**Mã Đề 2**

MSSV: 16520177

Lê Trường Danh

**Câu 1**

CREATE tablespace tbs\_user datafile ' userdata \_01.dbf' size 10M

CREATE user scott IDENTIFIED BY tiger

DEFAULT tablespace tbs\_user;

**Câu 2**

--function nhan doi so payment\_id, tra ra thong tin email cua staff da lap payment nay

CREATE OR REPLACE FUNCTION func\_printf\_email (paymentid PAYMENT.PAYMENT\_ID%TYPE)

RETURN STAFF.EMAIL%TYPE

IS

vemail STAFF.EMAIL%TYPE;

BEGIN

SELECT s.EMAIL INTO vemail

FROM PAYMENT p JOIN STAFF s ON p.STAFF\_ID = s.STAFF\_ID

WHERE p.PAYMENT\_ID = paymentid;

RETURN vemail;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Khong tim thay');

END;

--Viet procedure nhan doi so la title cua phim, in ra cac stored\_id luu tru phim nay

select \* from FILM

CREATE OR REPLACE PROCEDURE proc\_printf\_storeid (titlefilm FILM.TITLE%TYPE)

IS

TYPE arr\_storeid\_T IS TABLE OF STORE.STORE\_ID%TYPE;

arr\_storeid arr\_storeid\_T;

BEGIN

SELECT s.STORE\_ID bulk collect INTO arr\_storeid

FROM FILM f, STORE s, INVENTORY i

WHERE s.STORE\_ID = i.STORE\_ID AND f.FILM\_ID = i.FILM\_ID AND f.TITLE = titlefilm;

FOR i IN arr\_storeid.FIRST..arr\_storeid.LAST

LOOP

DBMS\_OUTPUT.PUT\_LINE(arr\_storeid(i));

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Khong tim thay');

END;

--

set SERVEROUTPUT ON

DECLARE

idstaff STAFF.EMAIL%TYPE;

BEGIN

idstaff := func\_printf\_email('559');

DBMS\_OUTPUT.PUT\_LINE(idstaff);

END;

-- KQ: Jon.Stephens@sakilastaff.com

--

BEGIN

proc\_printf\_storeid('FLYING HOOK');

END;

--KQ: 1 1 2 2 2 2

**Câu 3**

CREATE OR REPLACE PACKAGE pakage\_FILM

AS

FUNCTION func\_printf\_film (filmid FILM.FILM\_ID%TYPE)

RETURN FILM%ROWTYPE;

PROCEDURE proc\_printf\_info (film\_T FILM%ROWTYPE);

END;

CREATE OR REPLACE PACKAGE BODY pakage\_FILM

AS

FUNCTION func\_printf\_film (filmid FILM.FILM\_ID%TYPE)

RETURN FILM%ROWTYPE

IS

vfilm FILM%ROWTYPE;

BEGIN

SELECT f.FILM\_ID, f.TITLE, f.DESCRIPTION, f.RELEASE\_YEAR,

f.LANGUAGE\_ID, f.LENGTH, f.RATING, f.SPECIAL\_FEATURES INTO vfilm

FROM FILM f

WHERE f.FILM\_ID = filmid;

RETURN vfilm;

END func\_printf\_film;

PROCEDURE proc\_printf\_info (film\_T FILM%ROWTYPE)

AS

vtitle FILM.TITLE%TYPE;

vrating FILM.RATING%TYPE;

BEGIN

SELECT f.TITLE, f.RATING INTO vtitle, vrating

FROM FILM f

WHERE f.FILM\_ID = film\_T.FILM\_ID;

DBMS\_OUTPUT.PUT\_LINE( vtitle || CHR(9) || vrating );

EXCEPTION

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Khong tim thay');

END proc\_printf\_info;

END;

DECLARE

filmrow FILM%ROWTYPE;

BEGIN

filmrow := pakage\_FILM.func\_printf\_film('737');

pakage\_FILM.proc\_printf\_info(filmrow);

END;

KQ:



**Câu 4**

SET DEFONE ON;

CREATE OR REPLACE TRIGGER trigger\_release\_year

BEFORE INSERT OR UPDATE

ON FILM

FOR EACH ROW

BEGIN

IF(:NEW.RELEASE\_YEAR <= 0) THEN

RAISE\_APPLICATION\_ERROR (-20000, 'RELEASE\_YEAR phai lon hon 0');

END IF;

END;

SET DEFONE ON;

CREATE OR REPLACE TRIGGER trigger\_retal\_date

BEFORE INSERT OR UPDATE

ON RENTAL

FOR EACH ROW

DECLARE

createdate CUSTOMER.CREATE\_DATE%TYPE;

BEGIN

SELECT c.CREATE\_DATE INTO createdate

FROM CUSTOMER c

WHERE c.CUSTOMER\_ID = :NEW.CUSTOMER\_ID;

IF(:NEW.RENTAL\_DATE < ceatedate) THEN

RAISE\_APPLICATION\_ERROR (-20000, 'RETAL\_DATE phai lon hon hoac bang CREATE\_DATE');

END IF;

END;

Insert into FILM values (522,'LIES TREATMENT','A Fast-Paced Character Study of a Dentist And a Moose who must Defeat a Composer in The First Manned Space Station',0,1,147,'NC-17 ','Commentaries,Deleted Scenes,Behind the Scenes');

Insert into RENTAL (RENTAL\_ID,RENTAL\_DATE,INVENTORY\_ID,CUSTOMER\_ID,RETURN\_DATE,STAFF\_ID) values (10000,to\_date('19-JUN-05','DD-MON-RR'),91,406,to\_date('20-JUN-05','DD-MON-RR'),1);

**Câu 5**

--a

CREATE TYPE Dept\_T AS OBJECT(

dept\_id number,

departname varchar2(50)

);

CREATE TYPE Employee\_T AS OBJECT(

emp\_id number,

first\_name varchar2(50),

last\_name varchar2(50),

slaryUsd number,

years\_of\_experience number,

job REF Dept\_T,

MEMBER FUNCTION salary\_in\_eur RETURN number

);

CREATE OR REPLACE TYPE BODY Employee\_T IS

MEMBER FUNCTION salary\_in\_eur RETURN number

IS

BEGIN

RETURN slaryUsd \* 0.9;

END;

END;

--b

CREATE TABLE deptTBL OF Dept\_T(

PRIMARY KEY (dept\_id)

);

CREATE TABLE employeeTBL OF Employee\_T(

PRIMARY KEY (emp\_id)

);

--c

INSERT INTO deptTBL VALUES(1, 'Dept\_1');

INSERT INTO deptTBL VALUES(2, 'Dept\_2');

INSERT INTO employeeTBL VALUES(1, 'Nguyen', 'Minh', 1000 ,2 , (SELECT REF(C) FROM deptTBL C WHERE C.dept\_id=1));

INSERT INTO employeeTBL VALUES(2, 'Tran', 'Trung', 500 ,3 , (SELECT REF(C) FROM deptTBL C WHERE C.dept\_id=2));

--

SELECT e.emp\_id, e.first\_name, e.last\_name, e.job.departname, e.salary\_in\_eur() as salaryEUR

FROM employeeTBL e

