

نموذج المذاكرة الأولى

أنشئ إجرائية يمرر لها رقم الموظف تطبع جميع الاجازات لهذا الموظف وإذا لم يكن لهذا الموظف إجازات يطبع رسالة خطأ وتعيد إجرائية مجموع أيام هذه الاجازة.

```
CREATE PROCEDURE PRINT_VAC(EN NUMBER, SUMVAC OUT NUMBER) IS
CURSOR C1 IS SELECT SDATE, EDATE FROM VAC WHERE EMPNO = EN;
BEGIN
FOR I IN C1 LOOP
DBMS_OUTPUT.PUT_LINE(I.SDATE || ' ' || I.EDATE);
END LOOP;
SELECT SUM(EDATE - SDATE) INTO SUMVAC FROM VAC WHERE EMPNO
= EN;
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('NO VAC');
END;
```

أنشئ تابع يمرر له رقم القسم ويعيد اسم القسم وفي حال لم يكن القسم موجود يعيد كلمة 'no dept'.

```
CREATE FUNCTION GET_DNAME(DNO NUMBER) RETURN VARCHAR2 IS
DN VARCHAR2(50);
BEGIN
SELECT DNAME INTO DN FROM DEPT WHERE DEPTNO = DNO;
RETURN DN;
EXCEPTION
WHEN NO_DATA_FOUND THEN
RETURN 'NO DEPT';
END;
```

أنشئ user بحيث يكون مغلقاً وامنحه 7 سماحيات نظام وسماحيتي على جدول emp وسماحيتي أعمدة على جدول vac.

```
CREATE USER U1 IDENTIFIED BY XYZ ACCOUNT LOCK;
GRANT CREATE TABLE, CREATE VIEW, CREATE ROLE, CREATE
SEQUENCE,
CREATE PROCEDURE, CREATE SESSION, UNLIMITED TABLESPACE TO
U1;
GRANT INSERT, UPDATE ON EMP TO U1;
GRANT INSERT(EMPNO), UPDATE(SDATE, EDATE) ON VAC TO U1;
```

أنشئ procedure يمرر له رقم المدير تطبع أسماء الموظفين لهذا المدير ويعيد عددهم مع معالجة الحالات التالية :

1. موظف غير موجود يعيد null .
 2. موظف ليس له مدير يعيد 0.
- اكتب عبارة الاستدعاء لهذه الإجرائية في القسم 10 مع طباعة أسماء الموظفين .

```

CREATE PROCEDURE GET_CNT(MN NUMBER, CNT OUT NUMBER) IS
X NUMBER := 0;
CURSOR C1 IS SELECT E.ENAME FROM EMP E, EMP M WHERE M.EMPNO
= E.MGR AND M.EMPNO = MN;
BEGIN
SELECT EMPNO INTO CNT FROM EMP WHERE EMPNO = MN;
X := X + 1;
SELECT M.EMPNO INTO CNT FROM EMP E, EMP M WHERE M.EMPNO =
E.MGR AND E.EMPNO = MN;
FOR I IN C1 LOOP
DBMS_OUTPUT.PUT_LINE(I.ENAME);
CNT := C1%ROWCOUNT;
END LOOP;
EXCEPTION
WHEN NO_DATA_FOUND THEN
IF X = 0 THEN
CNT := NULL; ELSE CNT := 0;
END IF;
END;

DECLARE
CURSOR C1 IS SELECT ENAME FROM EMP WHERE DEPTNO = 10;
CNT NUMBER;
BEGIN
GET_CNT(7800, CNT);
DBMS_OUTPUT.PUT_LINE(CNT);
FOR I IN C1 LOOP
DBMS_OUTPUT.PUT_LINE(I.ENAME);
END LOOP;
END;

```

أنشئ Role وامنحه 8 سماحيات نظام و5 سماحيات على جدول emp .

اكتب ال view المستخدمة في المنح.

سحب سماحيات نظام وسماحيات من جدول emp .

```

CREATE ROLE R1;
GRANT CREATE TABLE, CREATE VIEW, CREATE ROLE, CREATE
SEQUENCE,
CREATE PROCEDURE, CREATE SESSION, UNLIMITED TABLESPACE,
CREATE USER TO R1;
GRANT INSERT, UPDATE, DELETE, REFERENCES, INDEX
ON EMP TO R1;
SELECT * FROM ROLE_SYS_PRIVS;
SELECT * FROM ROLE_TAB_PRIVS;
REVOKE CREATE TABLE, CREATE VIEW FROM R1;
REVOKE INSERT, UPDATE ON EMP FROM R1;

```

تابع يمرر له رقم المدير يطبع أسماء الموظفين وعددهم وعالج حالة الموظف غير موجود يعيد 1 وإذا موجود وليس له موظفين يعيد 0 .

```
CREATE FUNCTION GET_CNT(MN NUMBER) RETURN NUMBER IS
CURSOR C1 IS SELECT E.ENAME FROM EMP E, EMP M WHERE M.EMPNO
= E.MGR AND M.EMPNO = MN;
CNT NUMBER; X NUMBER := 0;
BEGIN
SELECT EMPNO INTO CNT FROM EMP WHERE EMPNO = MN;
X := X + 1;
SELECT COUNT(*) INTO CNT FROM EMP E, EMP M WHERE M.EMPNO =
E.MGR AND
M.EMPNO = MN GROUP BY M.EMPNO;
FOR I IN C1 LOOP
DBMS_OUTPUT.PUT_LINE(I.ENAME);
END LOOP;
DBMS_OUTPUT.PUT_LINE(CNT);
RETURN CNT;
WHEN NO_DATA_FOUND THEN
IF X = 0 THEN
RETURN 1 ELSE RETURN 0;
END IF;
END;
```

أنشئ profile وامنحه 4 خصائص ثم امنحه للمستخدم scott .

```
CREATE PROFILE NEW_PROFILE LIMIT
SESSION_PER_USER 1
FAILED_LOGIN_ATTEMPTS 1
PASSWORD_LOCK_TIME 1
PASSWORD_LIFE_TIME 1;
ALTER USER SCOTT PROFILE NEW_PROFILE;
```

اجعل المستخدم scott مجبر على تغيير كلمة المرور عند الدخول الثاني وامنحه 8 سماحيات نظام.

```
ALTER USER SCOTT PASSWORD EXPIRE;
GRANT CREATE TABLE, CREATE VIEW, CREATE ROLE, CREATE
SEQUENCE,
CREATE PROCEDURE, CREATE SESSION, UNLIMITED TABLESPACE,
CREATE USER TO SCOTT;
```

أنشئ function يمرر له بارمترين رقمين يعيد رقم على النحو التالي :

- إذا لم يكن البارمتران null أو 0 يعيد عدد الموظفين في السنة حسب البارمتر الأول والشهر حسب البارمتر الثاني .
- في حال كان رقم الشهر null أو 0 يعيد عدد الموظفين حسب البارمتر الأول دون الثاني.
- في حال كان الرقم الأول null أو 0 يعيد عدد الموظفين المعينين في هذا العام الحالي وحسب البارمتر الثاني يكون الشهر.
- وفي حال لا يوجد ولا موظف يعيد null .

```
CREATE OR REPLACE FUNCTION FUN_CHECK(NUM1 NUMBER, NUM2
NUMBER) RETURN NUMBER IS
CNT NUMBER;
BEGIN
IF (NUM1 IS NOT NULL OR NUM1 != 0) AND (NUM2 IS NOT NULL OR
NUM2 != 0) THEN
SELECT COUNT(*) INTO CNT FROM EMP WHERE TO_CHAR(HIRE_DATE,
'YYYY') = NUM1 AND
TO_CHAR(HIRE_DATE, 'MM') = NUM2;
ELIF (NUM1 IS NOT NULL OR NUM1 != 0) AND (NUM2 IS NULL OR
NUM2 = 0) THEN
SELECT COUNT(*) INTO CNT FROM EMP WHERE TO_CHAR(HIRE_DATE,
'YYYY') = NUM1;
ELIF (NUM1 IS NULL OR NUM1 = 0) AND (NUM2 IS NOT NULL OR
NUM2 != 0) THEN
SELECT COUNT(*) INTO CNT FROM EMP WHERE TO_CHAR(HIRE_DATE,
'YYYY') = TO_CHAR(SYSDATE, 'YYYY') AND
TO_CHAR(HIRE_DATE, 'MM') = NUM2;
END IF;
IF CNT = 0 THEN
RETURN NULL;
ELSE
RETURN CNT;
END IF;
END;
```

أنشئ procedure يمرر له رقم الموظف يطبع بداية ونهاية الاجازات وفي حال كان الموظف غير موجود يعيد رسالة وفي حال كان لا يوجد إجازات لهذا الموظف يطبع رسالة أخرى.

```
CREATE PROCEDURE PRINT_VAC(EN NUMBER) IS
CURSOR C1 IS SELECT SDATE, EDATE FROM VAC WHERE EMPNO = EN;
X NUMBER := 0; ENO NUMBER;
BEGIN
SELECT EMPNO INTO ENO FROM EMP WHERE EMPNO = EN;
X := X + 1;
SELECT SUM(EDATE - SDATE) INTO ENO FROM VAC WHERE EMPNO =
EN GROUP BY EMPNO;
FOR I IN LOOP
DBMS_OUTPUT.PUT_LINE(I.SDATE || ' ' || I.EDATE);
END LOOP;
```

```

EXCEPTION WHEN NO_DATA_FOUND THEN
IF X = 0 THEN DBMS_OUTPUT.PUT_LINE('NO EMPNO');
ELSE DBMS_OUTPUT.PUT_LINE('NO VAC');
END IF;
END;

```

المذاكرة الثانية

- الدفعة الاولى -

السؤال الأول: باكج

- (1) يطبع اسماء الموظفين وعدد الموظفين في اول استخدام للحزمة .
- (2) اجرائية تقوم بتخزين اسماء الاقسام عدد الموظفين في جدول برمجي عام يكون الفهرس فيه رقم القسم.
- (3) تابع يمرر لح رقم موظف يعيد سجل (اسم والمهنة و معاش الموظف) .
- (4) جملة استدعاء التابع.

```

CREATE PACKAGE PACK IS
TYPE REC1 IS RECORD(DN DEPT.DNAME%TYPE, CNT NUMBER);
TYPE T IS TABLE OF REC INDEX BY BINARY_INTEGER;
PROCEDURE SAVE_DEPT;
TYPE REC2 IS RECORD(EN EMP.ENAME%TYPE, JB EMP.JOB%TYPE,
SL EMP.SAL%TYPE);
FUNCTION GET_EMP(ENO NUMBER) RETURN REC2;
END PACK;

```

```

CREATE OR REPLACE PACKAGE BODY PACK IS

```

```

PROCEDURE SAVE_DEPT IS
CURSOR C1 IS SELECT DNAME, DEPT.DEPTNO, COUNT(*) EMPCNT
FROM EMP, DEPT WHERE DEPT.DEPTNO = EMP.DEPTNO
GROUP BY DNAME, DEPT.DEPTNO;
MYT T;
BEGIN
FOR I IN C1 LOOP
MYT(I.DEPTNO).DN := I.DNAME;
MYT(I.DEPTNO).CNT := I.EMPCNT;
END LOOP;
END SAVE_DEPT;

```

```

FUNCTION GET_EMP(ENO NUMBER) RETURN REC2 IS
MYR REC2;
BEGIN
SELECT ENAME, JOB, SAL INTO MYR.EN, MYR.JB, MYR.SL FROM
EMP;
RETURN MYR;
END GET_EMP;

```

```

CURSOR C1 IS SELECT ENAME FROM EMP; X NUMBER;
BEGIN
FOR I IN C1 LOOP
DBMS_OUTPUT.PUT_LINE(I.ENAME);
X := C1%ROWCOUNT;
END LOOP;
DBMS_OUTPUT.PUT_LINE(X);
END PACK;

DECLARE
MYREC PACK.REC2;
BEGIN
MYREC := PACK.GET_EMP(7800);
END;

```

السؤال الثاني: Trigger يعمل عند الادخال على جدول لاجازات يقوم :

- (1) اذا كان رقم الموظف غير موجود (في جدول emp) او null عندها يمنع من الادخال .
- (2) اذا كانت sdata is null يعملها تصير تساوي تاريخ الحاسب.
- (3) اذا كانت edata is null يعمل عدد ايام الاجازة يوم واحد.

```

CREATE TRIGGER TR BEFORE INSERT ON VAC FOR EACH ROW
DECLARE X NUMBER;
BEGIN
IF :NEW.EMPNO IS NULL THEN
RAISE_APPLICATION_ERROR(-20000, 'CAN NOT INSERT..EMPNO
IS NULL');
ELSE
SELECT EMPNO INTO X FROM EMP WHERE EMPNO = :NEW.EMPNO;
END IF;
IF :NEW.SDATE IS NULL THEN
:NEW.SDATE := SYSDATE;
END IF;
IF :NEW.EDATE IS NULL THEN
:NEW.EDATE := :NEW.SDATE + 1;
END IF;
EXCEPTION WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20001, 'CAN NOT INSERT..EMPNO
NOT FOUNT');
END TR;

```

- الدفعة الثانية -

السؤال الاول:

انشئ حزمة تحوي ما يلي :

- (1) اجرائية تعيد جدول برمجي يحوي اسماء الموظفين وعدد الايام الإجازات في عام ٢٠٢٠ يكون الفهرس فيه رقم الموظف.

- (2) اجرائية يمرر لها رقم القسم تخزن اسماء الموظفين ومعاشاتهم في ملف على المخدم .
(3) استدعاء الاجرائية الاولى .

```
CREATE PACKAGE PACK IS
TYPE REC IS RECORD(EN EMP.ENAME%TYPE, VCNT NUMBER);
TYPE T IS TABLE OF REC INDEX BY BINARY_INTEGER;
PROCEDURE GET_EVAC(MYT OUT T);
PROCEDURE SAVE_EMP(DNO NUMBER);
END PACK;

CREATE OR REPLACE PACKAGE BODY PACK IS

PROCEDURE GET_EVAC(MYT OUT T) IS
CURSOR C1 IS SELECT ENAME, EMP.EMPNO, SUM(EDATE - SDATE)
SUMVAC FROM EMP, VAC WHERE EMP.EMPNO = VAC.EMPNO
AND TO_CHAR(SDATE, 'YYYY') = 2020 GROUP BY ENAME,
EMP.EMPNO;
BEGIN
FOR I IN C1 LOOP
MYT(I.EMPNO).EN := I.ENAME;
MYT(I.EMPNO).VCNT := I.SUMVAC;
END LOOP;
END GET_EVAC;

PROCEDURE SAVE_EMP(DNO NUMBER) IS
CURSOR C1 IS SELECT ENAME, SAL FROM EMP WHERE DEPTNO =
DNO;
FIL UTL_FILE.FILE_TYPE;
BEGIN
FIL := UTL_FILE.FOPEN('DIR', 'T.TXT', 'W');
FOR I IN C1 LOOP
UTL_FILE.PUT_LINE(FIL, I.ENAME || I.SAL);
END LOOP;
UTL_FILE.FCLOSE(FIL);
END SAVE_EMP;
END PACK;
DECLARE
MYTAB PACK.T;
BEGIN
PACK.GET_EVAC(MYTAB);
END;
```

أنشئ Trigger يقوم بما يلي :

- (1) منع جميع الموظفين من انشاء جدول من الساعة 8 صباحاً الى الساعة 2 ظهراً.
- (2) منع حذف العمود empno من جدول emp لجميع المستخدمين.
- (3) منع حذف الجداول نهائياً .
- (4) منع انشاء و حذف اي sequence.

```
CREATE TRIGGER TR BEFORE CREATE OR DROP OR ALTER ON
SCHEMA
BEGIN
IF ORA_SYSEVENT = 'CREATE' AND ORA_DICT_OBJ_TYPE =
'TABLE'
AND TO_CHAR(SYSDATE, 'HH24') BETWEEN 8 AND 14 THEN
RAISE_APPLICATION_ERROR(-20000, 'NO CREATE TABLE EMP');
ELSIF ORA_SYSEVENT = 'ALTER' AND ORA_DICT_OBJ_TYPE =
'TABLE' AND ORA_DICT_OBJ_NAME = 'EMP'
AND ORA_IS_DROP_COLUMN = 'EMPNO' THEN
RAISE_APPLICATION_ERROR(-20000, 'NO ALTER TABLE EMP');
ELSIF ORA_SYSEVENT = 'DROP' AND ORA_DICT_OBJ_TYPE =
'TABLE' THEN
RAISE_APPLICATION_ERROR(-20000, 'NO DROP ANY TABLE');
ELSIF ORA_SYSEVENT IN ('CREATE', 'DROP') AND
ORA_DICT_OBJ_TYPE = 'SEQUENCE' THEN
RAISE_APPLICATION_ERROR(-20000, 'NO CREATE OR DROP ANY
SEQUENCE');
END TR;
```

- الدفعة الثالثة -

السؤال الاول:

انشئ حزمة تحتوي على :

- (1) متحول محمي يقوم بتخزين تاريخ اول استخدام لهذه الحزمة.
- (2) متحول محمي يقوم بتخزين عدد مرات استخدام الحزمة.
- (3) اجرائية لطباعة الصفة الأولى.
- (4) اجرائية تقوم بتخزين رقم القسم واسم القسم ومجموع المعاشات بجدول برمجي.
- (5) استدعاء اخر اجرائية وطباعة محتويات الجدول.

```
CREATE PACKAGE PACK IS
PROCEDURE PRINT;
TYPE REC IS RECORD(DNO DEPT.DEPTNO%TYPE, DN
DEPT.DNAME%TYPE, SMSAL NUMBER);
TYPE T IS TABLE OF REC INDEX BY BINARY_INTEGER;
MYTAB T;
```



```

PROCEDURE SAVE_DEPT;
END PACK;

CREATE OR REPLACE PACKAGE BODY PACK IS
D DATE; X NUMBER;

PROCEDURE PRINT IS
BEGIN
DBMS_OUTPUT.PUT_LINE(D);
END PRINT;

PROCEDURE SAVE_DEPT IS
CURSOR C1 IS SELECT DEPT.DEPTNO, DNAME, SUM(SAL) SUMSAL
FROM EMP, DEPT WHERE EMP.DEPTNO = DEPT.DEPTNO
GROUP BY DEPT.DEPTNO, DNAME;
BEGIN
FOR IN IN C1 LOOP
MYTAB(I.DEPTNO).DNO := I.DEPTNO;
MYTAB(I.DEPTNO).DN := I.DNAME;
MYTAB(I.DEPTNO).SMSAL := I.SUMSAL;
END LOOP;
END SAVE_DEPT;

BEGIN
D := SYSDATE; X := X + 1;
END PACK;

DECLARE X NUMBER;
BEGIN
PACK.SAVE_DEPT; X := PACK.MYTAB.FIRST;
FOR I IN 1..PACK.MYTAB.COUNT LOOP
DBMS_OUTPUT.PUT_LINE(PACK.MYTAB(X).DNO ||
PACK.MYTAB(X).DN || PACK.MYTAB(X).SMSAL);
X := PACK.MYTAB.NEXT(X);
END LOOP;
END;

```

السؤال الثاني :

انشئ trigger يقوم بتخزين سماحيات insert update delete على جدول emp فقط
في ملف مخزن على السيرفر مع تخزين تاريخ منح السماحية.

```

CREATE TRIGGER TR AFTER GRANT ON DATABASE
DECLARE
PN NUMBER; P ORA_NAME_LIST_T; FIL UTL_FILE.FILE_TYPE;
BEGIN
PN := ORA_PRIVILEGE_LIST(P);
FIL := UTL_FILE.FOPEN('DIR', 'T.TXT', 'W');
FOR I IN PN LOOP
IF P(I) IN ('INSERT', 'UPDATE', 'DELETE') AND
ORA_DICT_OBJ_TYPE = 'TABLE'
AND ORA_DICT_OBJ_NAME = 'EMP' THEN
UTL_FILE.PUT_LINE(FIL, P(I) || ORA_DICT_OBJ_TYPE ||
ORA_DICT_OBJ_NAME || SYSDATE);
END IF;
END LOOP;
UTL_FILE.FCLOSE(FIL);
END TR;

```

نموذج اعادة المذاكرة الأولى

انشئ حزمة برمجية تحوي ما يلي :

- (1) سجل محمي يحوي بداية التشغيل و عدد مرات تشغيل الحزمة.
- (2) إجرائية تحوي السجل السابق .
- (3) تابع يمرر لح رقم موظف يعيد جدول برمجياً يحوي بداية الاجازة و النهاية و الفهرس هو رقم تسلسلي.
- (4) اطبع اسماء الموظفين عند بدء التشغيل .
- (5) استدعي التابع و اطبع محتويات الجدول..

```

CREATE PACKAGE PACK IS
PROCEDURE PRINT;
TYPE REC IS RECORD(SD VAC.SDATE%TYPE, ED
VAC.EDATE%TYPE);
TYPE T IS TABLE OF REC INDEX BY BINARY_INTEGER;
MYTAB T;
FUNCTION GET_VAC(EN NUMBER) RETURN T;
END PACK;

```

```

CREATE OR REPLACE PACKAGE BODY PACK IS
TYPE REC2 IS RECORD(ST DATE, CN NUMBER);
MYREC2 REC2;

```

```
PROCEDURE PRINT IS
BEGIN
DBMS_OUTPUT.PUT_LINE(REC2.ST || MYREC2.CN);
END PRINT;
```

```
FUNCTION GET_VAC(EN NUMBER) RETURN T IS
CURSOR C1 IS SELECT EMPNO, SDATE, EDATE FROM VAC WHERE
EMPNO = EN;
BEGIN
FOR I IN C1 LOOP
MYTAB(I.EMPNO).SD := I.SDATE;
MYTAB(I.EMPNO).ED := I.EDATE;
END LOOP;
RETURN MYTAB;
END GET_VAC;
```

```
CURSOR C2 IS SELECT ENAME FROM EMP;
BEGIN
MYREC2.ST := SYSDATE;
MYREC2.CN := MYREC2.CN + 1;
FOR IN IN C2 LOOP
DBMS_OUTPUT.PUT_LINE(I.ENAME);
END LOOP;
END PACK;
```

```
DECLARE
X NUMBER;
BEGIN
PACK.MYTAB := PACK.GET_VAC(7800);
X := PACK.MYTAB.FIRST;
FOR I IN 1..PACK.MYTAB.COUNT LOOP
DBMS_OUTPUT.PUT_LINE(PACK.MYTAB(X).SD ||
PACK.MYTAB(X).ED);
X := PACK.MYTAB.NEXT(X);
END LOOP;
END;
```

نموذج اعادة المذاكرة الثانية

السؤال الأول :

انشاء حزمة تحوي ما يلي :

- (1) اجرائية يمرر لها رقم موظف تعيد سجل يحوي اسم الموظف واسم قسمه ودرجته الوظيفية.
- (2) اجرائية لتخزين اسماء الموظفين و مجموع ايام اجازاتهم في عام 2022 في جدول برمجي والفهرس هو رقم القسم .
- (3) اجرائية لطباعة الجدول البرمجي.
- (4) تابع يمرر له رقم القسم يعيد عدد الموظفين في القسم مع معالجة اذا كان :
 - لا يوجد موظفين في القسم فيعيد 0.
 - القسم غير موجود فيعيد null.
- (5) كتلة برمجية لاستدعاء الاجرائية رقم 1 (رقم الموظف اي رقم يقبل).

```
CREATE PACKAGE PACK IS
TYPE REC IS RECORD(EN EMP.ENAME%TYPE, DN
DEPT.DNAME%TYPE, GD SALGRADE.GRADE%TYPE);
PROCEDURE GET_REC(E NUMBER, MYREC OUT REC);
TYPE REC2 IS RECORD(EN EMP.ENAME%TYPE, CNTVAC NUMBER);
TYPE T IS TABLE OF REC2 INDEX BY BINARY_INTEGER;
MYTAB T;
PROCEDURE SAVE_VAC;
PROCEDURE PRINT;
FUNCTION GET_CNTEMP(DNO NUMBER) RETURN NUMBER;
END PACK;
```

```
CREATE OR REPLACE PACKAGE BODY PACK IS
```

```
PROCEDURE GET_REC(E NUMBER, MYREC OUT REC) IS
BEGIN
SELECT ENAME, DNAME, GRADE INTO MYREC.EN, MYREC.DN,
MYREC.GD FROM EMP E, DEPT D, SALGRADE
WHERE D.DEPTNO = E.DEPTNO AND SAL BETWEEN LOSAL AND
HISAL AND EMPNO = E;
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('NO DATA');
END GET_REC;
```

```

PROCEDURE SAVE_VAC IS
CURSOR C1 IS SELECT ENAME, DEPTNO, SUM(EDATE - SDATE)
SUMVAC FROM EMP E, VAC V
WHERE E.EMPNO = V.EMPNO AND TO_CHAR(SDATE, 'YYYY') =
2022 GROUP BY ENAME, DEPTNO;
BEGIN
FOR I IN C1 LOOP
MYTAB(I.DEPTNO).EN := I.ENAME;
MYTAB(I.DEPTNO).CNTVAC := I.SUMVAC;
END LOOP;
END SAVE_VAC;

```

```

PROCEDURE PRINT IS
X NUMBER := MYTAB.FIRST;
BEGIN
FOR I IN 1..MYTAB.COUNT LOOP
DBMS_OUTPUT.PUT_LINE(MYTAB(X).EN || ' : ' ||
MYTAB(X).CNTVAC);
X := MYTAB.NEXT(X);
END LOOP;
END PRINT;

```

```

FUNCTION GET_CNTEMP(DNO NUMBER) RETURN NUMBER IS
X NUMBER := 0; DCNT NUMBER;
BEGIN
SELECT DEPTNO INTO DCNT FROM DEPT;
X := X + 1;
SELECT COUNT(*) INTO DCNT FROM EMP, DEPT WHERE
EMP.DEPTNO =
DEPT.DEPTNO AND EMP.DEPTNO = DNO GROUP BY EMP.DEPTNO;
RETURN DCNT;
EXCEPTION
WHEN NO_DATA_FOUND THEN
IF X = 0 THEN
RETURN NULL;
ELSE
RETURN 0;
END IF;
END GET_CNTEMP;

END PACK;

```

```

DECLARE
MYREC PACK.REC;
BEGIN
PACK.GET_REC(7800, MYREC);
END;

```

السؤال الثاني :

انشاء تريغر عند ادخال سجل على جدول emp بالشروط الاتية :

- (1) اذا ادخل المستخدم رقم القسم null فيكون قيمته 40.
- (2) اذا كان رقم القسم غير موجود في جدول dept تلغى عملية الادخال .
- (3) ادخال رقم الموظف رقم تسلسلي من اخر رقم موظف.

```

CREATE TRIGGER TR BEFORE INSERT ON EMP FOR EACH ROW
BEGIN
IF :NEW.DEPTNO IS NULL THEN
:NEW.DEPTNO := 40;
END IF;
SELECT DEPTNO INTO :NEW.DEPTNO FROM DEPT WHERE DEPTNO =
:NEW.DEPTNO;
SELECT MAX(EMPNO) INTO :NEW.EMPNO FROM EMP;
:NEW.EMPNO := NVL(:NEW.EMPNO, 0) + 1;
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20000, 'NOT FOUND IN DEPT');
END TR;

```