

Výjimky

Ošetřování chyb

- V PL/SQL se mohou vyskytnout chyby:
 - **Syntaktické** – objeví se už při kompilaci,
 - **logické** – projeví se až za běhu funkce/procedury.
- Nejčastěji se vyskytují tyto vestavěné výjimky:
 - **DUP_VAL_ON_INDEX** – výskyt duplicitní hodnoty ve sloupci, kde jsou povoleny jen jedinečné hodnoty,
 - **INVALID_NUMBER** – neplatné číslo, data nelze převést na číslo,
 - **NO_DATA_FOUND** – nebyly nalezeny žádné záznamy,
 - **TOO_MANY_ROWS** – dotaz vrátil neočekávaně více než jeden záznam,
 - **VALUE_ERROR** – problém s matematickou funkcí, chybný argument,
 - **ZERO_DIVIDE** – chyba dělení nulou.

Všechny: <https://docs.oracle.com/database/121/LNPLS/errors.htm#LNPLS00703>

Ošetřování chyb

- Syntaxe ošetřování výjimek:

EXCEPTION

```
WHEN <název výjimky1> THEN <příkazy1>;  
[WHEN <název výjimky2> THEN <příkazy2>; ...]  
[WHEN OTHERS THEN <příkazy3>];
```

END;

- Při ošetřování výjimky můžeme použít systémové funkce **SQLCODE** (vrací kód chyby), **SQLERRM** (vrací textový popis chyby). Pro vlastní výjimky je **SQLCODE** rovno 1 a **SQLERRM** vrací text User-Defined Exception.
- Syntaxe vyvolání vestavěné nebo vlastní výjimky:

```
RAISE <název výjimky>;
```

Příklad: **RAISE** NO_DATA_FOUND;

Ošetřování chyb – příklad

DECLARE

v_vysledek **NUMBER**(9,2);

BEGIN

v_vysledek := 5/0;

EXCEPTION

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Nastala chyba.');

DBMS_OUTPUT.PUT_LINE('Kód chyby: ' || SQLCODE);

DBMS_OUTPUT.PUT_LINE('Popis chyby: ' || SQLERRM);

END;

Poznámky

- Pozor na to, že blok EXCEPTION se vykoná pouze, když bude výjimka vyhozena v bloku BEGIN ... EXCEPTION:

DECLARE

```
v_vysledek NUMBER(9,2) := 5/0;
```

BEGIN

```
dbms_output.put_line('test exception');
```

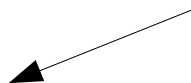
EXCEPTION

WHEN OTHERS THEN

```
dbms_output.put_line('Nastala chyba.');
```

END;

Tento kód se nevykoná!!!!



- Ze stejných důvodů je nutné dávat pozor na chyby v kódu, který se nachází v EXCEPTION ... END bloku.

Použití vlastních výjimek

- Deklarace, vyvolání a ošetření vlastní výjimky:

DECLARE

<název výjimky> **EXCEPTION**;

BEGIN

<příkazy>;

-- vyvolání výjimky při nějakém chybovém stavu

-- typicky při (ne)splnění nějaké podmínky:

IF <podmínka> **THEN**

RAISE <název výjimky>;

END IF;

EXCEPTION

-- při vyvolání výjimky se řízení přesune hned

-- do sekce EXCEPTION

WHEN <název výjimky> **THEN** <příkazy>;

END;

RAISE_APPLICATION_ERROR

- Vyvolat výjimku (tzv. aplikační výjimku) lze také pomocí procedury DBMS_STANDARD.RAISE_APPLICATION_ERROR(chybový kód, 'hlášení'). **Lze používat volné chybové kódy -20000 až -20999, které pak obdrží i programové API** volající uloženou proceduru/funkci (v objektu výjimky). Výjimku lze zachytit ve větvi OTHERS, případně pomocí konstrukce PRAGMA EXCEPTION_INIT namapovat chybový kód na vlastní jméno výjimky.
- RAISE_APPLICATION_ERROR se používá také v sekci EXCEPTION, když chceme zachycenou výjimku nechat vyvolat ven z procedury/funkce s určitým chybovým kódem a hlášením.

RAISE_APPLICATION_ERROR

- Příklad:

```
DECLARE
    v_vysledek NUMBER(9, 2);
BEGIN
    v_vysledek := 40/0;
EXCEPTION
    WHEN ZERO_DIVIDE THEN
        dbms_standard.raise_application_error(
            'Chyba dělení nulou', -20000, true);
END;
```

Parametry: Chybové hlášení, chybový kód,
3. parametr - nepovinný příznak keep error stack
true znamená, že bude předáván výpis zásobníku
obsahující informace o původu vzniku chyby.
Námi definovaný kód chyby bude mít k dispozici
aplikace, která volá PL/SQL kód, v objektu výjimky.