

# Oracle PL/SQL

Paweł Rajba

[pawel@ii.uni.wroc.pl](mailto:pawel@ii.uni.wroc.pl)

<http://www.kursy24.eu/>

# Zawartość modułu 4

---

- Wyjątki PL/SQL
- Mechanizmy dotyczące błędów
- Typy wyjątków
- Obsługa wyjątków PL/SQL
  - Obsługa błędów predefiniowanych, użytkownika i niepredefiniowanych
- Wyjątki w sekcji deklaracji i wyjątków
- Zasięg wyjątków PL/SQL
- Niestandardowe komunikaty błędów

# Wyjątki PL/SQL

---

- Wyjątek to błąd podczas wykonania
- Etapy obsługi wyjątków
  - Deklaracja (jawna lub niejawna)
  - Zgłoszenie wyjątku (jawne lub niejawne)
  - Obsługa wyjątku (jawna lub niejawna)
- Każdy wyjątek ma
  - Typ błędu (kompilacji lub wykonanie)
  - Kod błędu (liczba wskazująca numer)
  - Tekst błędu (opisuje błąd, zawiera również kod)

# Wyjątki PL/SQL

---

- Typ błędu
  - PLS – podczas kompilacji
  - ORA – podczas wykonania
    - może się zdarzyć, błąd PLS będzie osadzone w ORA (np. przy uruchamianiu dynamicznego kodu)
- Kod błędu
  - zwykle liczba ujemna
- Tekst błędu
  - Max 512 bajtów

# Mechanizmy dotyczące błędów

---

- Deklaracje/instrukcje
  - deklaracja EXCEPTION
    - deklaruje wyjątek użytkownika
  - instrukcja RAISE
    - służy do zgłaszania wyjątków
  - dyrektywa PRAGMA EXCEPTION\_INIT
    - służy do kojarzenia błędu Oracle z wyjątkiem użytkownika
  - klauzula SAVE\_EXCEPTIONS
    - pozwala kontynuować przetwarzanie po niepowodzeniu na poziomie wiersza podczas operacji masowego wiązania

# Mechanizmy dotyczące błędów

---

- Funkcje
  - SQLCODE – zwraca kod błędu
  - SQLERRM – zwraca komunikat błędu
    - komunikat zawsze zaczyna się kodem błędu
- Procedura
  - RAISE\_APPLICATION\_ERROR
    - pozwala definiować niestandardowe komunikaty błędów
    - pozwala zatrzymywać program

# Typy wyjątków

---

- Wyjątki predefiniowane
- Niepredefiniowane błędy Oracle
- Wyjątki zdefiniowane przez użytkowników
- Niestandardowe komunikaty błędów

# Obsługa wyjątków PL/SQL

---

- Deklarujemy w klauzuli exception:
  - EXCEPTION
    - WHEN exceptionA1 [OR exceptionA2...] THEN  
... /\* kod obsługi \*/
    - WHEN exceptionB1 [OR exceptionB2...] THEN  
... /\* kod obsługi \*/
    - ...
    - WHEN OTHERS THEN  
... /\* kod obsługi \*/



# Obsługa wyjątków PL/SQL

---

- Kilka uwag
  - W przypadku zgłoszenia wyjątku, szukana jest klauzula w najbliższym bloku, potem w bloku wyżej, itd. aż do klauzuli w bloku zewnętrznym (obejmującym)
  - WHEN OTHERS pełni rolę sita łapiącego nieobsłużone wyjątki
  - Po obsłudze wyjątku program może się kontynuować lub zatrzymać
    - Można tym sterować zagnieżdżaniem bloków

# Obsługa błędów predefiniowanych

---

- Błędy często się pojawiające zostały predefiniowane
  - Dostały typ, numer i tekst komunikatu
- Wyjątków predefiniowanych nie trzeba ani deklarować ani zgłaszać
  - można jednak je zgłaszać poprzez RAISE
- `SQLCODE` i `SQLERRM` dają możliwość logowania błędów
  - jedynym błędem dodatnim kodzie (+100) to `NO_DATA_FOUND`

# Obsługa błędów użytkownika

---

- Związane tylko z daną aplikacją PL/SQL
- Nie są skojarzone z żadnym numerem błędu ani komunikatem
- Deklaracja
  - nazwa EXCEPTION
- Zgłaszania
  - RAISE nazwa
- Obsługa
  - WHEN nazwa THEN ...

# Obsługa błędów użytkownika

---

- Kilka uwag
  - Wyjątek użytkownika jest zgłaszany tylko poprzez jawne użycie RAISE
  - Jeśli zadeklarujemy wyjątek o nazwie takiej jak wyjątek predefiniowany, to
    - wyjątek predefiniowany jest przykrywane przez ten użytkownika
    - dostęp do predefiniowanego jest poprzez konstrukcję
      - STANDARD.nazwa\_wyjatku
  - Ogólnie jednak nie zalecane tworzenie ww. wyjątków

# Obsługa błędów niepredefiniowanych

---

- Mamy dwa sposoby obsłużenia takich błędów:
  - za pomocą klazuli WHEN OTHERS w połączeniu z funkcja SQLCODE i SQLERRM
    - i to już znamy
  - kojarzymy wyjątek użytkownika z odpowiednim kodem błędu i obsługujemy poprzez mechanizm obsługi błędów użytkownika
- Przykłady takich błędów
  - naruszenie więzów klucza głównego lub obcego
  - naruszenie więzów kontrolnych kolumny (check)

# Obsługa błędów niepredefiniowanych

---

- Szablon obsługi jest następujący
  - DECLARE
    - deklaracje wyjątku użytkownika
    - wyjatek EXCEPTION;
    - kojarzenie błędu z wyjątkiem użytkownika
    - PRAGMA EXCEPTION\_INIT(wyjatek, kod\_bledu);
  - BEGIN
  - ...
  - EXCEPTION
  - WHEN wyjatek THEN ...
  - END;

# Obsługa błędów niepredefiniowanych

---

- Kilka uwag
  - Skojarzony wyjątek użytkownika zgłaszany jest automatycznie (o to w końcu chodziło)
    - chociaż można go zgłosić ręcznie
  - PRAGMA musi być zadeklarowana w bloku deklaracji wyjątku użytkownika (najlepiej zadeklarować jedno pod drugim)
  - Można skojarzyć więcej niż jeden wyjątek użytkownika z tym samym numerem błędu
- Przykład
  - `wyjatki-niepredefiniowane.txt`

# Wyjątki w sekcji deklaracji i wyjątków

---

- Błędy pojawiające się w sekcji deklaracji i wyjątków są obsługiwane
- Obsługa jest w klauzuli exception, w której znajduje się blok deklaracji lub klauzulu wyjątków
- Przykład
  - wyjatki-deklaracje.txt
  - wyjatki-wyjatki.txt



# Ponowne zgłaszanie wyjątku

---

- Polega na przekazaniu obsługi wyjątku do klauzuli exception bloku zawierającego dany blok
- Przykład
  - wyjatki-ponowne.txt

# Zasięg wyjątków PL/SQL

---

- Wyjątki predefiniowane są widoczne w całym kodzie
- Wyjątki użytkownika są widoczne w sekcji, w której są zadeklarowane i sekcjach zagnieżdżonych
- Przykłady
  - `wyjatki-zasieg.txt`

# Niestandardowe komunikaty błędów

---

- Do tej pory mieliśmy
  - Błędy Oracle, które miały zdefiniowany kod i opis
  - Wyjątki użytkownika
- Oracle daje możliwość tworzenia nowych błędów z własnym kodem i opisem

# Niestandardowe komunikaty błędów

---

- Do definiowania używamy procedury `RAISE_APPLICATION_ERROR( err_no number, err_msg varchar2, keep_errors boolean)`
  - `err_no` – numer z przedziału [-20000, -20999]
  - `err_msg` – napis o długości <512
  - `keep_errors`
    - `true` – nowy błąd zostanie dopisany do listy już zdefiniowanych błędów
    - `false` – nowy błąd zastąpi listę już zgłoszonych błędów

# Niestandardowe komunikaty błędów

---

- Kilka uwag
  - Mechanizm pozwala stworzyć zestaw komunikatów usztyty pod daną aplikację (np. błąd 'Kwota zbliża się do 2 progu podatkowego')
  - Szczególnie przydatne przy tworzeniu procedur (będziemy to jeszcze omawiać)
    - w bloku wywołującym procedurę, funkcje SQLCODE i SQLERRM zwrócą zarejestrowany kod i komunikat błędu

# Kilka uwag na koniec

---

- W bloku obejmującym obowiązkowo powinna być klauzule exception wraz z when others
- Każdy błąd warto logować w odpowiedniej tabeli
- Warto tworzyć zestaw niestandardowych komunikatów błędów dla konkretnej aplikacji
- Nie należy nadpisywać wyjątkami użytkowników wyjątków predefiniowanych
- Obsługę błędów należy oddzielać od kodu
  - Korzystajmy z mechanizmu wyjątków