

Laborator 7: EXCEPTII in PL/SQL

TRATAREA EXCEPȚIILOR

O excepție este un identificator PL/SQL asociat unei condiții anormale apărute în timpul execuției unui bloc PL/SQL. Invocarea unei excepții are ca efect terminarea blocului, deci ieșirea din blocul PL/SQL. Pentru evitarea unor situații de întrerupere anormală, excepția poate fi captată și poate fi specificată o rutină de tratare a acesteia.

O excepție poate fi invocată în două moduri:

- a. Apare o eroare Oracle și excepția asociată ei este automat invocată
- b. Excepția poate fi invocată în mod explicit prin instrucțiunea RAISE în cadrul blocului.

Captarea unei excepții

Dacă excepția este invocată în secțiunea executabilă a unui bloc se caută în cadrul secțiunii de tratare a excepțiilor o rutină de tratare asociată. Dacă PL/SQL poate trata excepția, ea nu este propagată în blocul exterior sau în mediul apelant, caz în care se consideră că execuția blocului s-a desfășurat cu succes.

Propagarea unei excepții

Dacă nu există o rutină pentru tratarea ei, excepția este propagată în mediul apelant, caz în care execuția blocului se termină cu eșec.

Tipuri de excepții

Tipul	Mod de manipulare
Excepții pre-definite asociate erorilor care apar cel mai frecvent în blocurile PL/SQL (de exemplu <i>NO_DATA_FOUND</i> , <i>TOO_MANY_ROWS</i> , <i>INVALID_CURSOR</i> , <i>ZERO_DIVIDE</i>)	Nu trebuie declarate, serverul Oracle le invocă în mod automat, dar trebuie tratate în secțiune EXCEPTION.
Excepții non-predefinite recunoscute de Oracle dar tratate de utilizator cu ajutorul codului de eroare returnat (de exemplu ORA- 01400).	Trebuie declarate în secțiunea declarativă. Serverul Oracle le invocă în mod automat, dar trebuie tratate în secțiune EXCEPTION.
Excepții definite de utilizator, asociate unor condiții specifice de prelucrare (de exemplu cazul în care valoarea stocului unui anumit produs este zero)	Trebuie declarate în secțiunea declarativă, invocate de către utilizator și tratate în secțiunea EXCEPTION.

Tratarea tuturor excepțiilor se realizează în secțiunea EXCEPTION a blocurilor PL/SQL astfel:

```
EXCEPTION
WHEN exception1 [OR exception2 ...] THEN
    statement1 ;
    statement2 ;
...
[WHEN exception3 [OR exception4 ...] THEN
    statement1 ;
    statement2 ;
...]
[WHEN OTHERS THEN
statement1 ;
    statement2 ;
...]
```

1. Tratarea excepțiilor predefinite ale Serverului Oracle

Acestea nu trebuie declarate, fiind definite de către Oracle Server și invocate implicit. Lista completă a excepțiilor predefinite Oracle poate fi consultată din PL/SQL Reference. Câteva exemple de excepții predefinite sunt prezentate mai jos:

Numele	Numărul erorii	Descriere
NO_DATA_FOUND	ORA-01403	O instrucțiune SELECT care ar fi trebuit să întoarcă o singură linie nu a returnat nici o linie.
TOO_MANY_ROWS	ORA-01422	O instrucțiune SELECT care ar fi trebuit să întoarcă o singură linie a returnat mai multe linii.
INVALID_CURSOR și CURSOR_ALREADY_OPEN	ORA-01001	Apariția unei operații ilegale asupra unui cursor (de exemplu încercarea de a deschide un cursor deja deschis).

Exemple:

Să se afișeze angajatul cu codul 10. Să se trateze eroarea apărută în cazul în care nu există nici un angajat cu acest cod.

```
DECLARE
v_nume VARCHAR2(20);
BEGIN
    SELECT ename INTO v_nume
    FROM emp
    WHERE empno=10;
    dbms_output.put_line(v_nume);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        dbms_output.put_line('Nu exista angajatul cu acest ID!');
END;
```

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. Below this is a section for SQL Commands, including a language dropdown set to PL/SQL, a rows limit of 10, and buttons for Clear Command and Find Tables. The main area contains a PL/SQL block:

```
1 DECLARE
2     v_ume VARCHAR2(20);
3 BEGIN
4     SELECT ename INTO v_ume
5     FROM emp
6     WHERE empno=10;
7     dbms_output.put_line(v_ume);
8     EXCEPTION
9         WHEN NO_DATA_FOUND THEN
10        dbms_output.put_line('Nu exista angajatul cu acest ID!');
11 END;
```

Below the code editor, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is active, displaying the output: "Nu exista angajatul cu acest ID!" and "Statement processed."

Să se afișeze salariul angajatului cu numele King. Să se trateze eroare apărută în cazul în care există mai mulți angajați cu același nume (interogarea SQL din bloc întoarce mai multe înregistrări).

```
DECLARE
    v_sal emp.sal%type;
BEGIN
    SELECT sal SAL v_sal
    FROM emp
    WHERE name='King';
    DBMS_OUTPUT.PUT_LINE('King are salariul de: '||v_sal);
    EXCEPTION
        WHEN TOO_MANY_ROWS THEN
            DBMS_OUTPUT.PUT_LINE('Exista mai multi salariati cu numele
            King! Utilizati un cursor pentru selectie!');
END;
```

2. Tratarea excepțiilor non-predefinite Oracle Server

Se poate capta o eroare a Serverului Oracle ce nu are asociată o excepție predefinită asociindu-i un nume codului de eroare returnat sau folosind clauza WHEN OTHERS. În PL/SQL, directiva EXCEPTION_INIT determină compilatorul să asocieze un nume de excepție unui număr (cod) de eroare standard a Serverului Oracle. Aceasta permite referirea erorii standard prin nume și scrierea unei rutine de tratare a ei.

Tratarea acestor erori se realizează în 3 pași:

- 1) Declararea excepției: se face în zona DECLARE a blocului
NUME_EXCEPTIE EXCEPTION;
- 2) Asocierea codului erorii cu excepția declarată anterior: se realizează tot în zona DECLARE prin utilizarea directivei de compilare EXCEPTION_INIT:
PRAGMA EXCEPTION_INIT(NUME_EXCEPTIE, COD_EROARE);
Unde COD_EROARE este un cod de eroare standard Oracle;
- 3) Tratarea excepției în zona EXCEPTION a blocului:
EXCEPTION
WHEN NUME_EXCEPTIE THEN

Se pot utiliza 2 atribute pentru a gestiona erorile apărute:

SQLCODE – returnează codul numeric al erorii. Poate avea următoarele valori:

- 0 – nu a apărut nici o excepție;
- 1 – este o excepție definită de utilizator;
- +100 – excepția NO_DATA_FOUND;
- un număr negativ – o eroare Oracle Server;

SQLERRM – returnează mesajul asociat erorii.

Aceste atribute pot fi încărcate în variabile și inserate într-o tabelă de erori pentru vizualizare și verificare ulterioară.

Exemplu:

Să se insereze în tabela dept un nou departament cu ID-ul 200, fără a preciza denumirea acestuia. În acest caz va apărea eroarea cu codul ORA-01400 prin care programatorul este avertizat de încălcarea unei restricții de integritate. Această excepție poate fi tratată astfel:

```
DECLARE
    -- se asociază un nume codului de eroare apărut
    INSERT_EXCEPT EXCEPTION;
    PRAGMA EXCEPTION_INIT(INSERT_EXCEPT, -01400);
BEGIN
    insert into dept (deptno, dname)
    values (200, NULL);
    EXCEPTION
    --se tratează eroarea prin numele său
    WHEN insert_except THEN
        DBMS_OUTPUT.PUT_LINE('Nu ati precizat informatii suficiente
    pentru departament');
    --se afișează mesajul erorii
    DBMS_OUTPUT.PUT_LINE(SQLERRM);
END;
```

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation menus for 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below that, the 'SQL Commands' section is active, showing 'Language' set to 'PL/SQL' and 'Rows' set to '10'. The main area contains a PL/SQL script:

```
1 DECLARE
2     -- se asociază un nume codului de eroare apărut
3     INSERT_EXCEPT EXCEPTION;
4     PRAGMA EXCEPTION_INIT(INSERT_EXCEPT, -01400);
5 BEGIN
6     insert into dept (deptno, dname)
7     values (200, NULL);
8     EXCEPTION
9     --se tratează eroarea prin numele său
10    WHEN insert_except THEN
11        DBMS_OUTPUT.PUT_LINE('Nu ati precizat informatii suficiente pentru departament');
12    --se afișează mesajul erorii
13    DBMS_OUTPUT.PUT_LINE(SQLERRM);
14 END;
```

```
ORA-01438: value larger than specified precision allowed for this column
ORA-06512: at line 6
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

```
4. PRAGMA EXCEPTION_INIT(INSERT_EXCEPT, -01400);
5. BEGIN
6. insert into dept (deptno, dname)
7. values (200, NULL);
8. EXCEPTION
```

3. Tratarea excepțiilor definite de utilizator

În PL/SQL se pot defini excepții ale utilizatorului. Ele trebuie declarate în secțiunea declarativă a blocului și invocate explicit prin instrucțiunea RAISE.

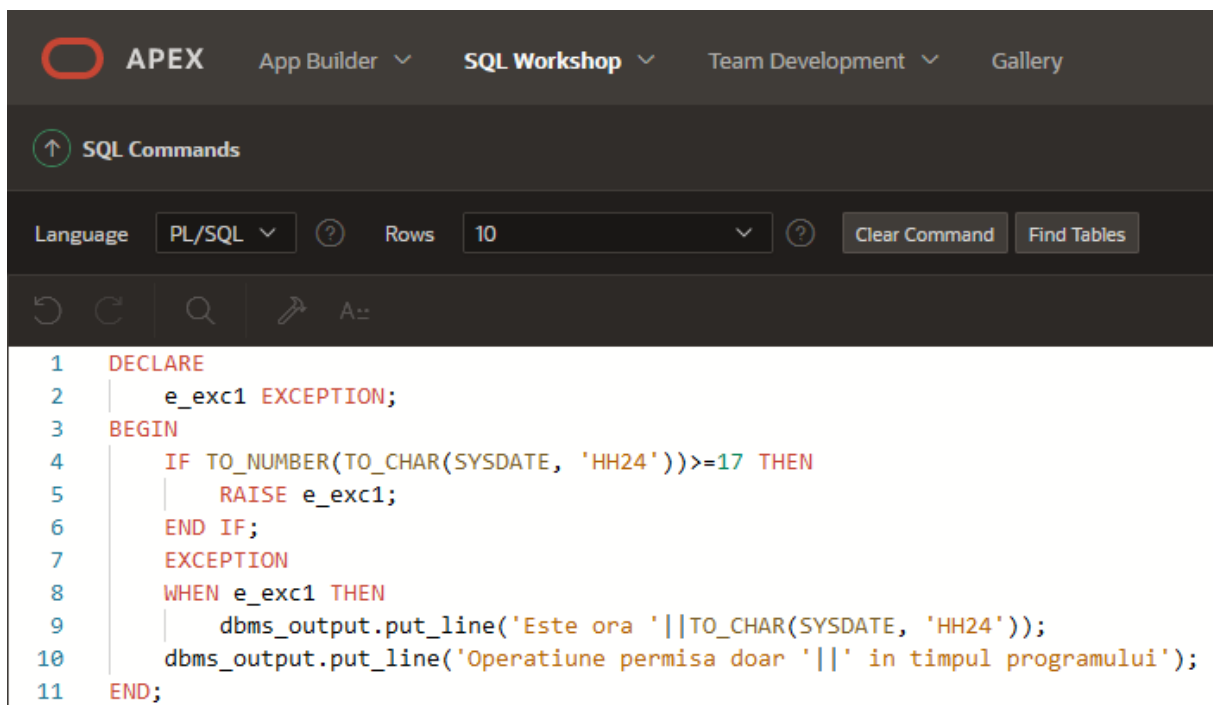
Etape:

1. Se declara excepția în secțiunea declarativă:
nume_excepție EXCEPTION;
2. Prin instrucțiunea RAISE se invocă în mod explicit, în cadrul secțiunii executabile:
RAISE nume_excepție;
3. Se tratează în rutina corespunzătoare din secțiunea de tratare a excepțiilor:
WHEN nume_excepție THEN.....

Exemple:

Să se invoce o eroare în cazul în care utilizatorul încearcă să execute blocul PL/SQL după ora 17.

```
DECLARE
    e_exc1 EXCEPTION;
BEGIN
    IF TO_NUMBER(TO_CHAR(SYSDATE, 'HH24'))>=17 THEN
        RAISE e_exc1;
    END IF;
EXCEPTION
    WHEN e_exc1 THEN
        dbms_output.put_line('Este ora '||TO_CHAR(SYSDATE,
'HH24'));
        dbms_output.put_line('Operatiune permisa doar '||' in timpul
programului');
END;
```



The screenshot shows the APEX SQL Workshop interface. At the top, there are navigation tabs for 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below this is a 'SQL Commands' section with a search icon. The main area shows the PL/SQL code from the previous block, with line numbers 1 through 11 on the left. The code is: 1 DECLARE, 2 e_exc1 EXCEPTION;, 3 BEGIN, 4 IF TO_NUMBER(TO_CHAR(SYSDATE, 'HH24'))>=17 THEN, 5 RAISE e_exc1;, 6 END IF;, 7 EXCEPTION, 8 WHEN e_exc1 THEN, 9 dbms_output.put_line('Este ora '||TO_CHAR(SYSDATE, 'HH24'));, 10 dbms_output.put_line('Operatiune permisa doar '||' in timpul programului');, 11 END;.

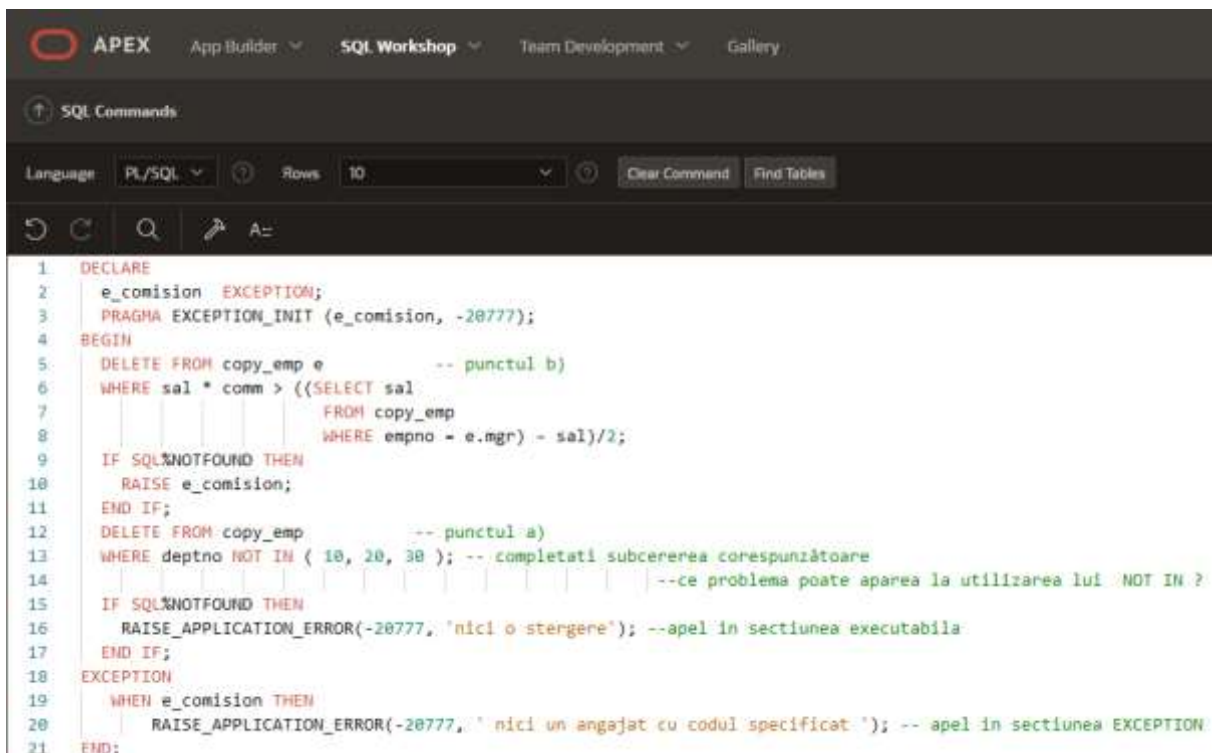
Exemplu:

a) Să se șteargă salariații asignați unui cod de departament inexistent în tabelul departments. Dacă nu există nici un angajat care a îndeplinit această condiție, să se lanseze o excepție cu mesajul „nici un angajat nu lucreaza in departament inexistent”.

b) Să se șteargă angajații al căror comision reprezintă mai mult decât jumătatea diferenței de salariu dintre șeful angajatului respectiv și angajat. Dacă nu există nici un angajat care a îndeplinit această condiție, să se lanseze o excepție cu mesajul „nici un angajat cu comisionul specificat”.

Invocați procedura RAISE_APPLICATION_ERROR în secțiuni diferite ale blocului PL/SQL.

```
DECLARE
  e_comision EXCEPTION;
  PRAGMA EXCEPTION_INIT (e_comision, -20777);
BEGIN
  DELETE FROM emp_pnu e      -- punctul b)
  WHERE sal * comm > ((SELECT sal
                        FROM emp_pnu
                        WHERE empno = e.manager_id) - sal)/2;
  IF SQL%NOTFOUND THEN
    RAISE e_comision;
  END IF;
  DELETE FROM emp_pnu      -- punctul a)
  WHERE deptno NOT IN ( ... ); -- completati subcererea corespunzătoare
                                --ce problema poate aparea la utilizarea lui NOT IN ?
  IF SQL%NOTFOUND THEN
    RAISE_APPLICATION_ERROR(-20777, 'nici o stergere'); --apel in sectiunea
    executabila
  END IF;
EXCEPTION
  WHEN e_comision THEN
    RAISE_APPLICATION_ERROR(-20777, ' nici un angajat cu codul specificat '); --
    apel in sectiunea EXCEPTION
END;
```



The screenshot shows the APEX SQL Workshop interface. At the top, there are navigation tabs for 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the tabs, there is a search bar for 'SQL Commands'. The main area displays the PL/SQL code from the previous block, with line numbers 1 through 21 on the left. The code is color-coded: DECLARE and EXCEPTION are in red, comments are in green, and the rest of the code is in black. The code is as follows:

```
1 DECLARE
2   e_comision EXCEPTION;
3   PRAGMA EXCEPTION_INIT (e_comision, -20777);
4 BEGIN
5   DELETE FROM copy_emp e      -- punctul b)
6   WHERE sal * comm > ((SELECT sal
7                         FROM copy_emp
8                         WHERE empno = e.mgr) - sal)/2;
9   IF SQL%NOTFOUND THEN
10    RAISE e_comision;
11  END IF;
12  DELETE FROM copy_emp      -- punctul a)
13  WHERE deptno NOT IN ( 10, 20, 30 ); -- completati subcererea corespunzătoare
14                                         --ce problema poate aparea la utilizarea lui NOT IN ?
15  IF SQL%NOTFOUND THEN
16    RAISE_APPLICATION_ERROR(-20777, 'nici o stergere'); --apel in sectiunea executabila
17  END IF;
18 EXCEPTION
19   WHEN e_comision THEN
20    RAISE_APPLICATION_ERROR(-20777, ' nici un angajat cu codul specificat '); -- apel in sectiunea EXCEPTION
21 END;
```

Structura tabelii **EMP**:

Object Type **TABLE** [?](#) Object **EMP** [?](#)

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMP	EMPNO	NUMBER	-	4	0	1	-	-	-
	ENAME	VARCHAR2	10	-	-	-	✓	-	-
	JOB	VARCHAR2	9	-	-	-	✓	-	-
	MGR	NUMBER	-	4	0	-	✓	-	-
	HIREDATE	DATE	7	-	-	-	✓	-	-
	SAL	NUMBER	-	7	2	-	✓	-	-
	COMM	NUMBER	-	7	2	-	✓	-	-
	DEPTNO	NUMBER	-	2	0	-	✓	-	-

Erorile definite de utilizator pot fi tratate la nivelul aplicațiilor ca și erorile Oracle Server prin atribuirea de coduri cu ajutorul funcției:

`RAISE_APPLICATION_ERROR (NR_EROAARE, MESAJ);`

unde NR_EROAARE poate fi un număr negativ cuprins între -20000 și -20999.

În acest caz tratarea se realizează asemănător cu erorile non-predefinite Oracle Server.

Structura tabelii **DEPT**:

Object Type **TABLE** [?](#) Object **DEPT** [?](#)

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPT	DEPTNO	NUMBER	-	2	0	1	-	-	-
	DNAME	VARCHAR2	40	-	-	-	✓	-	-
	LOC	VARCHAR2	13	-	-	-	✓	-	-

Exemplu:

Creați tabela MESAJE având o singură coloană, numită rezultate, de tip varchar2(50). Să se scrie un bloc PL/SQL prin care să se afișeze numele departamentelor dintr-o anumită locație care au angajați.

- a) Dacă rezultatul interogării returnează mai mult decât o linie, atunci să se trateze excepția și să se insereze în tabela MESAJE textul „mai multe departamente“.
- b) Dacă rezultatul interogării nu returnează nici o linie, atunci să se trateze excepția și să se insereze în tabela MESAJE textul „nici un departament“.
- c) Dacă rezultatul interogării este o singură linie, atunci să se insereze în tabela MESAJE numele departamentului și locația.
- d) Să se trateze orice altă eroare, inserând în tabela MESAJE textul „alte erori au apărut“.

Pasul 1 – crearea unei noi tabele numite MESAJE. Mai întâi dăm comanda de DROP TABLE mesaje pentru a șterge tabela existentă cu același nume dar cu o structură diferită.

```
DROP TABLE mesaje;
```

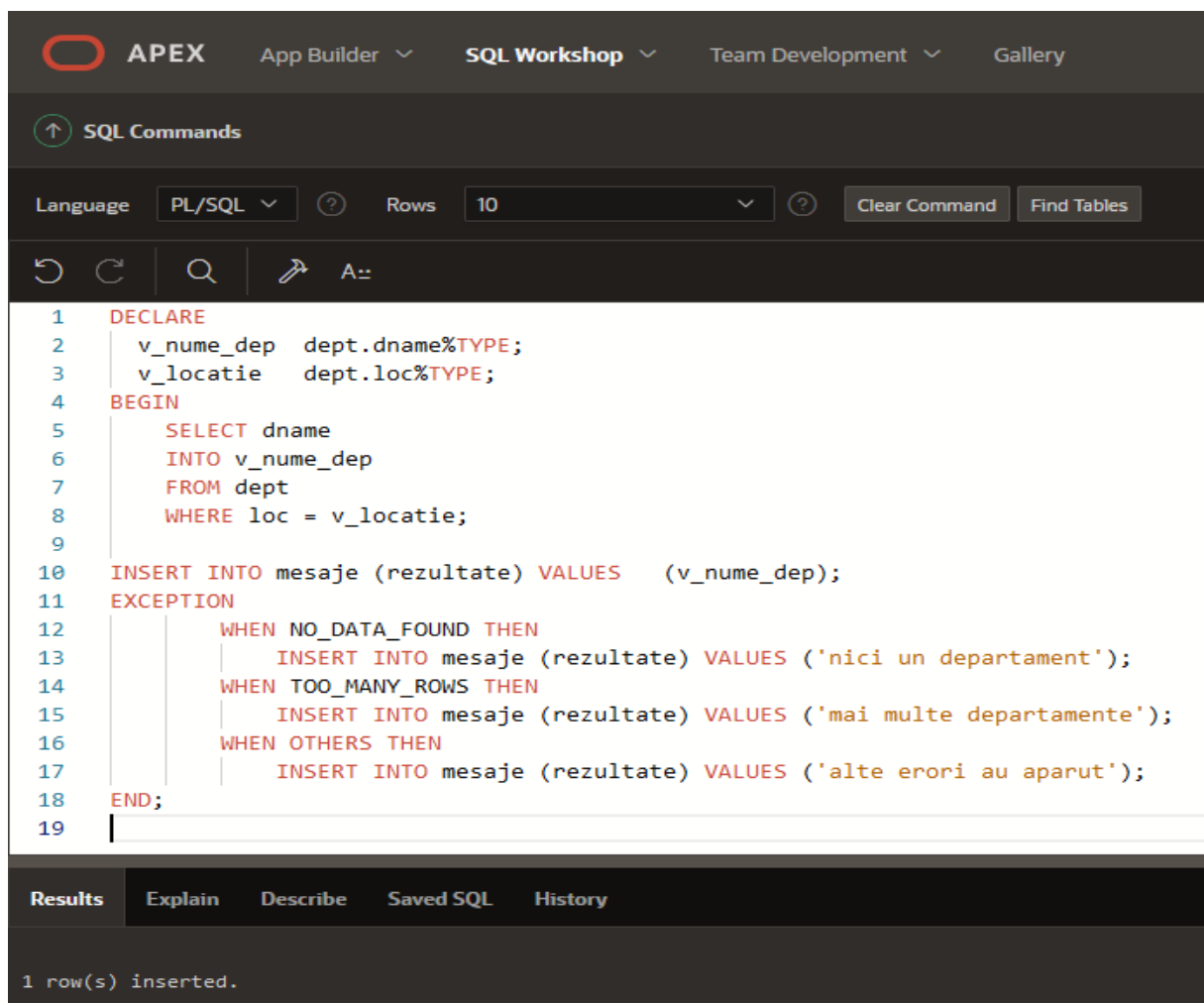
The screenshot shows the APEX SQL Workshop interface. At the top, there are navigation tabs: APEX, App Builder, SQL Workshop, Team Development, and Gallery. Below this is the 'SQL Commands' section with a dropdown menu set to 'PL/SQL' and 'Rows' set to '10'. There are buttons for 'Clear Command' and 'Find Tables'. The command editor shows the command: `DROP TABLE mesaje`. Below the editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active and displays the message: 'Table dropped.'

```
CREATE TABLE mesaje ( rezultate varchar2(50) );
```

The screenshot shows the APEX SQL Workshop interface. At the top, there are navigation tabs: APEX, App Builder, SQL Workshop, Team Development, and Gallery. Below this is the 'SQL Commands' section with a dropdown menu set to 'PL/SQL' and 'Rows' set to '10'. There are buttons for 'Clear Command' and 'Find Tables'. The command editor shows the command: `CREATE TABLE mesaje (rezultate varchar2(50));`. Below the editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active and displays the message: 'Table created.'

Pasul 2 -

```
DECLARE
  v_num dep dept.dname%TYPE;
  v_locatie dept.loc%TYPE;
BEGIN
  SELECT dname
  INTO v_num
  FROM dept
  WHERE loc = v_locatie;
INSERT INTO mesaje (rezultate) VALUES (v_num);
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    INSERT INTO mesaje (rezultate) VALUES ('nici un departament');
  WHEN TOO_MANY_ROWS THEN
    INSERT INTO mesaje (rezultate) VALUES ('mai multe departamente');
  WHEN OTHERS THEN
    INSERT INTO mesaje (rezultate) VALUES ('alte erori au aparut');
END;
```



The screenshot displays the Oracle APEX SQL Workshop interface. At the top, there are navigation tabs for 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below this is a 'SQL Commands' section with a search icon and a list of commands. The main area shows a PL/SQL script with line numbers 1 through 19. The script declares two variables, selects a department name, and inserts it into a table named 'mesaje'. It includes an exception block to handle 'NO_DATA_FOUND', 'TOO_MANY_ROWS', and 'OTHERS' errors. Below the script, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing the output: '1 row(s) inserted.'

4. Propagarea excepțiilor

Odată excepția declanșată în secțiunea executabilă a unui bloc, se caută în cadrul secțiunii de tratare a excepțiilor (EXCEPTION) o rutină de tratare asociată. Dacă PL/SQL poate trata excepția, ea nu este propagată în blocul exterior sau în mediul apelant, caz în care se consideră că execuția blocului s-a desfășurat cu succes.

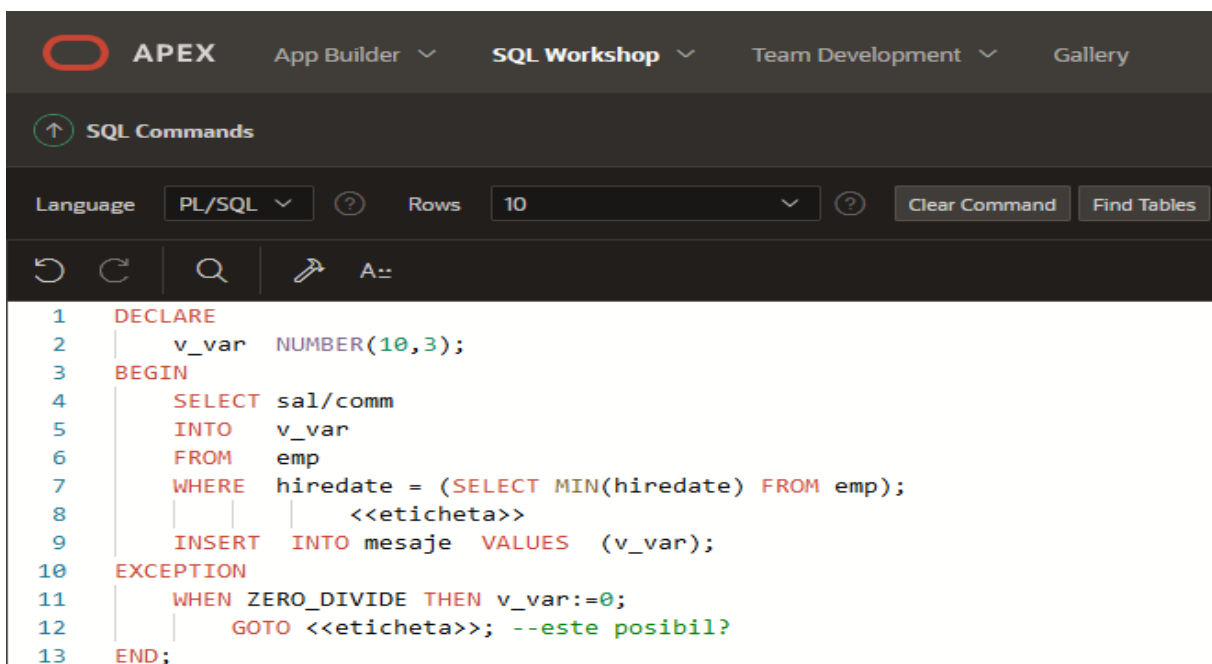
Atunci când un sub-bloc tratează o excepție, se termină normal iar execuția se reia în blocul ce-l cuprinde imediat după instrucțiunea END a sub-blocului.

Dacă apare o excepție iar în blocul curent nu există o rutină pentru tratarea sa, execuția blocului se termina cu eșec, iar excepția se propagă succesiv în blocurile exterioare până este găsită într-unul din ele o rutină pentru tratarea ei. Dacă nu se găsește nici una, în mediul apelant apare o situație de excepție nerezolvată, utilizator putând observa mesajul de eroare care a întrerupt execuția normală.

Să se rețină în tabelul MESAJE raportul dintre salariu și comision pentru angajatul având cea mai mare vechime.

Varianta 1:

```
DECLARE
    v_var NUMBER(10,3);
BEGIN
    SELECT sal/comm
    INTO v_var
    FROM emp_pnu
    WHERE hire_date = (SELECT MIN(hire_date) FROM emp_pnu);
    <<eticheta>>
    INSERT INTO mesaje_pnu VALUES (v_var);
EXCEPTION
    WHEN ZERO_DIVIDE THEN v_var:=0;
    GOTO <<eticheta>>; --este posibil?
END;
```



The screenshot shows the APEX SQL Workshop interface. At the top, there are navigation tabs: APEX, App Builder, SQL Workshop, Team Development, and Gallery. Below the tabs, there is a section for SQL Commands. The interface includes a Language dropdown set to PL/SQL, a Rows dropdown set to 10, and buttons for Clear Command and Find Tables. The main area displays the PL/SQL code from the previous block, with line numbers 1 through 13 on the left. The code is as follows:

```
1 DECLARE
2     v_var NUMBER(10,3);
3 BEGIN
4     SELECT sal/comm
5     INTO v_var
6     FROM emp
7     WHERE hiredate = (SELECT MIN(hiredate) FROM emp);
8     | | | <<eticheta>>
9     INSERT INTO mesaje VALUES (v_var);
10 EXCEPTION
11     WHEN ZERO_DIVIDE THEN v_var:=0;
12     | GOTO <<eticheta>>; --este posibil?
13 END;
```

```

ORA-06550: line 12, column 11:
PLS-00103: Encountered the symbol "<" when expecting one of the following:

    <an identifier> <a double-quoted delimited-identifier>
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_210200", line 673
ORA-06512: at "SYS.DBMS_SYS_SQL", line 1658
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_210200", line 659
ORA-06512: at "APEX_210200.WWV_FLOW_DYNAMIC_EXEC", line 1829

```

Varianta 2:

```

DECLARE
    v_var NUMBER(10,3);
BEGIN
    IF NVL(comm, 0)=0 THEN
        GOTO eticheta; --este posibil?
    END IF;
    SELECT sal/comm
    INTO v_var
    FROM emp_pnu
    WHERE hire_date = (SELECT MIN(hire_date) FROM emp_pnu);
    INSERT INTO mesaje_pnu VALUES (v_var);
EXCEPTION
    <<eticheta>>
    WHEN ZERO_DIVIDE THEN v_var:=0;
END;

```

The screenshot shows the APEX SQL Workshop interface. At the top, there are navigation tabs for 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the tabs is a 'SQL Commands' section with a search icon. The main area shows the PL/SQL code with a 'Language' dropdown set to 'PL/SQL' and 'Rows' set to '10'. The code is as follows:

```

1 DECLARE
2     v_var NUMBER(10,3);
3 BEGIN
4     IF NVL(comm, 0)=0 THEN
5         GOTO eticheta; --este posibil?
6     END IF;
7     SELECT sal/comm
8     INTO v_var
9     FROM emp
10    WHERE hire_date = (SELECT MIN(hire_date) FROM emp);
11    INSERT INTO mesaje VALUES (v_var);
12 EXCEPTION
13     <<eticheta>>
14     WHEN ZERO_DIVIDE THEN v_var:=0;
15 END;

```

```
ORA-06550: line 13, column 7:
PLS-00103: Encountered the symbol "<" when expecting one of the following:

pragma when
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_210200", line 673
ORA-06512: at "SYS.DBMS_SYS_SQL", line 1658
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_210200", line 659
ORA-06512: at "APEX_210200.WWV_FLOW_DYNAMIC_EXEC", line 1829
```

Obs: Instrucțiunea GOTO nu permite:

- ✓ saltul la secțiunea de tratare a unei excepții;
- ✓ saltul de la secțiunea de tratare a unei excepții, în blocul curent.

Comanda GOTO permite totuși saltul de la secțiunea de tratare a unei excepții la un bloc care include blocul curent.

Cum se poate remedia situația de mai sus? Schimbați blocul propus (utilizând un subbloc) astfel încât să funcționeze corespunzător.

Exemple de excepții predefinite:

1. **NO_DATA_FOUND** - Single row SELECT returned no data.
2. **too_many_rows** - Single row SELECT returned more than one row.
3. **INVALID_CURSOR** - Illegal cursor operation was attempted.
4. **VALUE_ERROR** - Arithmetic, conversion, truncation, or constraint error occurred.
5. **INVALID_NUMBER** - Conversion of a number to a character string failed.
6. **ZERO_DIVIDE** - Attempted to divide by zero.
7. **DUP_VAL_ON_INDEX** - Attempted to insert a duplicate value into a column that has a unique index.
8. **CURSOR_ALREADY_OPEN** - Attempted to open a cursor that was previously opened.
9. **NOT_LOGGED_ON** - A database call was made without being logged into Oracle.
10. **TRANSACTION_BACKED_OUT** - Usually raised when a remote portion of a transaction is rolled back.
11. **LOGIN_DENIED** - Login to Oracle failed because of invalid username and password.
12. **PROGRAM_ERROR** - Raised if PL/SQL encounters an internal problem.
13. **STORAGE_ERROR** - Raised if PL/SQL runs out of memory or if memory is corrupted.
14. **TIMEOUT_ON_RESOURCE** - Timeout occurred while Oracle was waiting for a resource.
15. **VALUE_ERROR** - Arithmetic, conversion, truncation, or constraint error occurred.
16. **OTHERS** - This is a catchall. If the error was not trapped in the previous exception traps, the error will be trapped by this statement

Probleme propuse spre rezolvare

1. Scrieți program în PL/SQL pentru a mări cu 40% salariul unui angajat al cărui cod este introdus de la tastatură. Să se trateze cazul în care nu există angajatul al cărui cod este specificat. Tratarea excepției se va face în secțiunea executabilă.
2. Scrieți program în PL/SQL pentru a afișa numele departamentului și numele managerului unui departament al cărui cod este dat ca valoare specificata. Să se trateze cazul în care nu există departamentul al cărui cod este specificat. Tratarea excepție se va face în secțiunea de tratare a erorilor.
3. Scrieți program în PL/SQL pentru a afișa numărul de departamente care au suma salariilor mai mare decât o valoare dată. Să se trateze cazul în care nici un departament nu îndeplinește această condiție.
4. Scrieți program în PL/SQL pentru a afișa numele departamentului care funcționează într-o anumită locație. Dacă rezultatul interogării nu returnează nici o linie, atunci să se trateze excepția și să se insereze în tabelul erori_pnu codul erorii -20001 cu mesajul “nu exista departamente in locatia data”. Dacă rezultatul interogării este o singură linie, atunci să se afișeze numele departamentului. Dacă rezultatul interogării presupune mai multe linii, atunci să se introducă în tabelul erori_pnu codul erorii -20002 cu mesajul “exista mai multe departamente in locatia data”.

Bibliografie web:

<https://www.w3resource.com/>

<https://www.bullraider.com/database/pl-sql/pl-sql-examples>

<https://www.oracletutorial.com/plsql-tutorial/plsql-exception/>

<https://docs.oracle.com/database/121/LNPLS/errors.htm#LNPLS007>