**Упражнение 8**

DECLARE

v\_lname VARCHAR2(15);

BEGIN

SELECT last\_name INTO v\_lname FROM employees WHERE first\_name='John';

DBMS\_OUTPUT.PUT\_LINE ('John''s last name is : ' ||v\_lname);

END;



DECLARE

v\_lname VARCHAR2(15);

BEGIN

SELECT last\_name INTO v\_lname FROM employees WHERE first\_name='John';

DBMS\_OUTPUT.PUT\_LINE ('John''s last name is : ' ||v\_lname);

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE (' Your select statement retrieved multiple rows. Consider using a cursor.');

END;



DECLARE

e\_insert\_excep EXCEPTION;

PRAGMA EXCEPTION\_INIT (e\_insert\_excep, -01400);

BEGIN

INSERT INTO departments (department\_id, department\_name) VALUES (280, NULL);

EXCEPTION

WHEN e\_insert\_excep THEN

DBMS\_OUTPUT.PUT\_LINE('INSERT OPERATION FAILED');

DBMS\_OUTPUT.PUT\_LINE(SQLERRM);

END;



SET verify off

---

DECLARE

v\_deptno NUMBER := 500;

v\_name VARCHAR2(20) := 'Testing';

e\_invalid\_department EXCEPTION;

BEGIN

UPDATE departments

SET department\_name = v\_name

WHERE department\_id = v\_deptno;

IF SQL % NOTFOUND THEN

RAISE e\_invalid\_department;

END IF;

COMMIT;

EXCEPTION

WHEN e\_invalid\_department THEN

DBMS\_OUTPUT.PUT\_LINE('No such department id.');

END;



**Изтриване на таблица messages**

DROP TABLE messages;

---

**Създаване на таблица messages**

CREATE TABLE messages (results VARCHAR2(80));

---

SET VERIFY OFF

---

**/\* Тествайте със заплата 6000; 24000; 35000 \*/**

DECLARE

v\_ename employees.last\_name%TYPE;

v\_emp\_sal employees.salary%TYPE := &Input\_salary;

BEGIN

SELECT last\_name INTO v\_ename

FROM employees

WHERE salary = v\_emp\_sal;

INSERT INTO messages (results) VALUES (v\_ename || ' - ' || v\_emp\_sal);

EXCEPTION

WHEN no\_data\_found THEN

INSERT INTO messages (results)

VALUES ('No employee with a salary of '|| TO\_CHAR(v\_emp\_sal));

WHEN too\_many\_rows THEN

INSERT INTO messages (results)

VALUES ('More than one employee with a salary of '|| TO\_CHAR(v\_emp\_sal));

WHEN others THEN

INSERT INTO messages (results) VALUES ('Some other error occurred.');

END;

---

SELECT \* FROM messages;



DECLARE

e\_childrecord\_exists EXCEPTION;

PRAGMA EXCEPTION\_INIT(e\_childrecord\_exists, -02292);

BEGIN

DBMS\_OUTPUT.PUT\_LINE(' Deleting department 40........');

DELETE FROM departments WHERE department\_id=40;

EXCEPTION

WHEN e\_childrecord\_exists THEN

DBMS\_OUTPUT.PUT\_LINE(' Cannot delete this department. There are employees in this department (child records exist.) ');

END;



---------------------------------------------------------------------

DECLARE

e\_insert\_excep EXCEPTION;

PRAGMA EXCEPTION\_INIT(e\_insert\_excep, -01400);

BEGIN

INSERT INTO departments   
 (department\_id, department\_name) VALUES (280, NULL);

EXCEPTION

WHEN e\_insert\_excep THEN

DBMS\_OUTPUT.PUT\_LINE('INSERT OPERATION FAILED');

DBMS\_OUTPUT.PUT\_LINE(SQLERRM);

END;



**Създаване на таблица staff**

CREATE TABLE staff AS SELECT \* FROM EMPLOYEES;

---------

DECLARE

v\_mgr staff.manager\_id%TYPE:=&Input\_manager\_id;

BEGIN

DELETE FROM staff

WHERE manager\_id = v\_mgr;

IF SQL%NOTFOUND THEN

RAISE\_APPLICATION\_ERROR(-20202,

'This is not a valid manager');

END IF;

END;



DECLARE

v\_mgr staff.manager\_id%TYPE:=&Input\_manager\_id;

v\_lastname staff.last\_name%TYPE;

BEGIN

select last\_name into v\_lastname FROM staff

WHERE manager\_id = v\_mgr;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR (-20201,'Manager is not a valid employee.');

END;



DECLARE

e\_name EXCEPTION;

BEGIN

DELETE FROM staff WHERE last\_name = 'Anybody';

IF SQL%NOTFOUND THEN

RAISE e\_name;

END IF;

EXCEPTION

WHEN e\_name THEN

RAISE\_APPLICATION\_ERROR (-20999, 'This is not a valid last name');

END;

-----------------------------------------------------------------------------

**Пример за част от код за улавяне на грешки и записването им в таблица errors с полета e\_user, e\_date, error\_code, error\_message.**

**DECLARE**

**error\_code NUMBER;**

**error\_message VARCHAR2(255);**

**BEGIN**

**...**

**EXCEPTION**

**...**

**WHEN OTHERS THEN**

**ROLLBACK;**

**error\_code := SQLCODE ;**

**error\_message := SQLERRM ;**

**INSERT INTO errors (e\_user, e\_date, error\_code, error\_message)**

**VALUES(USER, SYSDATE, error\_code, error\_message);**

**END;**

**Зад 1**. Напишете PL/SQL блок за намиране на имената на отдели от зададен от клавиатурата номер на локация. Вмъкнете запис в таблица mess\_dept ( името на отдела и локацията, системната дата съответно в полета RESULTS и dat). Включете изключения при:

* Не намерен отдел - Вмъкнете запис в таблица mess\_dept с текст 'No department with a location id' и въведения номер на локация; системна дата.
* При върнати повече от един отдел за локация - Вмъкнете запис в таблица mess\_dept с текст 'More than one department with a location id ' и въведения номер на локация; системна дата.
* При поява на някаква друга грешка – Да се изведе номера и текста на грешката.

Тествайте блока последователно с 1700, 1800, 3000 и 2400 за номер на локация!

**Създаване на таблица mess\_dept**

CREATE TABLE mess\_dept (RESULTS VARCHAR2(80), dat date);

**Зад 2**. Напишете PL/SQL блок за изтриване на страна от таблица countries по въведен от клавиатурата идентификатор на страна. При ненамерени данни, вдигнете потребителска грешка с номер -20204 и текст 'This is not a valid country id' като използвате RAISE\_APPLICATION\_ERROR. Използвате PRAGMA EXCEPTION\_INIT и потребителско изключение, което да се свърже с грешка -02292, която възниква при опит да се изтрие запис с първичен ключ, към който има обръщение (външен ключ) от друга таблица.