

# Limbajul PL/SQL- Tratarea erorilor

- Relansarea unei exceptii:

```
DECLARE
```

```
    eroare1 EXCEPTION;
```

```
BEGIN
```

```
...
```

```
    BEGIN
```

```
        ...
```

```
        IF conditie THEN
```

```
            RAISE eroare1;
```

```
        END IF;
```

```
    EXCEPTION
```

```
        WHEN eroare1 THEN
```

```
            -- prelucrarea erorii
```

```
            RAISE; relansarea exceptiei curente
```

```
        ... END;
```

# Limbajul PL/SQL- Tratarea erorilor

EXCEPTION

WHEN eroare1 THEN

--prelucrarea erorii altfel decit in subbloc.

END;

- Prelucrarea lansarii exceptiilor:

EXCEPTION

WHEN eroare1 THEN

    sir1

WHEN eroare2 THEN

    sir2

...

WHEN OTHERS THEN

    sir

# Limbajul PL/SQL- Tratarea erorilor

- EXCEPTION

WHEN eroare1 OR eroare2 OR ... OR VALUE\_ERROR THEN ...

Cind exceptia este ridicata in FOR cursorul este inchis.

- Exceptiile din declaratii:

DECLARE

constanta1 CONSTANT NUMBER(2):=200;

BEGIN

...

EXCEPTION

WHEN OTHERS THEN ...-- nu poate capta exceptia

- Exceptiile ce apar in handleri

EXCEPTION

WHEN INVALID\_NUMBER THEN

INSERT INTO ...

WHEN DUP\_VAL\_ON\_INDEX THEN...--nu poate fi captata exceptia

# Limbajul PL/SQL- Tratarea erorilor

- GOTO nu poate face saltul spre un handler si nici din handler spre blocul curent. Totusi GOTO poate face salt de la un handler intr-un bloc superior.
- Folosirea lui SQLCODE si SQLERRM.

Pentru exceptiile definite de utilizator, SQLCODE este +1 si SQLERRM da: User-Defined Exception, daca nu folosim pragma EXCEPTION\_INIT.

Daca nicio exceptie nu apare, SQLCODE este 0 si SQLERRM da ORA-0000:normal, successful completion.

SQLERRM(cod\_eroare)—returneaza mesajul erorii.

cod\_eroare>0 –User\_Defined Exception

- Nu putem folosi SQLCODE , SQLERRM direct in instructiuni SQL.

```
DECLARE cod_er NUMBER; mesaj_er VARCHAR2(100);
```

```
BEGIN ... EXCEPTION WHEN OTHERS THEN
```

```
cod_er:=SQLCODE; mesaj_er :=SQLERRM;
```

# Limbajul PL/SQL- Tratarea erorilor

- Tehnici de crestere a flexibilitatii

1. Continuarea dupa ce apare exceptia.

```
DECLARE
```

```
    coeficient NUMBER(3,1);
```

```
BEGIN
```

```
DELETE FROM stare WHERE symbol='XYZT';
```

```
SELECT pret/NVL(cistig,0) INTO coeficient FROM stocuri  
    WHERE symbol='XYZT';
```

```
INSERT INTO stare(symbol, coeficient) VALUES  
( 'XYZT',coeficient);
```

```
EXCEPTION
```

```
    WHEN ZERO_DIVIDE THEN ...
```

---

```
DECLARE
```

```
    coeficient NUMBER(3,1);    BEGIN
```

# Limbajul PL/SQL- Tratarea erorilor

```
DELETE FROM stare WHERE symbol = 'XYZT';  
  
BEGIN  
  SELECT pret/NVL(cistig,0) INTO coeficient FROM stocuri  
        WHERE symbol = 'XYZT';  
  
  EXCEPTION  
    WHEN ZERO_DIVIDE THEN  
      coeficient:=0;  
  
  END;  
  
  INSERT INTO stare(symbol,coeficient) VALUES ('XYZT',  
        coeficient);  
  
  EXCEPTION  
  
  ...
```

# Limbajul PL/SQL- Tratarea erorilor

- Reincercarea unei tranzactii:

```
DECLARE
```

```
nume CHAR(20); ans1 CHAR(5); ans2 CHAR(5); ans3 CHAR(5);
```

```
I NUMBER :=1;
```

```
BEGIN
```

```
  LOOP
```

```
    BEGIN
```

```
      SAVEPOINT start_transaction;
```

```
      DELETE FROM result WHERE raspuns='NO';
```

```
      INSERT INTO result VALUES (nume,ans1,ans2,ans3);
```

```
      COMMIT;
```

```
      EXIT;
```

```
    EXCEPTION
```

```
      WHEN DUP_VAL_ON_INDEX THEN ROLLBACK TO  
start_transaction;
```

# Limbajul PL/SQL- Tratarea erorilor

```
nume:=nume||TO_CHAR(I);
```

```
I:=I+1;
```

```
END;
```

```
END LOOP;
```

```
END;
```

- Folosirea variabilelor locator.

```
BEGIN
```

```
SELECT...
```

```
SELECT...
```

```
SELECT ...
```

```
...
```

```
EXCEPTION
```

```
WHEN NO_DATA_FOUND THEN ...
```

```
END;
```

# Limbajul PL/SQL- Tratarea erorilor

```
DECLARE
```

```
I INTEGER :=1;
```

```
BEGIN
```

```
    SELECT ...
```

```
    I:=2;
```

```
    SELECT ...
```

```
    I:=3;
```

```
    SELECT ...
```

```
    ...
```

```
EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
```

```
        INSERT INTO errors VALUES('Eroare in instructiunea:'||I);
```

```
        ...
```

```
END;
```