PL/SQL Advanced(2)

Aya Sabry

Submitted to:

Eng. Yehia Momtaz

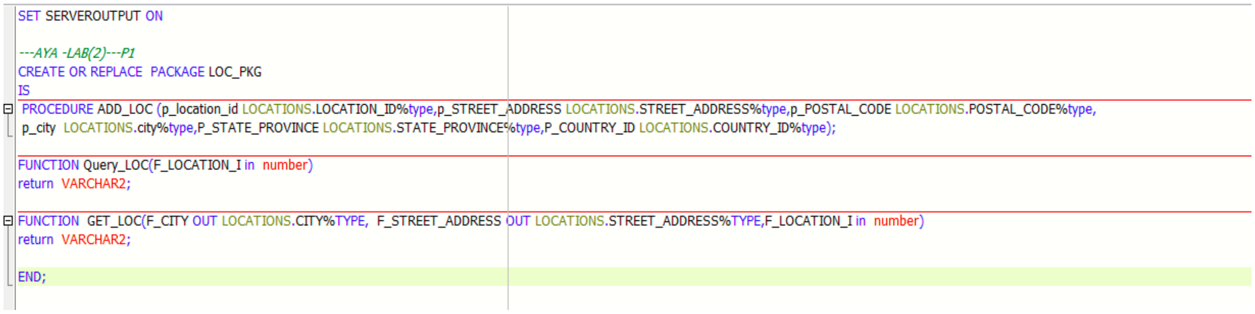
Dec 30, 2022

Task 2

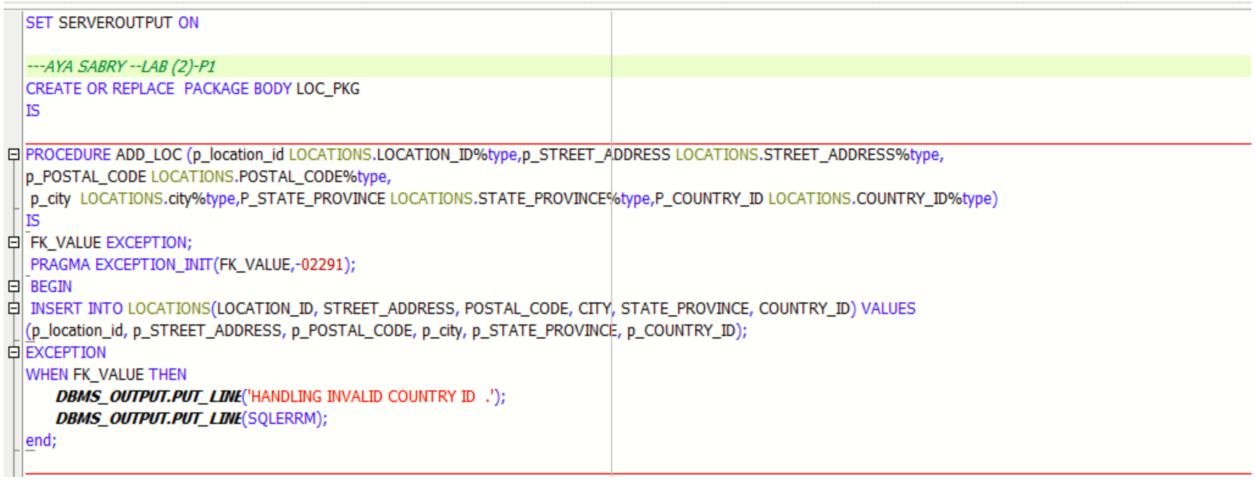
(packages)

1-

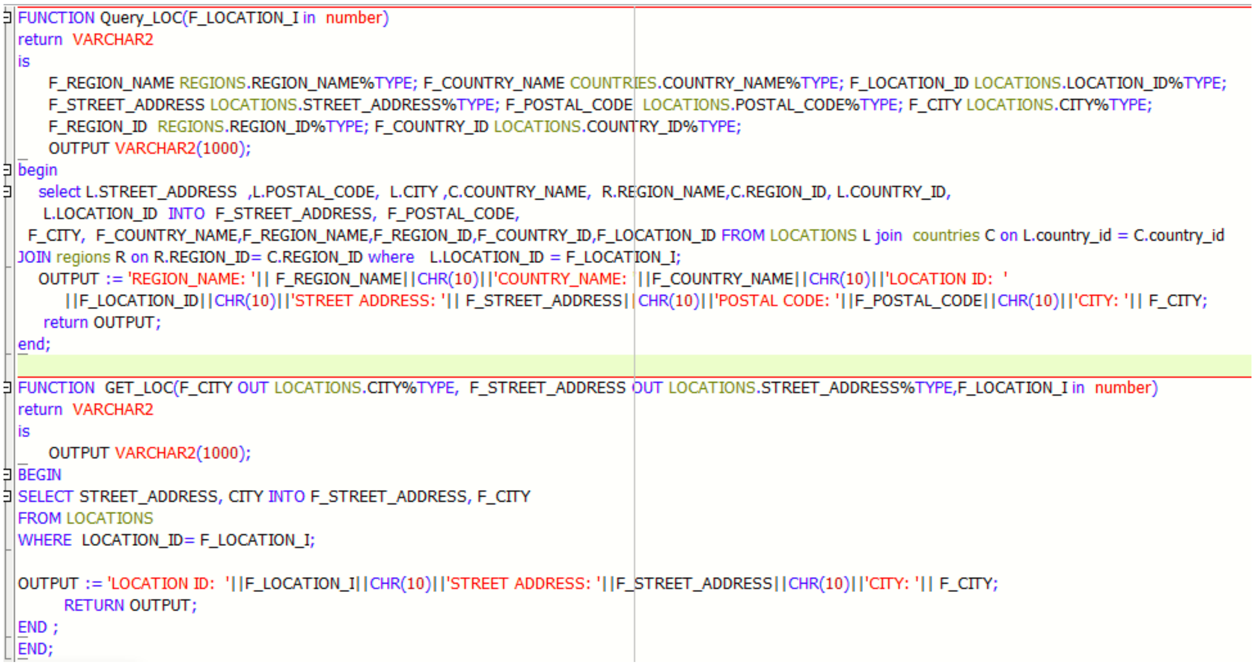
Create a package specification and body called LOC_PKG, containing a copy of your ADD_LOC procedure and Query_LOC function as well as your GET_LOC function.



```
SET SERVEROUTPUT ON

---AYA -LAB(2)---P1
CREATE OR REPLACE  PACKAGE LOC_PKG
IS
 PROCEDURE ADD_LOC (p_location_id LOCATIONS.LOCATION_ID%type,p_STREET_ADDRESS LOCATIONS.STREET_ADDRESS%type,p_POSTAL_CODE LOCATIONS.POSTAL_CODE%type,
 p_city  LOCATIONS.city%type,P_STATE_PROVINCE LOCATIONS.STATE_PROVINCE%type,P_COUNTRY_ID LOCATIONS.COUNTRY_ID%type);

FUNCTION Query_LOC(F_LOCATION_I in  number)
return  VARCHAR2;

FUNCTION  GET_LOC(F_CITY OUT LOCATIONS.CITY%TYPE,  F_STREET_ADDRESS OUT LOCATIONS.STREET_ADDRESS%TYPE,F_LOCATION_I in  number)
return  VARCHAR2;

END;
```



```
SET SERVEROUTPUT ON

---AYA SABRY --LAB (2)-P1
CREATE OR REPLACE  PACKAGE BODY LOC_PKG
IS

PROCEDURE ADD_LOC (p_location_id LOCATIONS.LOCATION_ID%type,p_STREET_ADDRESS LOCATIONS.STREET_ADDRESS%type,
p_POSTAL_CODE LOCATIONS.POSTAL_CODE%type,
 p_city  LOCATIONS.city%type,P_STATE_PROVINCE LOCATIONS.STATE_PROVINCE%type,P_COUNTRY_ID LOCATIONS.COUNTRY_ID%type)
IS
 FK_VALUE EXCEPTION;
 PRAGMA EXCEPTION_INIT(FK_VALUE,-02291);
 BEGIN
 INSERT INTO LOCATIONS(LOCATION_ID, STREET_ADDRESS, POSTAL_CODE, CITY, STATE_PROVINCE, COUNTRY_ID) VALUES
(p_location_id, p_STREET_ADDRESS, p_POSTAL_CODE, p_city, p_STATE_PROVINCE, p_COUNTRY_ID);
EXCEPTION
WHEN FK_VALUE THEN
      DBMS_OUTPUT.PUT_LINE('HANDLING INVALID COUNTRY ID  .');
      DBMS_OUTPUT.PUT_LINE(SQLERRM);
end;
```



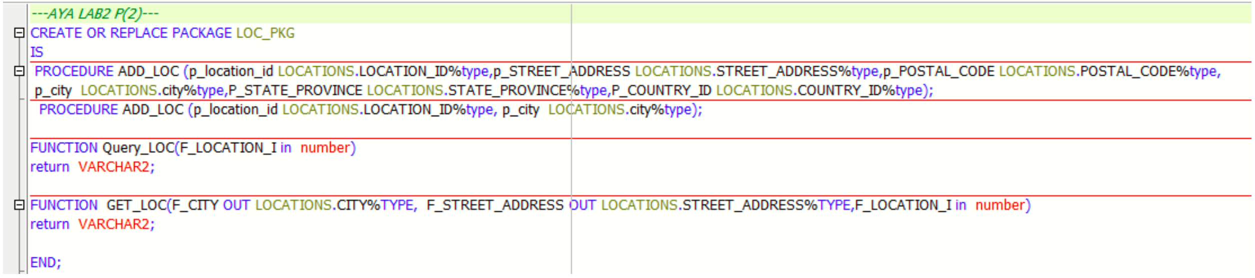
```
FUNCTION Query_LOC(F_LOCATION_I in  number)
return  VARCHAR2
is
     F_REGION_NAME REGIONS.REGION_NAME%TYPE; F_COUNTRY_NAME COUNTRIES.COUNTRY_NAME%TYPE; F_LOCATION_ID LOCATIONS.LOCATION_ID%TYPE;
     F_STREET_ADDRESS LOCATIONS.STREET_ADDRESS%TYPE; F_POSTAL_CODE  LOCATIONS.POSTAL_CODE%TYPE; F_CITY LOCATIONS.CITY%TYPE;
     F_REGION_ID  REGIONS.REGION_ID%TYPE; F_COUNTRY_ID LOCATIONS.COUNTRY_ID%TYPE;
     OUTPUT VARCHAR2(1000);
begin
   select L.STREET_ADDRESS  ,L.POSTAL_CODE,  L.CITY ,C.COUNTRY_NAME,  R.REGION_NAME,C.REGION_ID, L.COUNTRY_ID,
    L.LOCATION_ID  INTO  F_STREET_ADDRESS,  F_POSTAL_CODE,
 F_CITY,  F_COUNTRY_NAME,F_REGION_NAME,F_REGION_ID,F_COUNTRY_ID,F_LOCATION_ID FROM LOCATIONS L join  countries C on L.country_id = C.country_id
JOIN regions R on R.REGION_ID= C.REGION_ID where   L.LOCATION_ID = F_LOCATION_I;
   OUTPUT := 'REGION_NAME: '|| F_REGION_NAME||CHR(10)||'COUNTRY_NAME: '||F_COUNTRY_NAME||CHR(10)||'LOCATION ID:  '
       ||F_LOCATION_ID||CHR(10)||'STREET ADDRESS: '|| F_STREET_ADDRESS||CHR(10)||'POSTAL CODE: '||F_POSTAL_CODE||CHR(10)||'CITY: '|| F_CITY;
    return OUTPUT;
end;

FUNCTION  GET_LOC(F_CITY OUT LOCATIONS.CITY%TYPE,  F_STREET_ADDRESS OUT LOCATIONS.STREET_ADDRESS%TYPE,F_LOCATION_I in  number)
return  VARCHAR2
is
     OUTPUT VARCHAR2(1000);
BEGIN
SELECT STREET_ADDRESS, CITY INTO F_STREET_ADDRESS, F_CITY
FROM LOCATIONS
WHERE  LOCATION_ID= F_LOCATION_I;

OUTPUT := 'LOCATION ID:  '||F_LOCATION_I||CHR(10)||'STREET ADDRESS: '||F_STREET_ADDRESS||CHR(10)||'CITY: '|| F_CITY;
       RETURN OUTPUT;
END ;
END;
```

2-

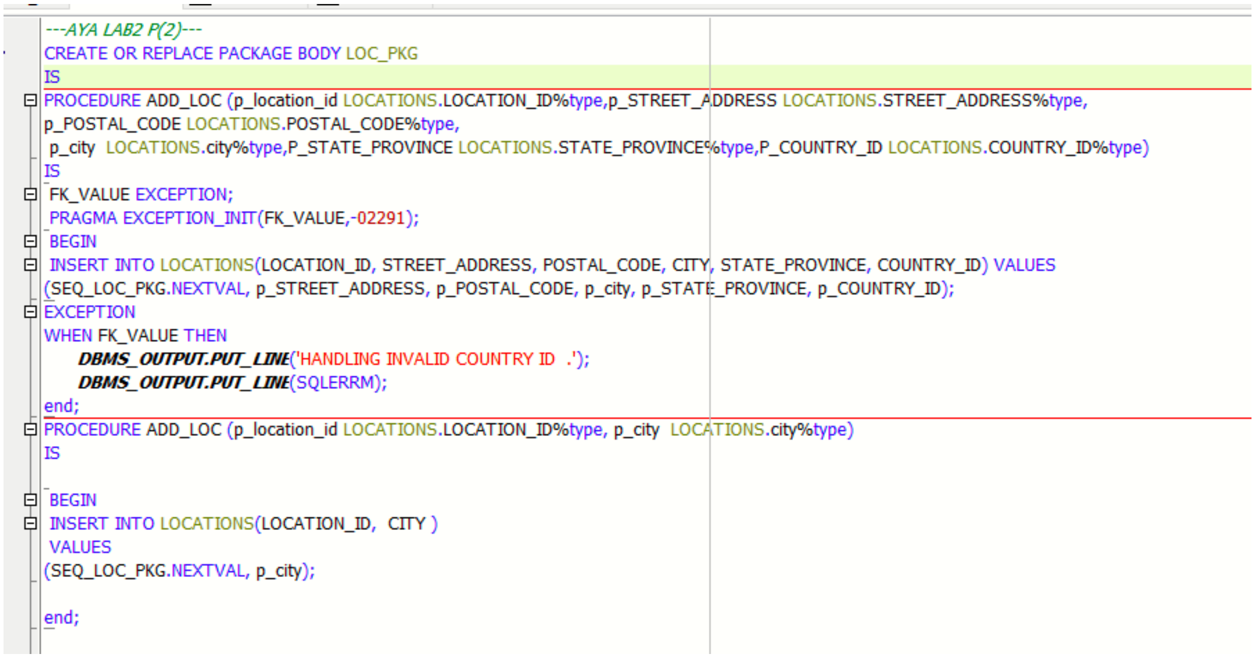
Copy and modify the code for the LOC_PKG package that you created and overload the ADD_LOC procedure. As you can insert only city only, and use sequence to insert location_id.

CREATE SEQUENCE SEQ_LOC_PKG

START WITH 3300
INCREMENT BY 100;



```
---AYA LAB2 P(2)---
CREATE OR REPLACE PACKAGE LOC_PKG
IS
 PROCEDURE ADD_LOC (p_location_id LOCATIONS.LOCATION_ID%type,p_STREET_ADDRESS LOCATIONS.STREET_ADDRESS%type,p_POSTAL_CODE LOCATIONS.POSTAL_CODE%type,
 p_city  LOCATIONS.city%type,P_STATE_PROVINCE LOCATIONS.STATE_PROVINCE%type,P_COUNTRY_ID LOCATIONS.COUNTRY_ID%type);
  PROCEDURE ADD_LOC (p_location_id LOCATIONS.LOCATION_ID%type, p_city  LOCATIONS.city%type);

FUNCTION Query_LOC(F_LOCATION_I in  number)
return  VARCHAR2;

FUNCTION  GET_LOC(F_CITY OUT LOCATIONS.CITY%TYPE,  F_STREET_ADDRESS OUT LOCATIONS.STREET_ADDRESS%TYPE,F_LOCATION_I in  number)
return  VARCHAR2;

END;
```



```
---AYA LAB2 P(2)---
CREATE OR REPLACE PACKAGE BODY LOC_PKG
IS
PROCEDURE ADD_LOC (p_location_id LOCATIONS.LOCATION_ID%type,p_STREET_ADDRESS LOCATIONS.STREET_ADDRESS%type,
p_POSTAL_CODE LOCATIONS.POSTAL_CODE%type,
 p_city  LOCATIONS.city%type,P_STATE_PROVINCE LOCATIONS.STATE_PROVINCE%type,P_COUNTRY_ID LOCATIONS.COUNTRY_ID%type)
IS
 FK_VALUE EXCEPTION;
 PRAGMA EXCEPTION_INIT(FK_VALUE,-02291);
 BEGIN
 INSERT INTO LOCATIONS(LOCATION_ID, STREET_ADDRESS, POSTAL_CODE, CITY, STATE_PROVINCE, COUNTRY_ID) VALUES
(SEQ_LOC_PKG.NEXTVAL, p_STREET_ADDRESS, p_POSTAL_CODE, p_city, p_STATE_PROVINCE, p_COUNTRY_ID);
EXCEPTION
WHEN FK_VALUE THEN
      DBMS_OUTPUT.PUT_LINE('HANDLING INVALID COUNTRY ID  .');
      DBMS_OUTPUT.PUT_LINE(SQLERRM);
end;
PROCEDURE ADD_LOC (p_location_id LOCATIONS.LOCATION_ID%type, p_city  LOCATIONS.city%type)
IS

 BEGIN
 INSERT INTO LOCATIONS(LOCATION_ID,  CITY )
 VALUES
(SEQ_LOC_PKG.NEXTVAL, p_city);

end;
```

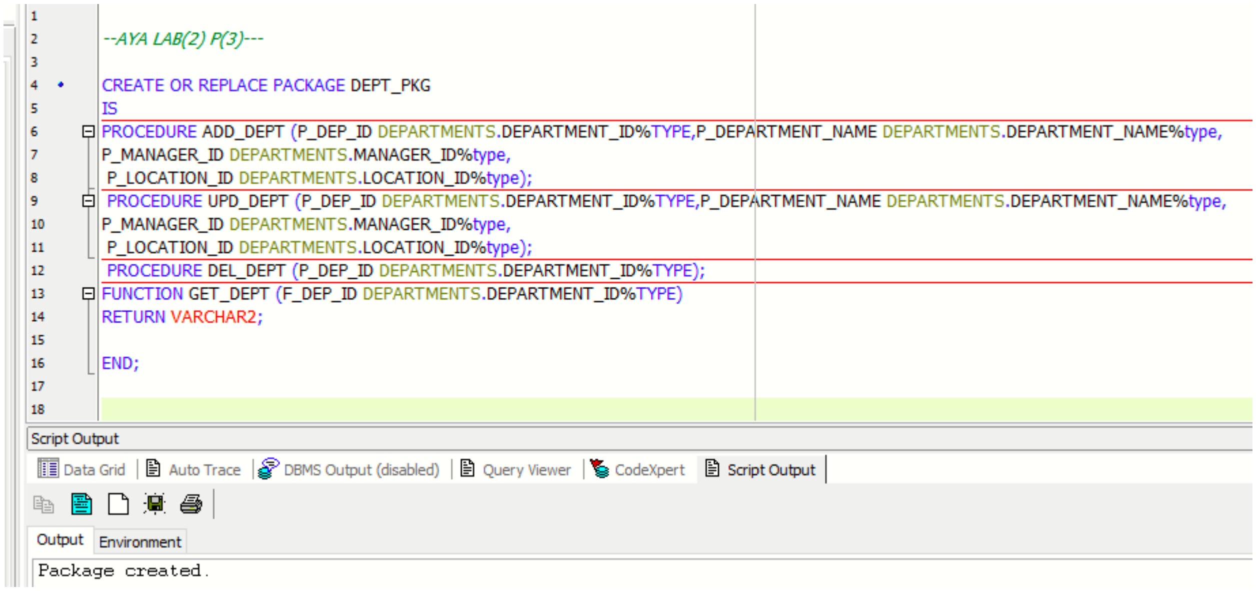
3-

Create a package specification and body called DEPT_PKG, be creating ADD_DEPT (4 params) , UPD_DEPT (4 params), and DEL_DEPT(1 param) procedures as well as your GET_DEPT function(1 param) return dept_name only.

CREATE SEQUENCE SEQ_DEPT_PKG

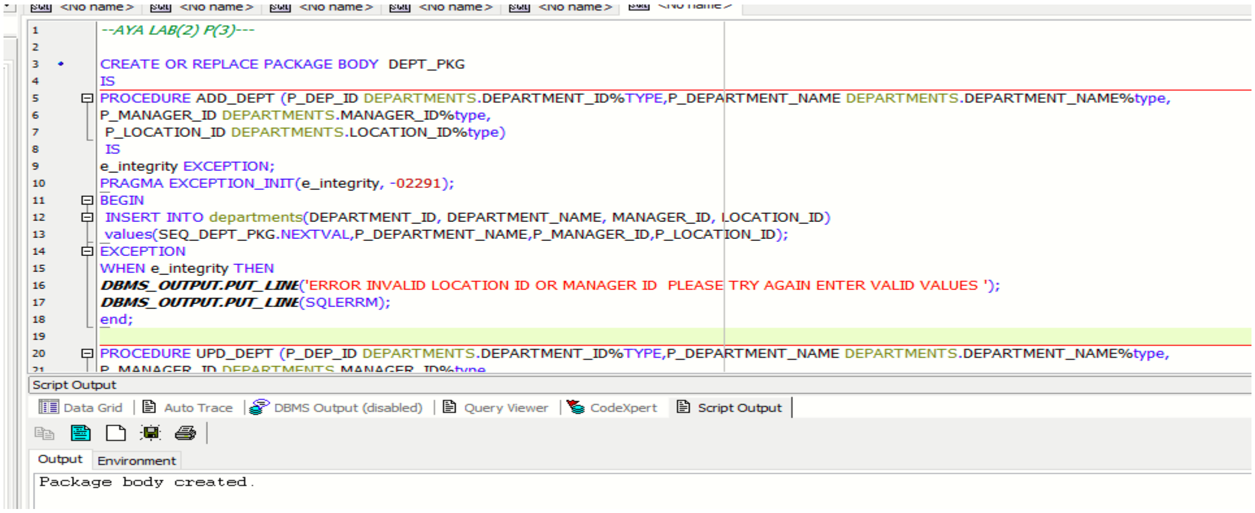
START WITH 290

INCREMENT BY 10;



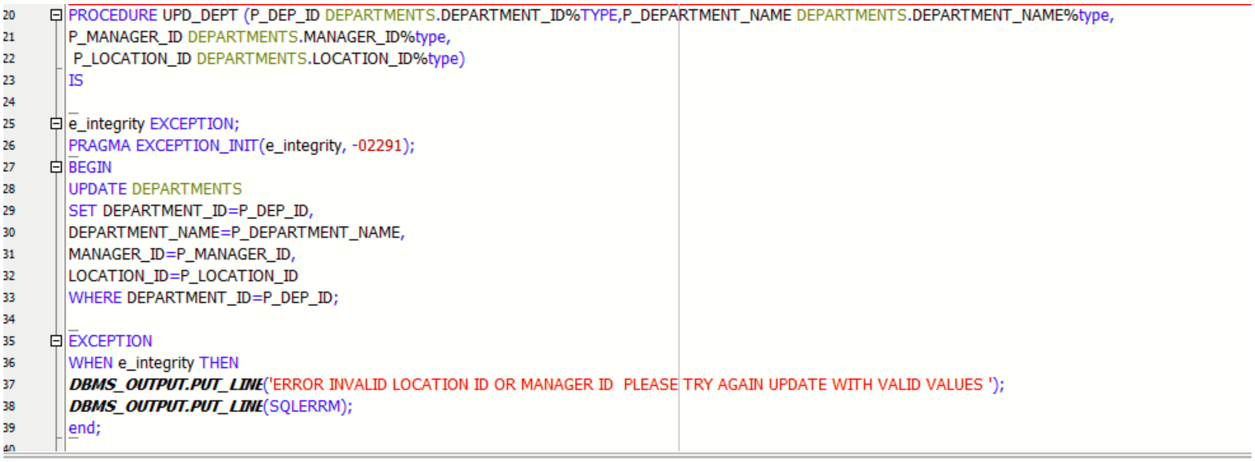
```
--AYA LAB(2) P(3)---

CREATE OR REPLACE PACKAGE DEPT_PKG
IS
PROCEDURE ADD_DEPT (P_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE,P_DEPARTMENT_NAME DEPARTMENTS.DEPARTMENT_NAME%type,
P_MANAGER_ID DEPARTMENTS.MANAGER_ID%type,
 P_LOCATION_ID DEPARTMENTS.LOCATION_ID%type);
 PROCEDURE UPD_DEPT (P_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE,P_DEPARTMENT_NAME DEPARTMENTS.DEPARTMENT_NAME%type,
P_MANAGER_ID DEPARTMENTS.MANAGER_ID%type,
 P_LOCATION_ID DEPARTMENTS.LOCATION_ID%type);
 PROCEDURE DEL_DEPT (P_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE);
FUNCTION GET_DEPT (F_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE)
RETURN VARCHAR2;

END;
```

```
Package created.
```

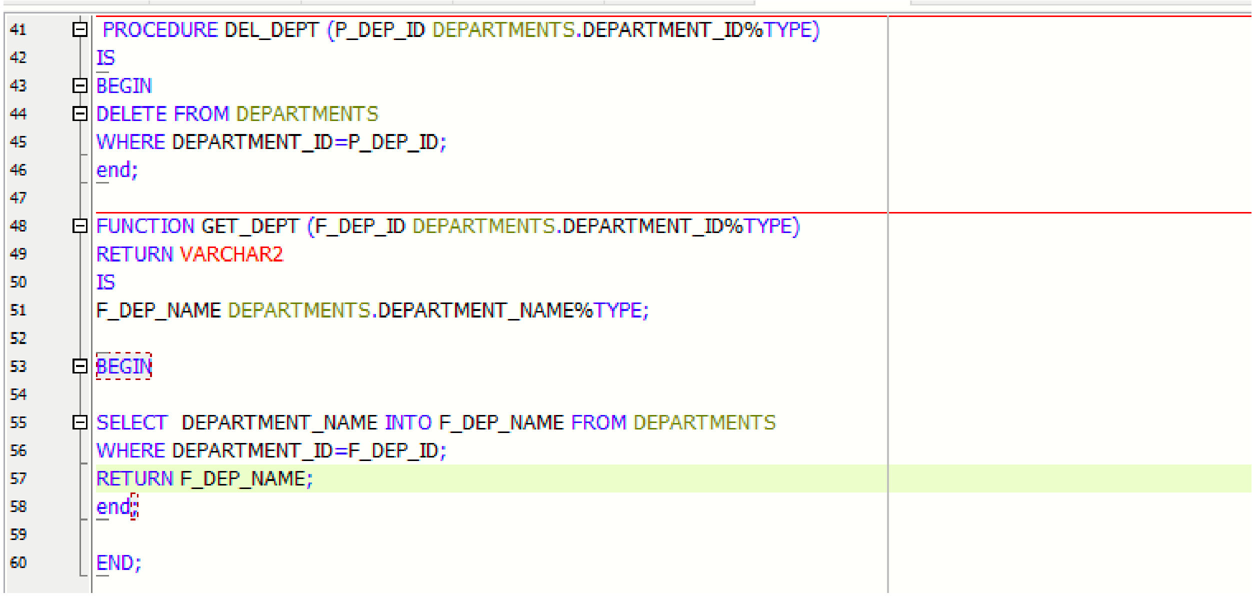


```
--AYA LAB(2) P(3)---

CREATE OR REPLACE PACKAGE BODY  DEPT_PKG
IS
PROCEDURE ADD_DEPT (P_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE,P_DEPARTMENT_NAME DEPARTMENTS.DEPARTMENT_NAME%type,
P_MANAGER_ID DEPARTMENTS.MANAGER_ID%type,
 P_LOCATION_ID DEPARTMENTS.LOCATION_ID%type)
 IS
e_integrity EXCEPTION;
PRAGMA EXCEPTION_INIT(e_integrity, -02291);
BEGIN
 INSERT INTO departments(DEPARTMENT_ID, DEPARTMENT_NAME, MANAGER_ID, LOCATION_ID)
 values(SEQ_DEPT_PKG.NEXTVAL,P_DEPARTMENT_NAME,P_MANAGER_ID,P_LOCATION_ID);
EXCEPTION
WHEN e_integrity THEN
DBMS_OUTPUT.PUT_LINE('ERROR INVALID LOCATION ID OR MANAGER ID  PLEASE TRY AGAIN ENTER VALID VALUES ');
DBMS_OUTPUT.PUT_LINE(SQLERRM);
end;

PROCEDURE UPD_DEPT (P_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE,P_DEPARTMENT_NAME DEPARTMENTS.DEPARTMENT_NAME%type,
P_MANAGER_ID DEPARTMENTS.MANAGER_ID%type
```

```
Package body created.
```



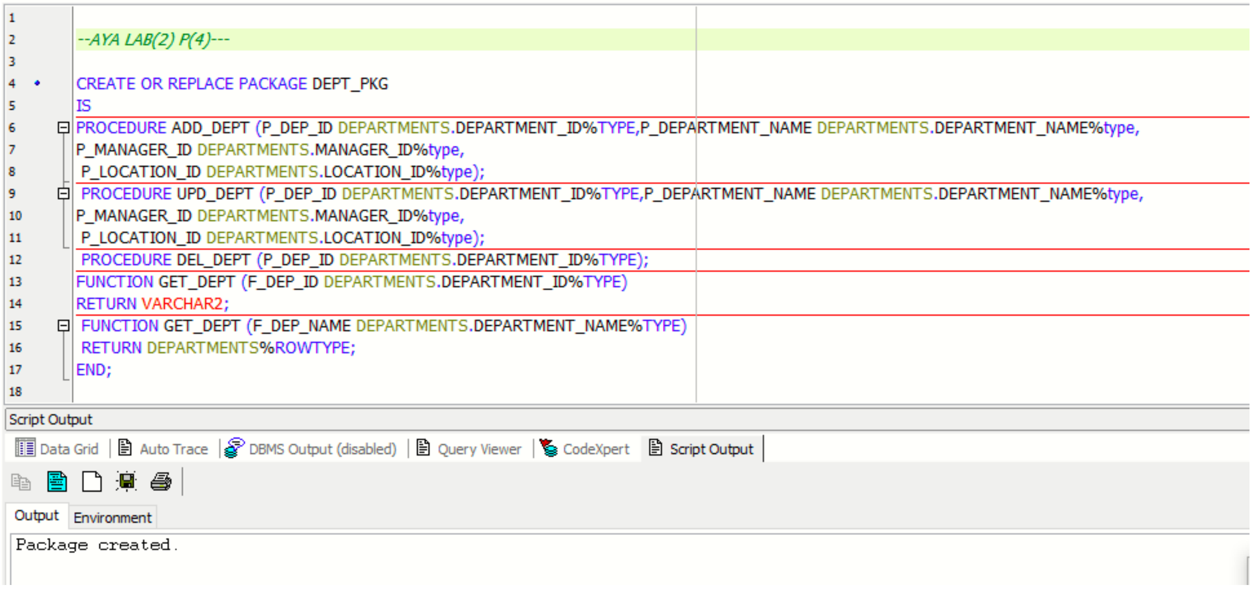
```
PROCEDURE UPD_DEPT (P_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE,P_DEPARTMENT_NAME DEPARTMENTS.DEPARTMENT_NAME%type,
P_MANAGER_ID DEPARTMENTS.MANAGER_ID%type,
 P_LOCATION_ID DEPARTMENTS.LOCATION_ID%type)
IS

e_integrity EXCEPTION;
PRAGMA EXCEPTION_INIT(e_integrity, -02291);
BEGIN
UPDATE DEPARTMENTS
SET DEPARTMENT_ID=P_DEP_ID,
DEPARTMENT_NAME=P_DEPARTMENT_NAME,
MANAGER_ID=P_MANAGER_ID,
LOCATION_ID=P_LOCATION_ID
WHERE DEPARTMENT_ID=P_DEP_ID;

EXCEPTION
WHEN e_integrity THEN
DBMS_OUTPUT.PUT_LINE('ERROR INVALID LOCATION ID OR MANAGER ID  PLEASE TRY AGAIN UPDATE WITH VALID VALUES ');
DBMS_OUTPUT.PUT_LINE(SQLERRM);
end;
```



```
 PROCEDURE DEL_DEPT (P_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE)
IS
BEGIN
DELETE FROM DEPARTMENTS
WHERE DEPARTMENT_ID=P_DEP_ID;
end;

FUNCTION GET_DEPT (F_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE)
RETURN VARCHAR2
IS
F_DEP_NAME DEPARTMENTS.DEPARTMENT_NAME%TYPE;

BEGIN

SELECT  DEPARTMENT_NAME INTO F_DEP_NAME FROM DEPARTMENTS
WHERE DEPARTMENT_ID=F_DEP_ID;
RETURN F_DEP_NAME;
end;

END;
```

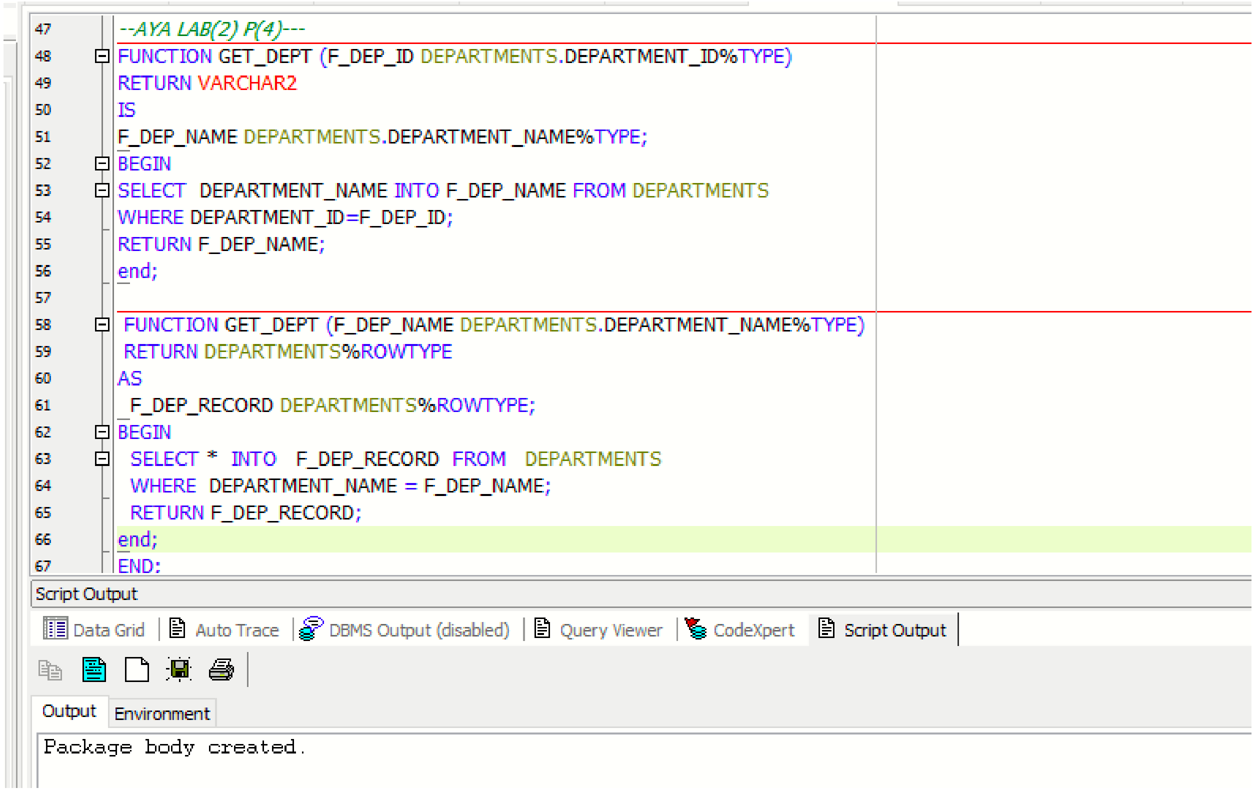
4-

Modify DEPT_PKG by adding GET_DEPT function overloading that accepts DEPT_name, and return %departments row type.



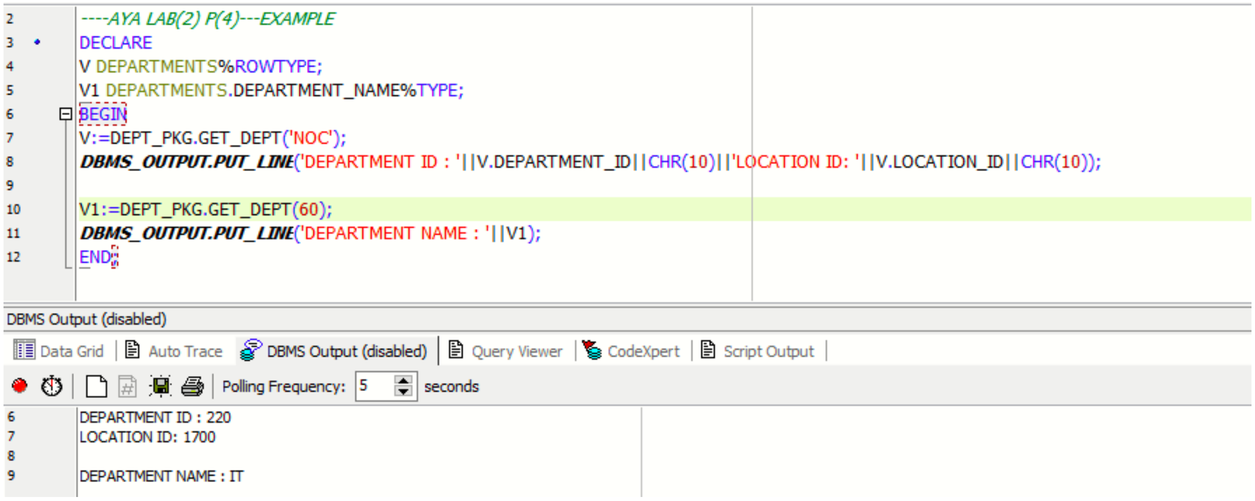
```
--AYA LAB(2) P(4)---

CREATE OR REPLACE PACKAGE DEPT_PKG
IS
PROCEDURE ADD_DEPT (P_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE,P_DEPARTMENT_NAME DEPARTMENTS.DEPARTMENT_NAME%type,
P_MANAGER_ID DEPARTMENTS.MANAGER_ID%type,
 P_LOCATION_ID DEPARTMENTS.LOCATION_ID%type);
 PROCEDURE UPD_DEPT (P_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE,P_DEPARTMENT_NAME DEPARTMENTS.DEPARTMENT_NAME%type,
P_MANAGER_ID DEPARTMENTS.MANAGER_ID%type,
 P_LOCATION_ID DEPARTMENTS.LOCATION_ID%type);
 PROCEDURE DEL_DEPT (P_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE);
FUNCTION GET_DEPT (F_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE)
RETURN VARCHAR2;
 FUNCTION GET_DEPT (F_DEP_NAME DEPARTMENTS.DEPARTMENT_NAME%TYPE)
 RETURN DEPARTMENTS%ROWTYPE;
END;

```

```
Package created.
```



```
--AYA LAB(2) P(4)---
FUNCTION GET_DEPT (F_DEP_ID DEPARTMENTS.DEPARTMENT_ID%TYPE)
RETURN VARCHAR2
IS
F_DEP_NAME DEPARTMENTS.DEPARTMENT_NAME%TYPE;
BEGIN
SELECT  DEPARTMENT_NAME INTO F_DEP_NAME FROM DEPARTMENTS
WHERE DEPARTMENT_ID=F_DEP_ID;
RETURN F_DEP_NAME;
end;

 FUNCTION GET_DEPT (F_DEP_NAME DEPARTMENTS.DEPARTMENT_NAME%TYPE)
 RETURN DEPARTMENTS%ROWTYPE
AS
  F_DEP_RECORD DEPARTMENTS%ROWTYPE;
BEGIN
  SELECT *  INTO   F_DEP_RECORD  FROM   DEPARTMENTS
  WHERE  DEPARTMENT_NAME = F_DEP_NAME;
  RETURN F_DEP_RECORD;
end;
END;
```

```
Package body created.
```



```
----AYA LAB(2) P(4)---EXAMPLE
DECLARE
V DEPARTMENTS%ROWTYPE;
V1 DEPARTMENTS.DEPARTMENT_NAME%TYPE;
BEGIN
V:=DEPT_PKG.GET_DEPT('NOC');
DBMS_OUTPUT.PUT_LINE('DEPARTMENT ID : '||V.DEPARTMENT_ID||CHR(10)||'LOCATION ID: '||V.LOCATION_ID||CHR(10));

V1:=DEPT_PKG.GET_DEPT(60);
DBMS_OUTPUT.PUT_LINE('DEPARTMENT NAME : '||V1);
END;
```

Polling Frequency: 5 seconds

```
DEPARTMENT ID : 220
LOCATION ID: 1700

DEPARTMENT NAME : IT
```