor%notfound;
dbms_output.put_line('nature of job: ' || v_job_title);
end loop;
close job_cursor;
end;
```

Output:



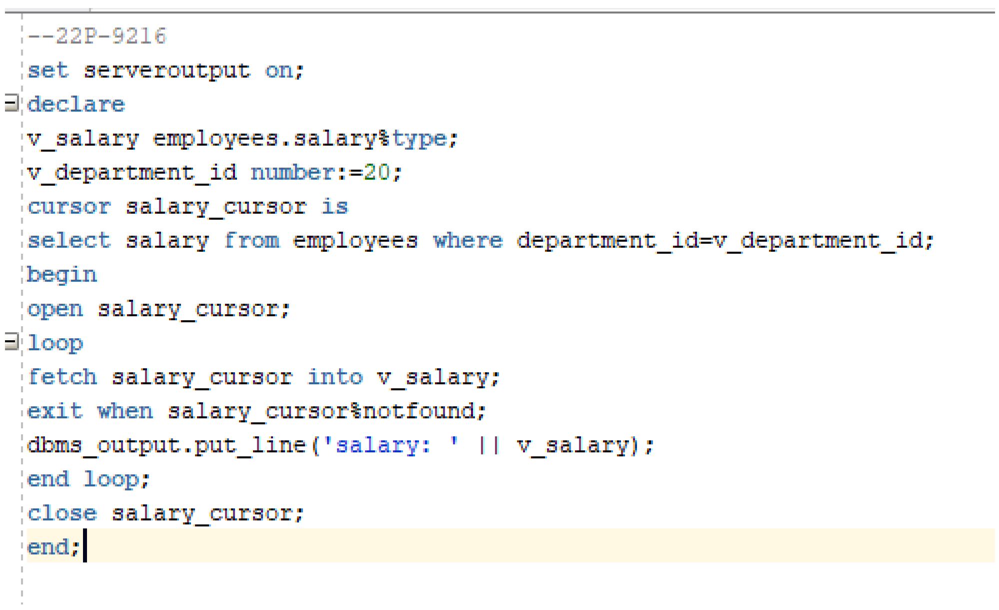
```
nature of job: Senior Marketing Manager
nature of job: Senior Marketing Representative


PL/SQL procedure successfully completed.
```

____

Question 5:

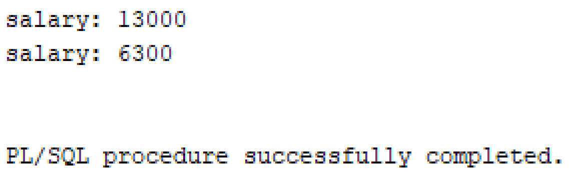
Write a PL /SQL block to find the salary of the employee who is working in the deptno 20(to be passed as an argument).

Answer:



```
--22P-9216
set serveroutput on;
declare
v_salary employees.salary%type;
v_department_id number:=20;
cursor salary_cursor is
select salary from employees where department_id=v_department_id;
begin
open salary_cursor;
loop
fetch salary_cursor into v_salary;
exit when salary_cursor%notfound;
dbms_output.put_line('salary: ' || v_salary);
end loop;
close salary_cursor;
end;
```

Output:



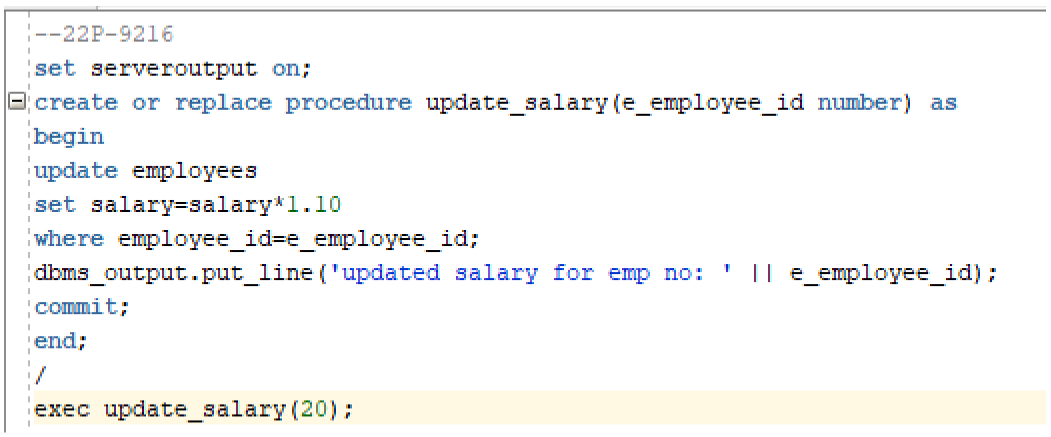
```
salary: 13000
salary: 6300


PL/SQL procedure successfully completed.
```

____

Question 6:

Write a PL/SQL block to update the salary of the employee with a 10% increase whose empno is to be passed as an argument for the procedure.

Answer:



```sql
--22P-9216
set serveroutput on;
create or replace procedure update_salary(e_employee_id number) as
begin
update employees
set salary=salary*1.10
where employee_id=e_employee_id;
dbms_output.put_line('updated salary for emp no: ' || e_employee_id);
commit;
end;
/
exec update_salary(20);
```

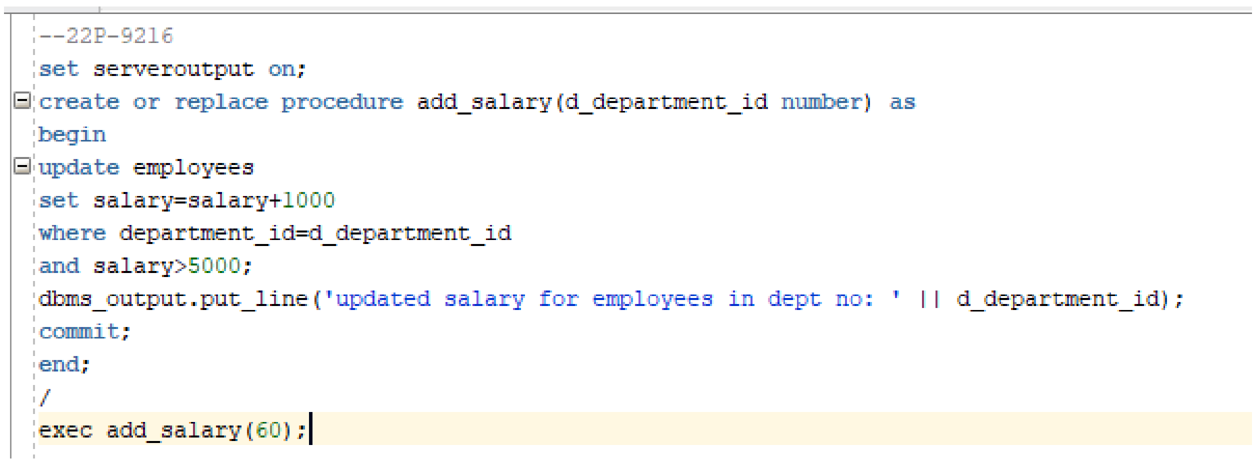
Output:

```
updated salary for emp no: 20


PL/SQL procedure successfully completed.
```

Question 7:

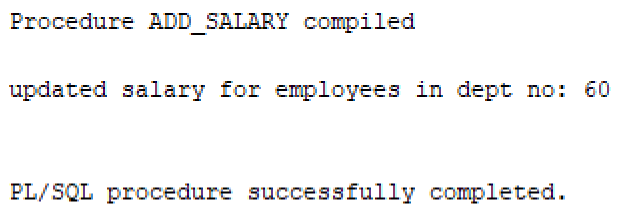
Write a procedure to add an amount of Rs.1000 for the employees whose salaries is greater than 5000 and who belongs to the deptno passed as an argument.

Answer:



```sql
--22P-9216
set serveroutput on;
create or replace procedure add_salary(d_department_id number) as
begin
update employees
set salary=salary+1000
where department_id=d_department_id
and salary>5000;
dbms_output.put_line('updated salary for employees in dept no: ' || d_department_id);
commit;
end;
/
exec add_salary(60);
```

Output:



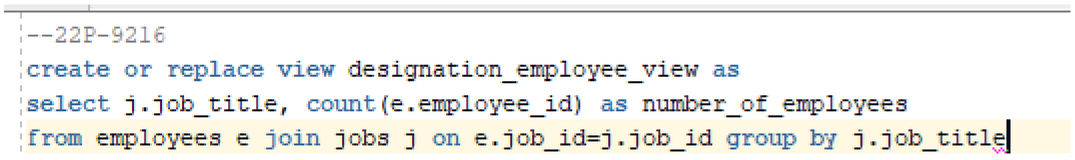
```
Procedure ADD_SALARY compiled

updated salary for employees in dept no: 60


PL/SQL procedure successfully completed.
```

Question 8:

Create views for following purposes: -

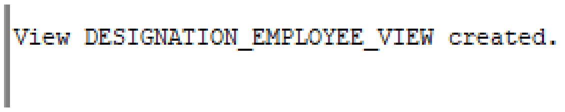
a. Display each designation and number of employees with that particular designation.

Answer:



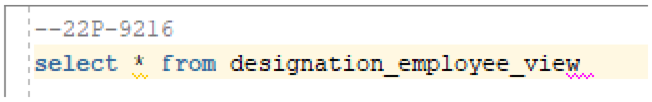
```
--22P-9216
create or replace view designation_employee_view as
select j.job_title, count(e.employee_id) as number_of_employees
from employees e join jobs j on e.job_id=j.job_id group by j.job_title
```

Output:

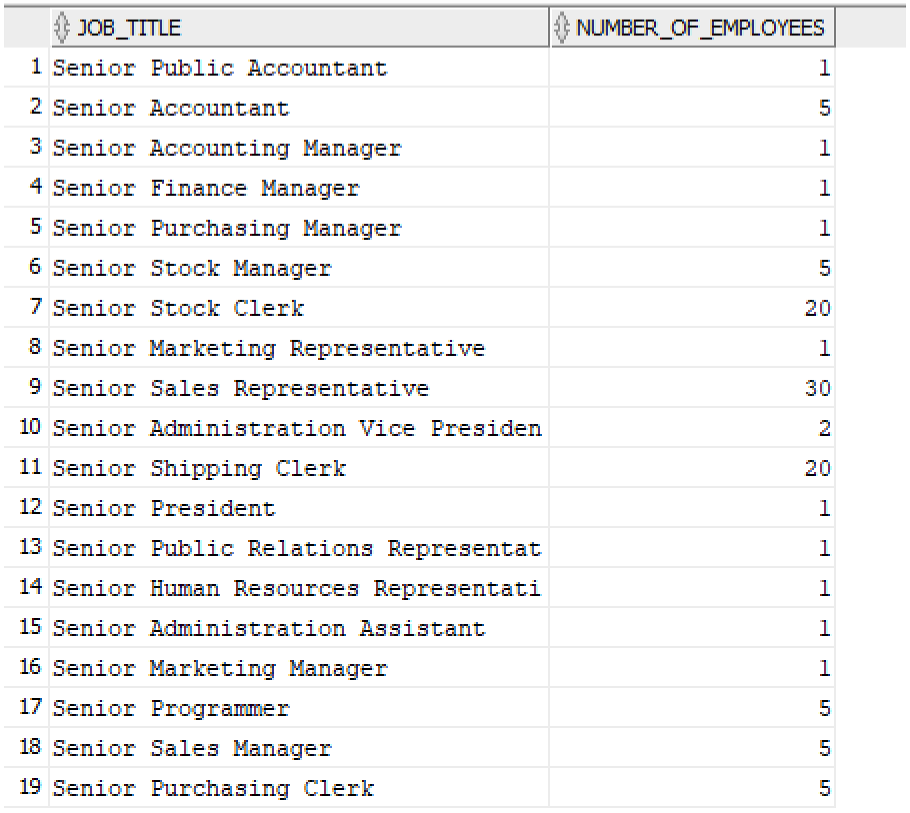


```
View DESIGNATION_EMPLOYEE_VIEW created.
```

To view the views:



```
--22P-9216
select * from designation_employee_view
```



	JOB_TITLE	NUMBER_OF_EMPLOYEES
1	Senior Public Accountant	1
2	Senior Accountant	5
3	Senior Accounting Manager	1
4	Senior Finance Manager	1
5	Senior Purchasing Manager	1
6	Senior Stock Manager	5
7	Senior Stock Clerk	20
8	Senior Marketing Representative	1
9	Senior Sales Representative	30
10	Senior Administration Vice Presiden	2
11	Senior Shipping Clerk	20
12	Senior President	1
13	Senior Public Relations Representat	1
14	Senior Human Resources Representati	1
15	Senior Administration Assistant	1
16	Senior Marketing Manager	1
17	Senior Programmer	5
18	Senior Sales Manager	5
19	Senior Purchasing Clerk	5

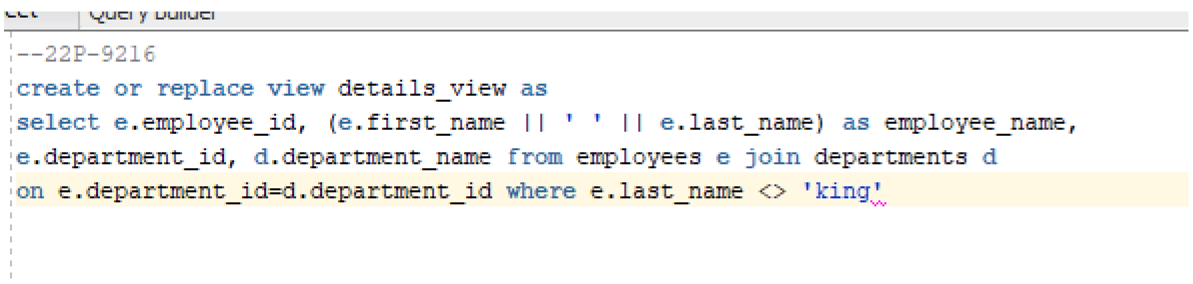
____

Question 8:

Create views for following purposes: -

b. The organization wants to display only the details like empno, empname , deptno , deptname of all the employee except king.

Answer:

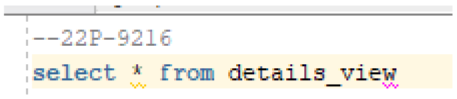


```
--22P-9216
create or replace view details_view as
select e.employee_id, (e.first_name || ' ' || e.last_name) as employee_name,
e.department_id, d.department_name from employees e join departments d
on e.department_id=d.department_id where e.last_name <> 'king'
```

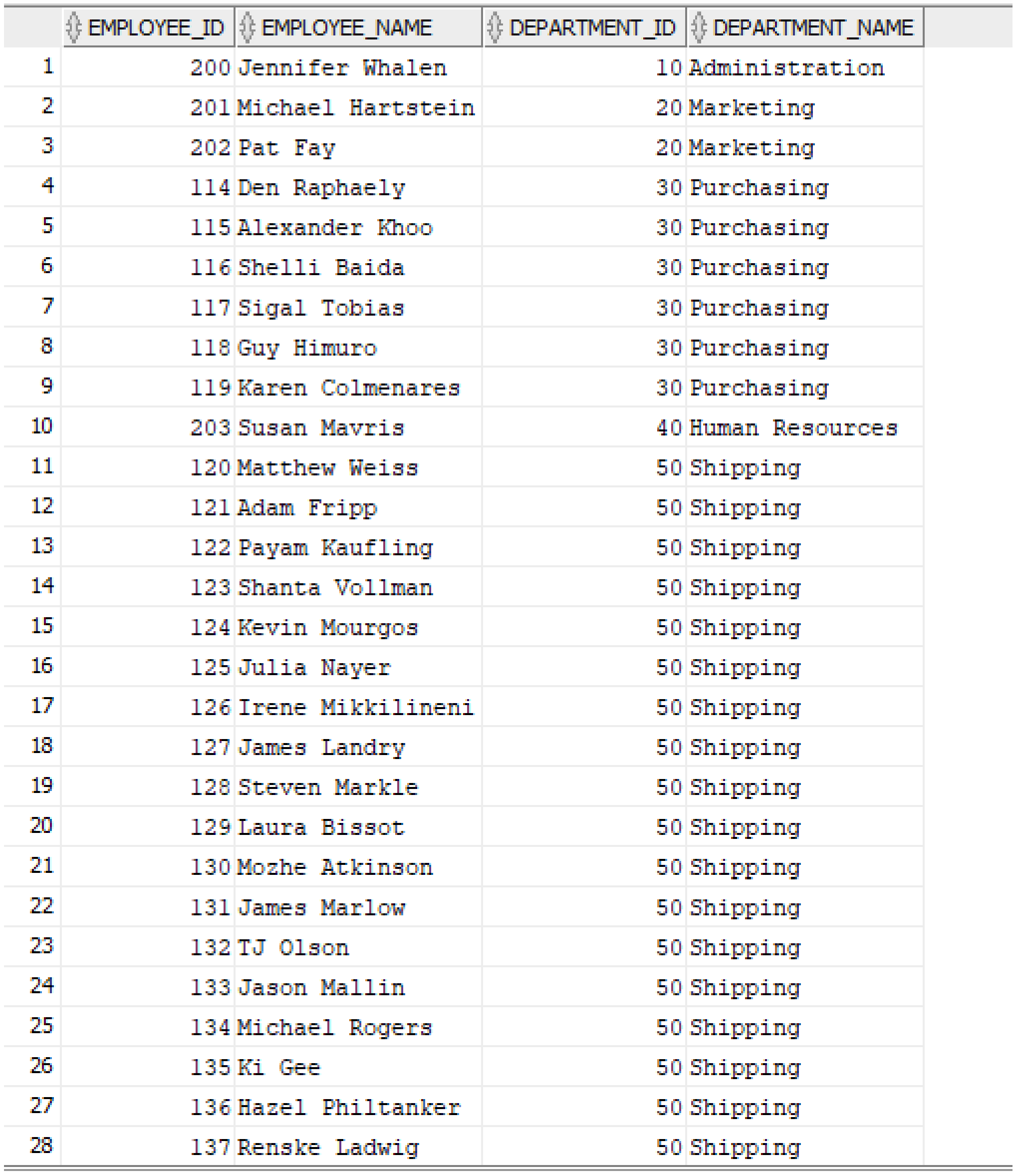
Output:

```
View DETAILS_VIEW created.
```

To view it:



```
--22P-9216
select * from details_view
```



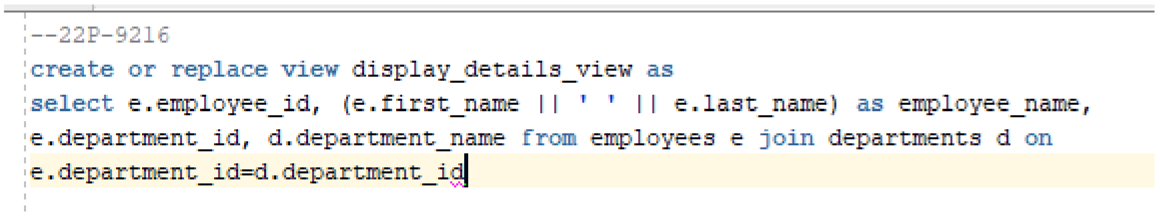
	EMPLOYEE_ID	EMPLOYEE_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	200	Jennifer Whalen	10	Administration
2	201	Michael Hartstein	20	Marketing
3	202	Pat Fay	20	Marketing
4	114	Den Raphaely	30	Purchasing
5	115	Alexander Khoo	30	Purchasing
6	116	Shelli Baida	30	Purchasing
7	117	Sigal Tobias	30	Purchasing
8	118	Guy Himuro	30	Purchasing
9	119	Karen Colmenares	30	Purchasing
10	203	Susan Mavris	40	Human Resources
11	120	Matthew Weiss	50	Shipping
12	121	Adam Fripp	50	Shipping
13	122	Payam Kaufling	50	Shipping
14	123	Shanta Vollman	50	Shipping
15	124	Kevin Mourgos	50	Shipping
16	125	Julia Nayer	50	Shipping
17	126	Irene Mikkilineni	50	Shipping
18	127	James Landry	50	Shipping
19	128	Steven Markle	50	Shipping
20	129	Laura Bissot	50	Shipping
21	130	Mozhe Atkinson	50	Shipping
22	131	James Marlow	50	Shipping
23	132	TJ Olson	50	Shipping
24	133	Jason Mallin	50	Shipping
25	134	Michael Rogers	50	Shipping
26	135	Ki Gee	50	Shipping
27	136	Hazel Philtanker	50	Shipping
28	137	Renske Ladwig	50	Shipping

Question 8:

Create views for following purposes: -

c. The organization wants to display only the details empno, empname, deptno, deptname of the employees.

Answer:

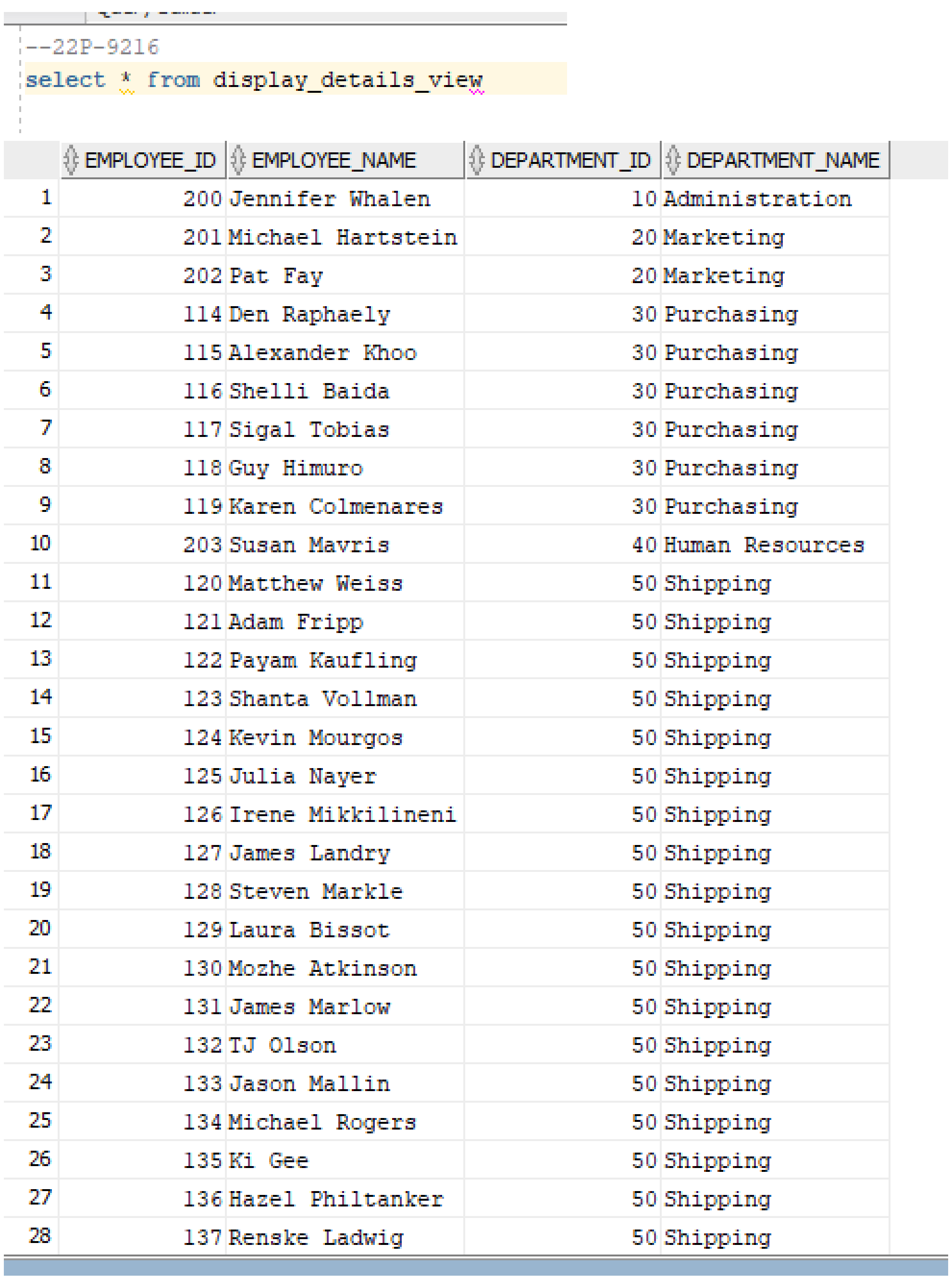


```sql
--22P-9216
create or replace view display_details_view as
select e.employee_id, (e.first_name || ' ' || e.last_name) as employee_name,
e.department_id, d.department_name from employees e join departments d on
e.department_id=d.department_id
```

Output:

```
View DISPLAY_DETAILS_VIEW created.
```

To view it:



```
--22P-9216
select * from display_details_view
```

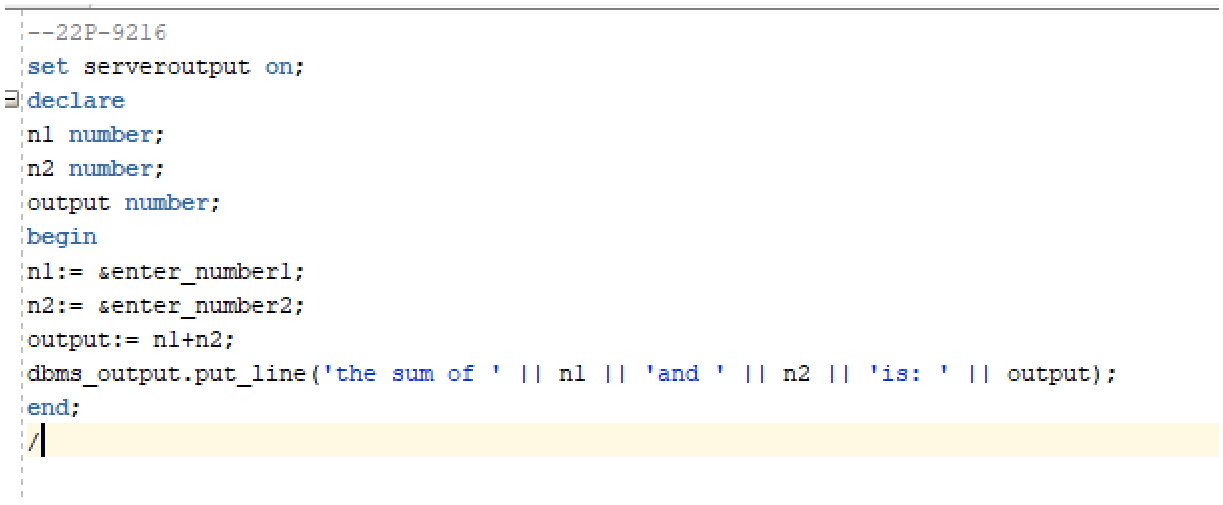
	EMPLOYEE_ID	EMPLOYEE_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	200	Jennifer Whalen	10	Administration
2	201	Michael Hartstein	20	Marketing
3	202	Pat Fay	20	Marketing
4	114	Den Raphaely	30	Purchasing
5	115	Alexander Khoo	30	Purchasing
6	116	Shelli Baida	30	Purchasing
7	117	Sigal Tobias	30	Purchasing
8	118	Guy Himuro	30	Purchasing
9	119	Karen Colmenares	30	Purchasing
10	203	Susan Mavris	40	Human Resources
11	120	Matthew Weiss	50	Shipping
12	121	Adam Fripp	50	Shipping
13	122	Payam Kaufling	50	Shipping
14	123	Shanta Vollman	50	Shipping
15	124	Kevin Mourgos	50	Shipping
16	125	Julia Nayer	50	Shipping
17	126	Irene Mikkilineni	50	Shipping
18	127	James Landry	50	Shipping
19	128	Steven Markle	50	Shipping
20	129	Laura Bissot	50	Shipping
21	130	Mozhe Atkinson	50	Shipping
22	131	James Marlow	50	Shipping
23	132	TJ Olson	50	Shipping
24	133	Jason Mallin	50	Shipping
25	134	Michael Rogers	50	Shipping
26	135	Ki Gee	50	Shipping
27	136	Hazel Philtanker	50	Shipping
28	137	Renske Ladwig	50	Shipping

____

Question 9:

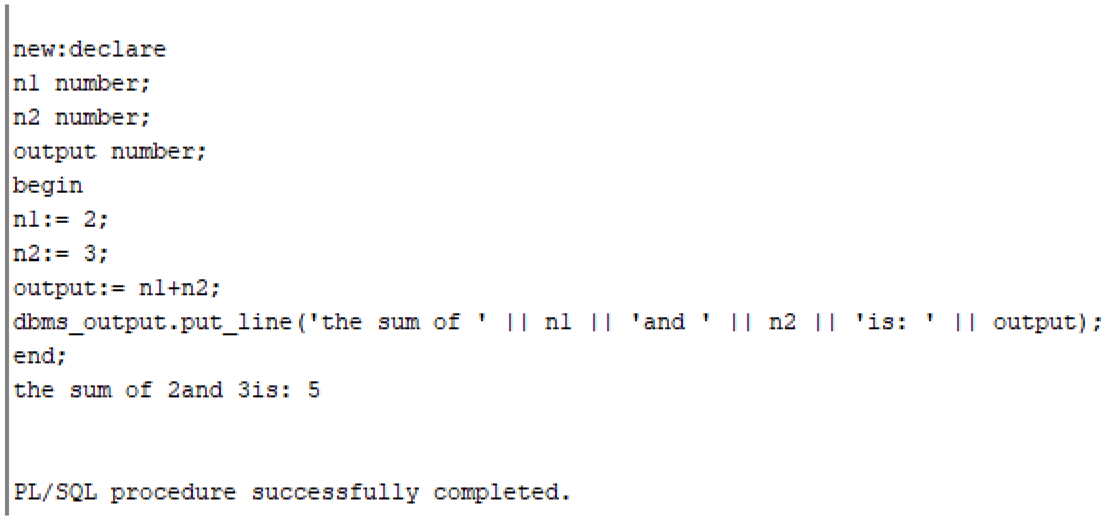
Write a PL/SQL code that takes two inputs from user, add them and store the sum in new variable and show the output.

Answer:



```
--22P-9216
set serveroutput on;
declare
n1 number;
n2 number;
output number;
begin
n1:= &enter_number1;
n2:= &enter_number2;
output:= n1+n2;
dbms_output.put_line('the sum of ' || n1 || 'and ' || n2 || 'is: ' || output);
end;
/
```

Output:

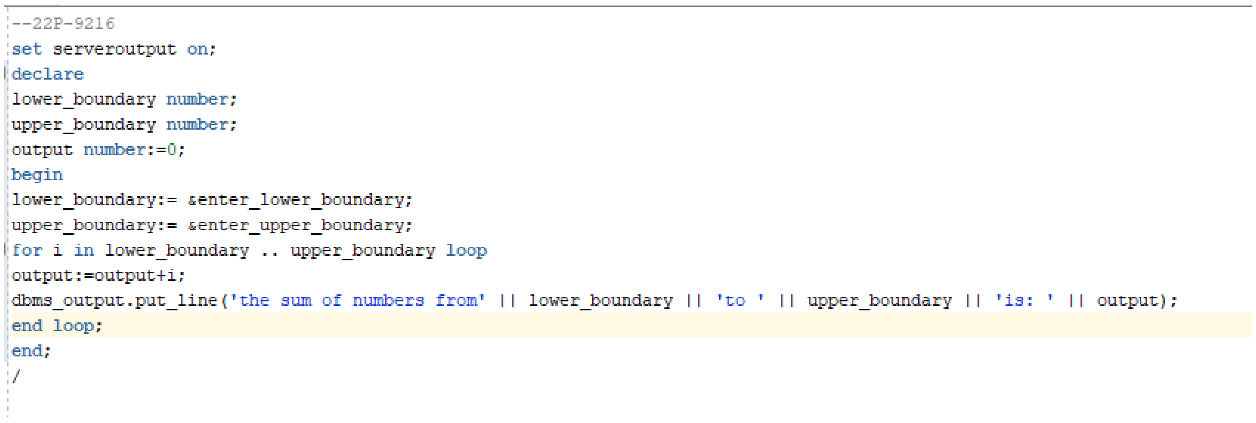


```
new:declare
n1 number;
n2 number;
output number;
begin
n1:= 2;
n2:= 3;
output:= n1+n2;
dbms_output.put_line('the sum of ' || n1 || 'and ' || n2 || 'is: ' || output);
end;
the sum of 2and 3is: 5


PL/SQL procedure successfully completed.
```

Question 10:

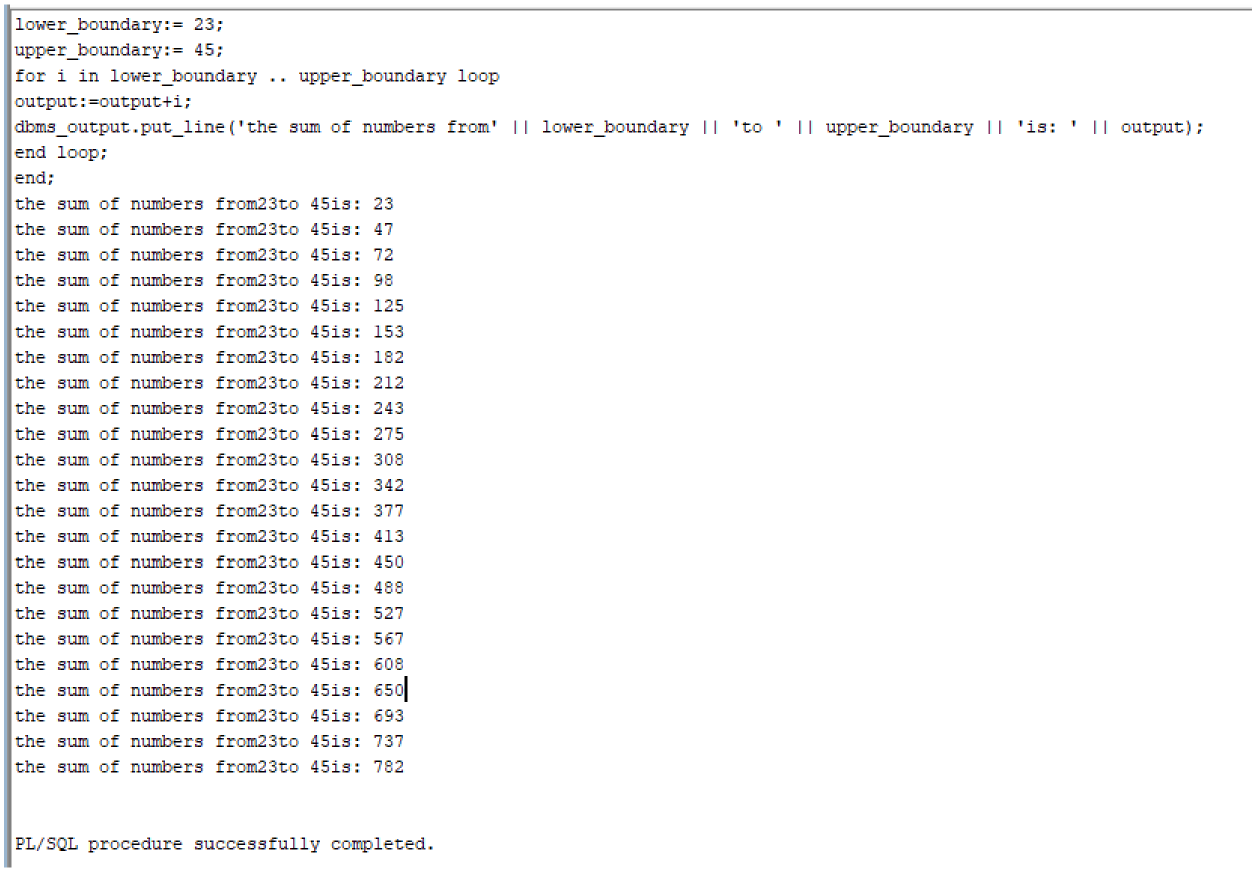
Write a PL/SQL code that takes two inputs, lower boundary and upper boundary, then print the sum of all the numbers between the boundaries INCLUSIVE.

Answer:



```
--22P-9216
set serveroutput on;
declare
lower_boundary number;
upper_boundary number;
output number:=0;
begin
lower_boundary:= &enter_lower_boundary;
upper_boundary:= &enter_upper_boundary;
for i in lower_boundary .. upper_boundary loop
output:=output+i;
dbms_output.put_line('the sum of numbers from' || lower_boundary || 'to ' || upper_boundary || 'is: ' || output);
end loop;
end;
/
```

Output:

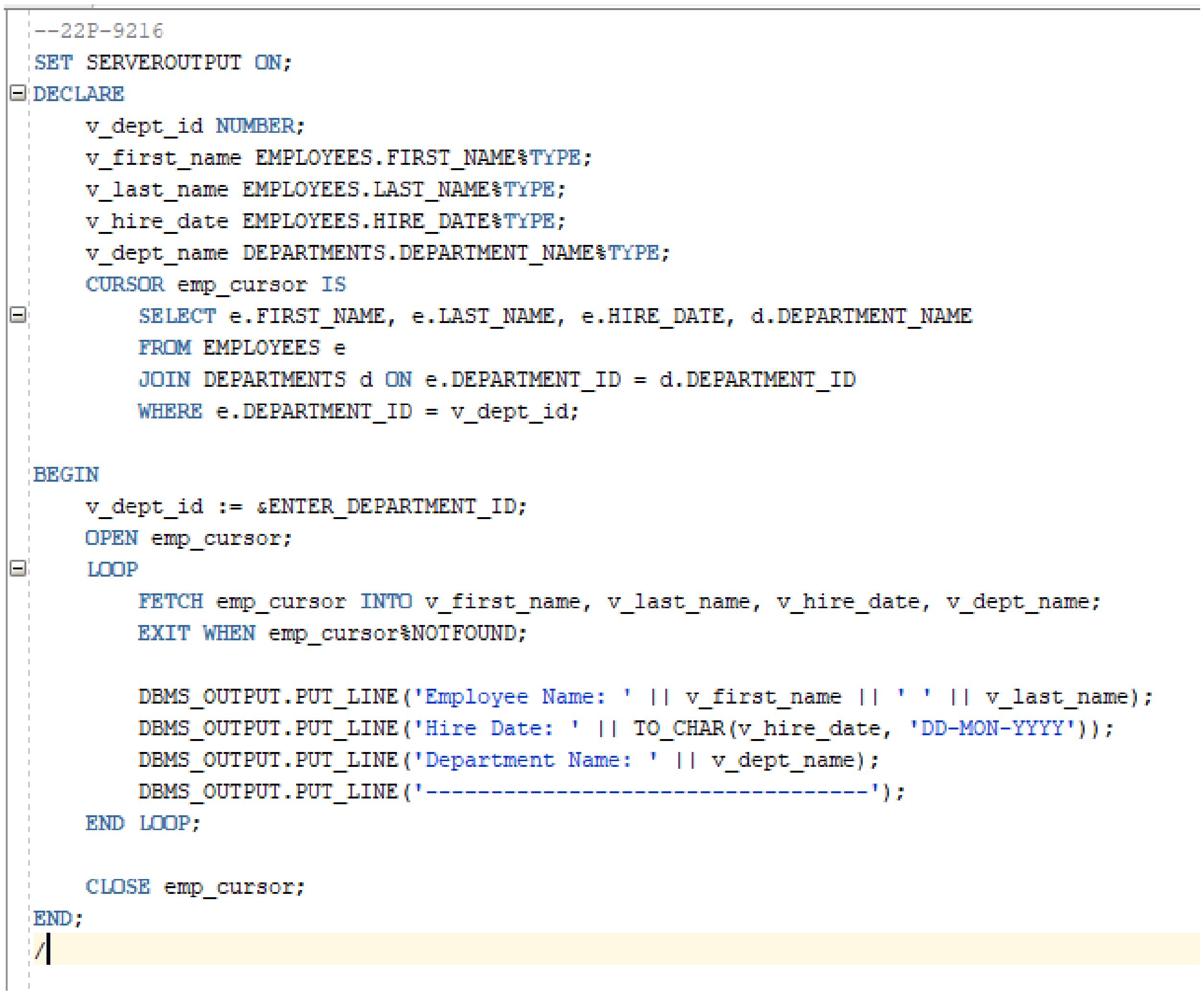


```
lower_boundary:= 23;
upper_boundary:= 45;
for i in lower_boundary .. upper_boundary loop
output:=output+i;
dbms_output.put_line('the sum of numbers from' || lower_boundary || 'to ' || upper_boundary || 'is: ' || output);
end loop;
end;
the sum of numbers from23to 45is: 23
the sum of numbers from23to 45is: 47
the sum of numbers from23to 45is: 72
the sum of numbers from23to 45is: 98
the sum of numbers from23to 45is: 125
the sum of numbers from23to 45is: 153
the sum of numbers from23to 45is: 182
the sum of numbers from23to 45is: 212
the sum of numbers from23to 45is: 243
the sum of numbers from23to 45is: 275
the sum of numbers from23to 45is: 308
the sum of numbers from23to 45is: 342
the sum of numbers from23to 45is: 377
the sum of numbers from23to 45is: 413
the sum of numbers from23to 45is: 450
the sum of numbers from23to 45is: 488
the sum of numbers from23to 45is: 527
the sum of numbers from23to 45is: 567
the sum of numbers from23to 45is: 608
the sum of numbers from23to 45is: 650
the sum of numbers from23to 45is: 693
the sum of numbers from23to 45is: 737
the sum of numbers from23to 45is: 782


PL/SQL procedure successfully completed.
```

Question 11:

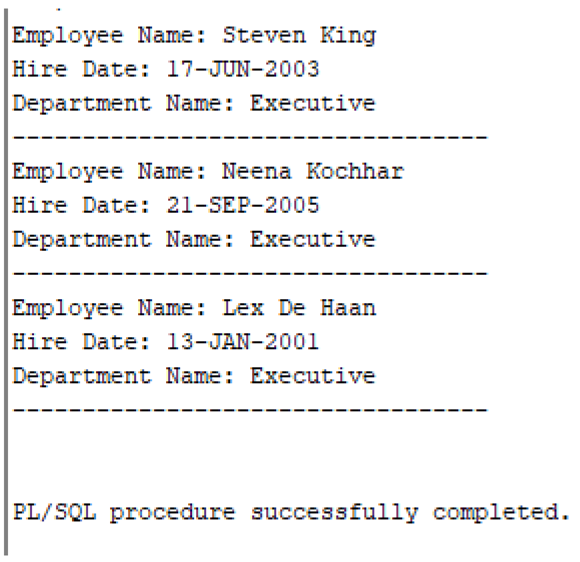
Write a PL/SQL code to retrieve the employee name, hiredate, and the department name in which he works, whose number is input by the user.

Answer:



```
--22P-9216
SET SERVEROUTPUT ON;
DECLARE
    v_dept_id NUMBER;
    v_first_name EMPLOYEES.FIRST_NAME%TYPE;
    v_last_name EMPLOYEES.LAST_NAME%TYPE;
    v_hire_date EMPLOYEES.HIRE_DATE%TYPE;
    v_dept_name DEPARTMENTS.DEPARTMENT_NAME%TYPE;
    CURSOR emp_cursor IS
        SELECT e.FIRST_NAME, e.LAST_NAME, e.HIRE_DATE, d.DEPARTMENT_NAME
        FROM EMPLOYEES e
        JOIN DEPARTMENTS d ON e.DEPARTMENT_ID = d.DEPARTMENT_ID
        WHERE e.DEPARTMENT_ID = v_dept_id;

BEGIN
    v_dept_id := &ENTER_DEPARTMENT_ID;
    OPEN emp_cursor;
    LOOP
        FETCH emp_cursor INTO v_first_name, v_last_name, v_hire_date, v_dept_name;
        EXIT WHEN emp_cursor%NOTFOUND;

        DBMS_OUTPUT.PUT_LINE('Employee Name: ' || v_first_name || ' ' || v_last_name);
        DBMS_OUTPUT.PUT_LINE('Hire Date: ' || TO_CHAR(v_hire_date, 'DD-MON-YYYY'));
        DBMS_OUTPUT.PUT_LINE('Department Name: ' || v_dept_name);
        DBMS_OUTPUT.PUT_LINE('----------------------------------');
    END LOOP;

    CLOSE emp_cursor;
END;
/
```

Output:

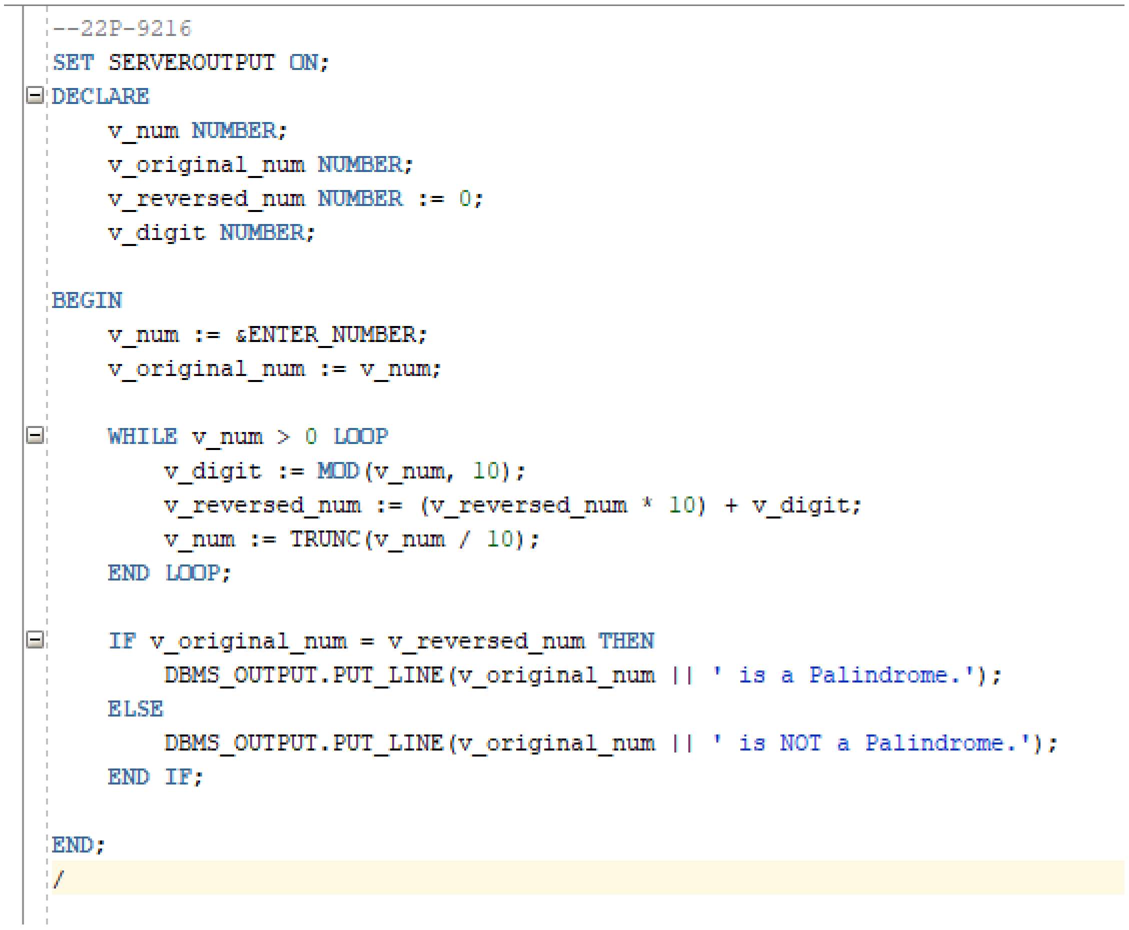


```
Employee Name: Steven King
Hire Date: 17-JUN-2003
Department Name: Executive
----------------------------------
Employee Name: Neena Kochhar
Hire Date: 21-SEP-2005
Department Name: Executive
----------------------------------
Employee Name: Lex De Haan
Hire Date: 13-JAN-2001
Department Name: Executive
----------------------------------


PL/SQL procedure successfully completed.
```

Question 12:

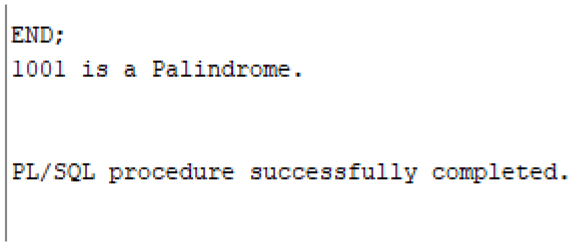
Write a PL/SQL code to check whether the given number is palindrome or not.

Answer:



```
--22P-9216
SET SERVEROUTPUT ON;
DECLARE
    v_num NUMBER;
    v_original_num NUMBER;
    v_reversed_num NUMBER := 0;
    v_digit NUMBER;

BEGIN
    v_num := &ENTER_NUMBER;
    v_original_num := v_num;

    WHILE v_num > 0 LOOP
        v_digit := MOD(v_num, 10);
        v_reversed_num := (v_reversed_num * 10) + v_digit;
        v_num := TRUNC(v_num / 10);
    END LOOP;

    IF v_original_num = v_reversed_num THEN
        DBMS_OUTPUT.PUT_LINE(v_original_num || ' is a Palindrome.');
    ELSE
        DBMS_OUTPUT.PUT_LINE(v_original_num || ' is NOT a Palindrome.');
    END IF;

END;
/
```

Output:

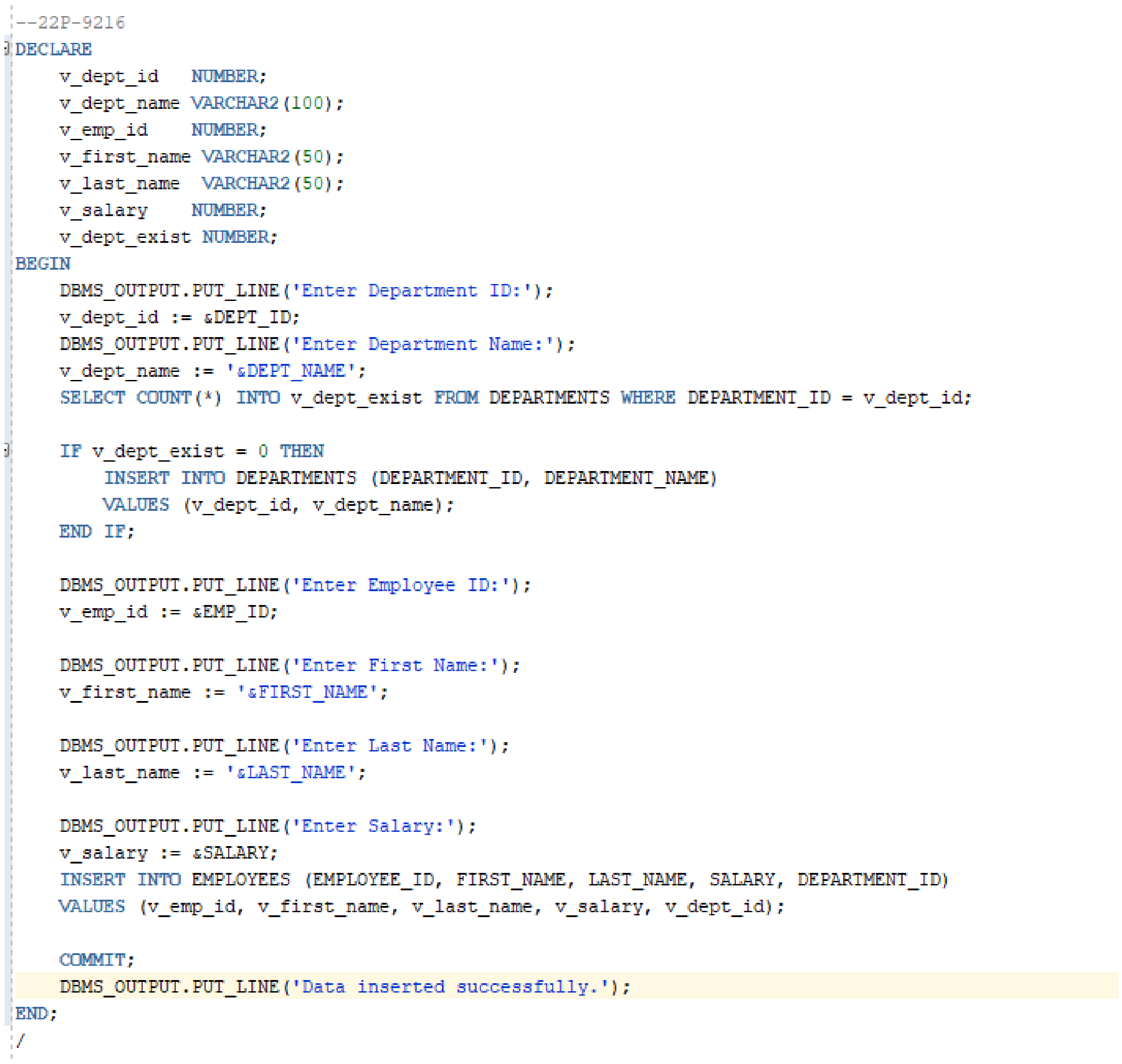


```
END;
1001 is a Palindrome.


PL/SQL procedure successfully completed.
```

Question 13:

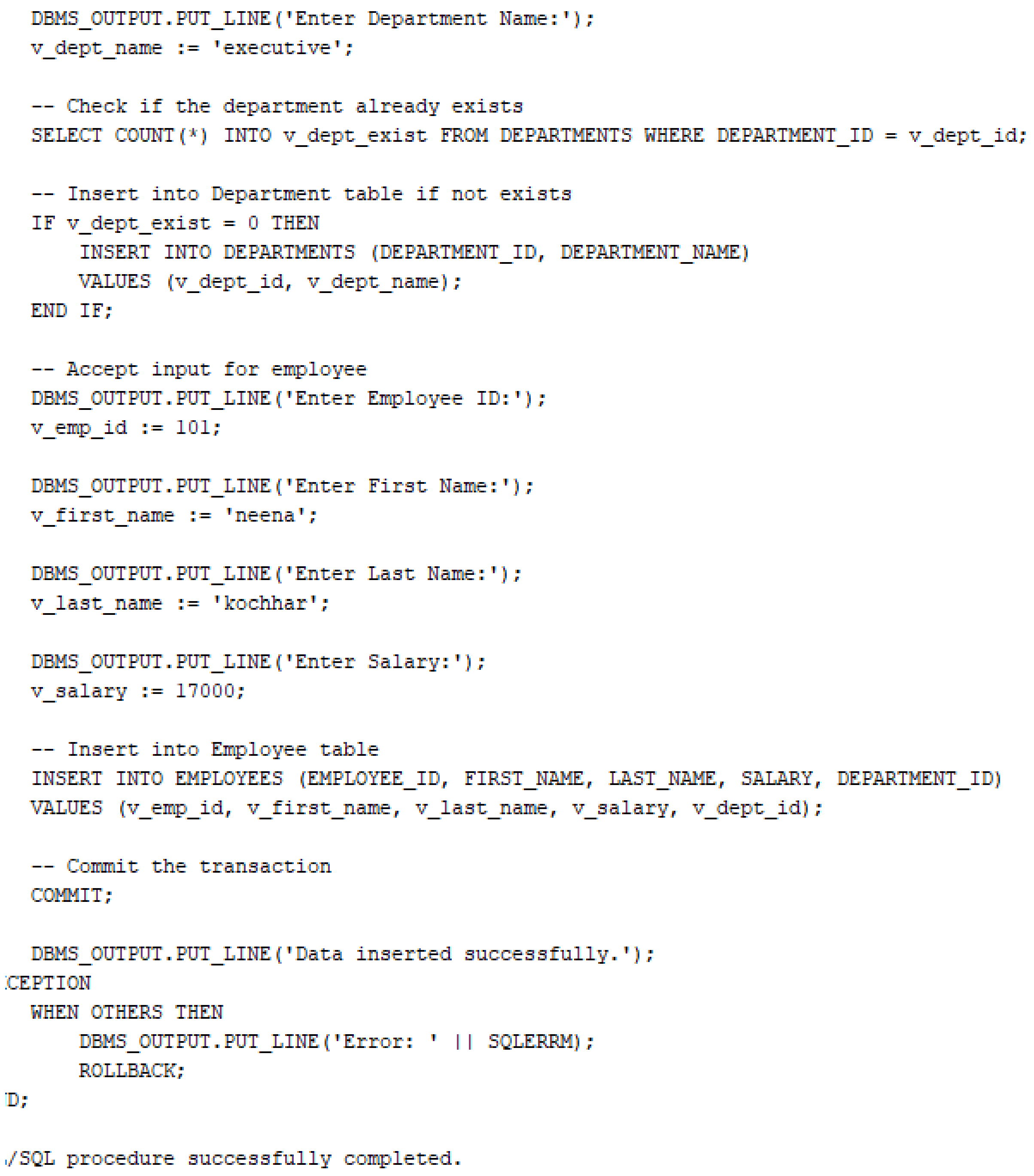
Write a PL/SQL code that takes all the required inputs from the user for the Employee table and then insert it into the Employee and Department table in the database.

Answer:



```
--22P-9216
DECLARE
    v_dept_id   NUMBER;
    v_dept_name VARCHAR2(100);
    v_emp_id    NUMBER;
    v_first_name VARCHAR2(50);
    v_last_name  VARCHAR2(50);
    v_salary    NUMBER;
    v_dept_exist NUMBER;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Enter Department ID:');
    v_dept_id := &DEPT_ID;
    DBMS_OUTPUT.PUT_LINE('Enter Department Name:');
    v_dept_name := '&DEPT_NAME';
    SELECT COUNT(*) INTO v_dept_exist FROM DEPARTMENTS WHERE DEPARTMENT_ID = v_dept_id;

    IF v_dept_exist = 0 THEN
        INSERT INTO DEPARTMENTS (DEPARTMENT_ID, DEPARTMENT_NAME)
        VALUES (v_dept_id, v_dept_name);
    END IF;

    DBMS_OUTPUT.PUT_LINE('Enter Employee ID:');
    v_emp_id := &EMP_ID;

    DBMS_OUTPUT.PUT_LINE('Enter First Name:');
    v_first_name := '&FIRST_NAME';

    DBMS_OUTPUT.PUT_LINE('Enter Last Name:');
    v_last_name := '&LAST_NAME';

    DBMS_OUTPUT.PUT_LINE('Enter Salary:');
    v_salary := &SALARY;
    INSERT INTO EMPLOYEES (EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY, DEPARTMENT_ID)
    VALUES (v_emp_id, v_first_name, v_last_name, v_salary, v_dept_id);

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Data inserted successfully.');
END;
/
```

Output:

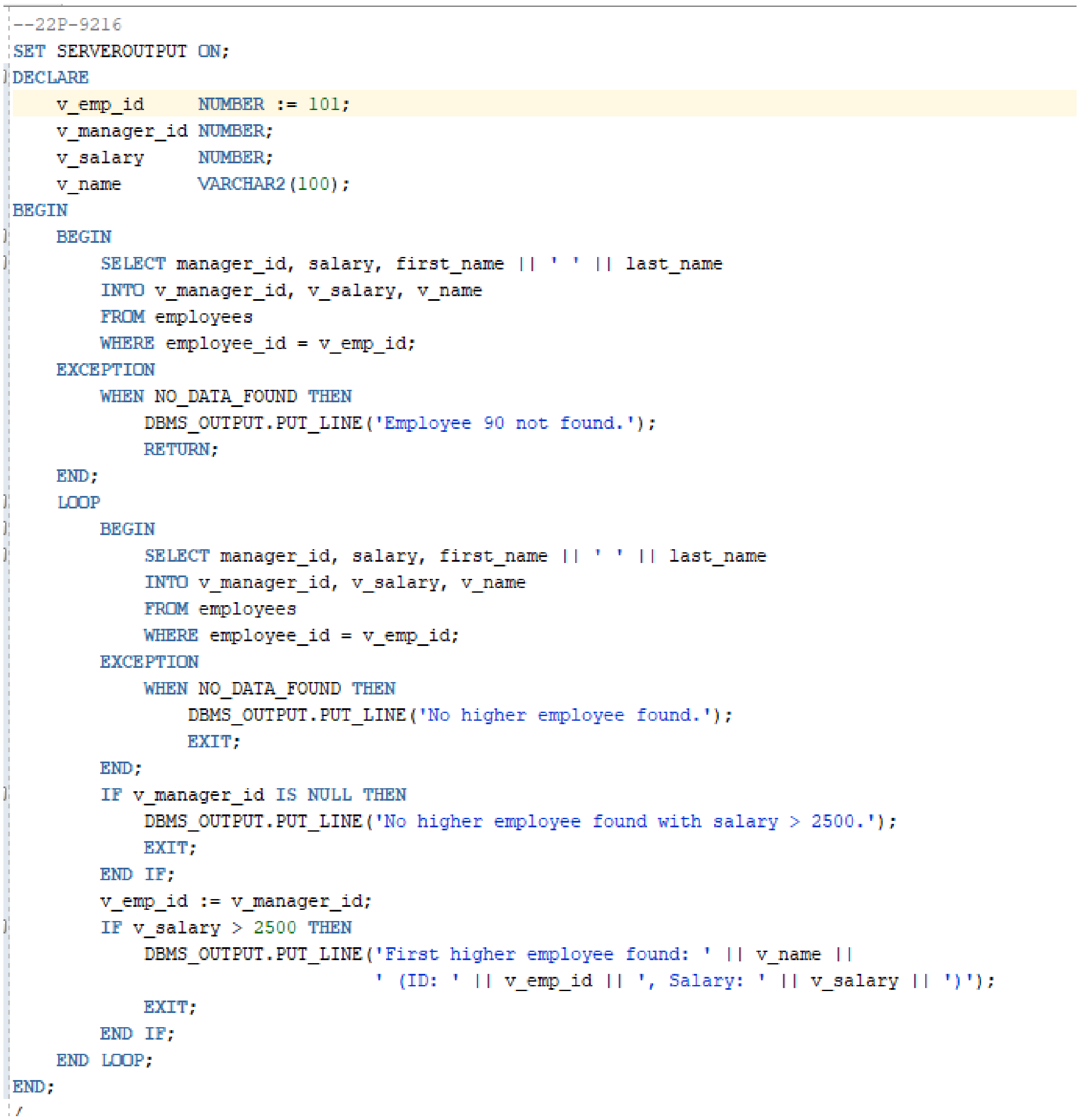


```
  DBMS_OUTPUT.PUT_LINE('Enter Department Name:');
  v_dept_name := 'executive';

  -- Check if the department already exists
  SELECT COUNT(*) INTO v_dept_exist FROM DEPARTMENTS WHERE DEPARTMENT_ID = v_dept_id;

  -- Insert into Department table if not exists
  IF v_dept_exist = 0 THEN
      INSERT INTO DEPARTMENTS (DEPARTMENT_ID, DEPARTMENT_NAME)
      VALUES (v_dept_id, v_dept_name);
  END IF;

  -- Accept input for employee
  DBMS_OUTPUT.PUT_LINE('Enter Employee ID:');
  v_emp_id := 101;

  DBMS_OUTPUT.PUT_LINE('Enter First Name:');
  v_first_name := 'neena';

  DBMS_OUTPUT.PUT_LINE('Enter Last Name:');
  v_last_name := 'kochhar';

  DBMS_OUTPUT.PUT_LINE('Enter Salary:');
  v_salary := 17000;

  -- Insert into Employee table
  INSERT INTO EMPLOYEES (EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY, DEPARTMENT_ID)
  VALUES (v_emp_id, v_first_name, v_last_name, v_salary, v_dept_id);

  -- Commit the transaction
  COMMIT;

  DBMS_OUTPUT.PUT_LINE('Data inserted successfully.');
CEPTION
  WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
      ROLLBACK;
D;

/SQL procedure successfully completed.
```

Question 14:

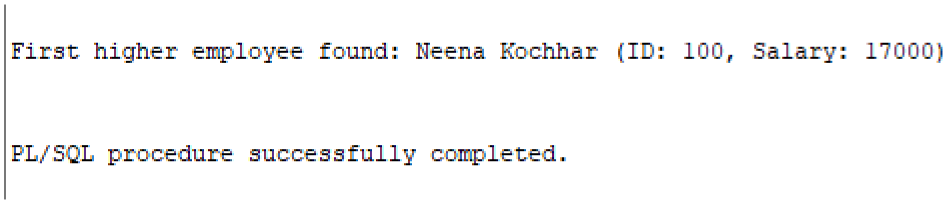
Write a PL/SQL code to find the first employee who has a salary over $2500 and is higher in the chain of command than employee 90. Note: For chain, use of LOOP is necessary.

Answer:



```
--22P-9216
SET SERVEROUTPUT ON;
DECLARE
    v_emp_id     NUMBER := 101;
    v_manager_id NUMBER;
    v_salary     NUMBER;
    v_name       VARCHAR2(100);
BEGIN
    BEGIN
        SELECT manager_id, salary, first_name || ' ' || last_name
        INTO v_manager_id, v_salary, v_name
        FROM employees
        WHERE employee_id = v_emp_id;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Employee 90 not found.');
            RETURN;
    END;
    LOOP
        BEGIN
            SELECT manager_id, salary, first_name || ' ' || last_name
            INTO v_manager_id, v_salary, v_name
            FROM employees
            WHERE employee_id = v_emp_id;
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                DBMS_OUTPUT.PUT_LINE('No higher employee found.');
                EXIT;
        END;
        IF v_manager_id IS NULL THEN
            DBMS_OUTPUT.PUT_LINE('No higher employee found with salary > 2500.');
            EXIT;
        END IF;
        v_emp_id := v_manager_id;
        IF v_salary > 2500 THEN
            DBMS_OUTPUT.PUT_LINE('First higher employee found: ' || v_name ||
                                 ' (ID: ' || v_emp_id || ', Salary: ' || v_salary || ')');
            EXIT;
        END IF;
    END LOOP;
END;
/
```

Output:

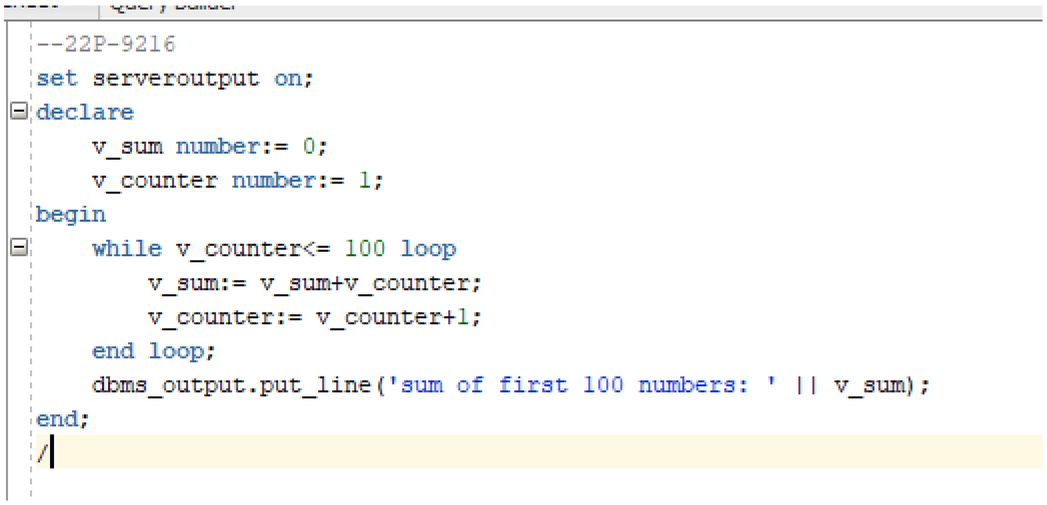


```
First higher employee found: Neena Kochhar (ID: 100, Salary: 17000)


PL/SQL procedure successfully completed.
```

Question 15:

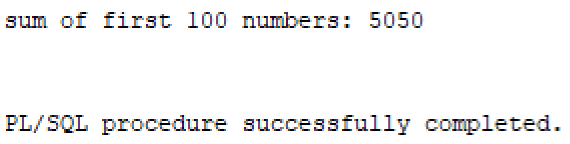
Write a PL/SQL code to print the sum of first 100 numbers.

Answer:



```
--22P-9216
set serveroutput on;
declare
    v_sum number:= 0;
    v_counter number:= 1;
begin
    while v_counter<= 100 loop
        v_sum:= v_sum+v_counter;
        v_counter:= v_counter+1;
    end loop;
    dbms_output.put_line('sum of first 100 numbers: ' || v_sum);
end;
/
```

Output:



```
sum of first 100 numbers: 5050


PL/SQL procedure successfully completed.
```
