

Laboratorium 5

Bazy Danych 2

mgr. inż. Aleksander Wojtowicz

Sekwencja to obiekt w bazie danych, który generuje unikalne liczby w określonym porządku.

Sekwencje są często używane do tworzenia unikalnych identyfikatorów (*często nazywanych "kluczami głównymi"*) dla rekordów w tabeli.

Sekwencja składa się z następujących elementów:

- **Początkowa wartość START WITH**

Pierwsza wartość, którą sekwencja wygeneruje.

- **Wartość minimalna MINVALUE**

Najmniejsza wartość, którą sekwencja może wygenerować.

- **Wartość maksymalna MAXVALUE**

Największa wartość, którą sekwencja może wygenerować.

- **Wartość przyrostu INCREMENT BY**

Wartość, o którą sekwencja zwiększa się za każdym razem, gdy jest generowana nowa wartość.

- **Wartość bufora CACHE**

Liczba wartości sekwencji, które są przechowywane w pamięci dla szybkiego dostępu.

Sekwencje są często używane do generowania unikalnych identyfikatorów dla rekordów w tabeli.

Na przykład, gdy dodajemy nowy rekord do tabeli, sekwencja może automatycznie generować unikalny identyfikator dla tego rekordu.

To jest szczególnie przydatne w przypadku dużych baz danych, gdzie ręczne zarządzanie identyfikatorami byłoby trudne i czasochłonne.

Sekwencje mogą być również używane do generowania unikalnych numerów dla różnych transakcji lub operacji.

Na przykład, sekwencja może być używana do generowania unikalnego numeru zamówienia dla każdej nowej transakcji.

To pomaga w śledzeniu i zarządzaniu transakcjami w systemie.

Ponieważ sekwencje generują unikalne wartości, pomagają one zapewnić spójność i integralność danych.

Dzięki temu można uniknąć konfliktów i błędów związanych z duplikatami.

Przykład tworzenia sekwencji w SQL:

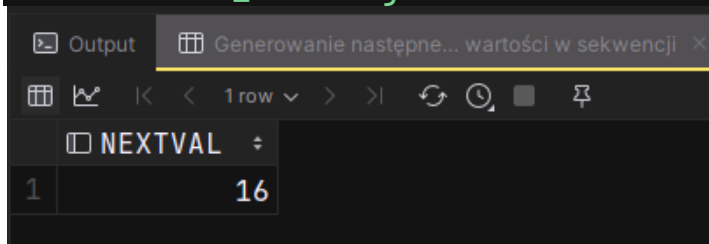
```
CREATE SEQUENCE nazwa_sekwencji  
START WITH 1  
INCREMENT BY 1  
MINVALUE 1  
MAXVALUE 10000  
CACHE 10;
```

W tym przykładzie:

- **nazwa_sekwencji** to nazwa, którą chcemy przypisać do sekwencji.
- **START WITH 1** określa, że pierwsza wygenerowana wartość będzie wynosić 1.
- **INCREMENT BY 1** oznacza, że każda kolejna wartość będzie większa o 1 od poprzedniej.
- **MINVALUE 1** i **MAXVALUE 10000** określają, że sekwencja będzie generować wartości od 1 do 10000.
- **CACHE 10** oznacza, że baza danych będzie przechowywać 10 wartości sekwencji w pamięci podręcznej dla szybszego dostępu.

Przykłady użycia sekwencji:

```
-- Generowanie następnej wartości w sekwencji  
SELECT nazwa_sekwencji.NEXTVAL FROM dual;  
-- Pobieranie bieżącej wartości sekwencji  
SELECT nazwa_sekwencji.CURRVAL FROM dual;
```



The screenshot shows a SQL Developer window with the title 'Generowanie następne... wartości w sekwencji'. The output table has one column named 'NEXTVAL' and one row with the value '16'.

NEXTVAL
16

Wynik po wielokrotny wywołaniu `nazwa_sekwencji.NEXTVAL`

W powyższym przykładzie:

- `nazwa_sekwencji` to nazwa sekwencji, której chcemy użyć.
- `NEXTVAL` jest używane do generowania następnej wartości w sekwencji. Za każdym razem, gdy używamy `NEXTVAL`, sekwencja jest zwiększana o wartość określoną w `INCREMENT BY`.
- `CURRVAL` jest używane do pobrania bieżącej wartości sekwencji. To jest ostatnia wartość, która została wygenerowana przez `NEXTVAL` dla tej sekwencji.

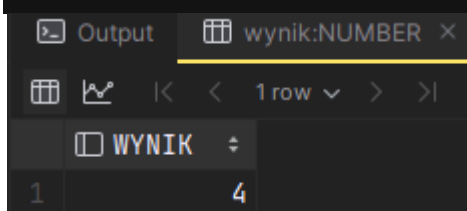
Czym jest `dual` z powyższego przykładu?

DUAL to specjalna jednowierszowa i jednokolumnowa tabela. Jest dostępna dla każdego użytkownika bazy danych.

Tabela **DUAL** jest używana do wykonywania obliczeń, które nie wymagają faktycznych danych z bazy danych.

Na przykład, jeśli chcemy obliczyć wartość wyrażenia $2+2$ za pomocą SQL w Oracle:

```
SELECT 2 + 2 AS wynik FROM dual;
```



The screenshot shows a SQL Developer window with the title 'wynik:NUMBER'. The output table has one column named 'WYNIK' and one row with the value '4'.

WYNIK
4

DUAL działa jako tabela, z której możemy "wybrać" wynik naszego wyrażenia.

Pamiętajmy, że nie wszystkie systemy zarządzania bazami danych mają tabelę **DUAL**.

Na przykład, w MySQL możemy po prostu napisać `SELECT 2 + 2;` bez używania **DUAL**.

Sekwencje są bezpieczne dla wielu użytkowników, co oznacza, że wiele sesji może jednocześnie generować wartości z tej samej sekwencji bez konfliktów.

To jest możliwe dzięki mechanizmom blokowania i synchronizacji używanym przez systemy zarządzania bazami danych.

Wartości generowane przez sekwencję nie są gwarantowane, że będą w kolejności, jeśli są generowane przez wiele sesji.

Na przykład, jeśli dwie sesje jednocześnie wywołują NEXTVAL na tej samej sekwencji, jedna sesja może otrzymać niższą wartość niż druga, nawet jeśli wywołała NEXTVAL później.

Wartości sekwencji mogą być również generowane w kolejności, nawet jeśli transakcja jest wycofywana. To oznacza, że jeśli transakcja jest wycofywana po wygenerowaniu wartości sekwencji, ta wartość **nie jest "odzyskiwana"**. Jeśli następna transakcja wywoła NEXTVAL, otrzyma następną wartość w sekwencji, a nie wartość, która została "stracona" podczas wycofania.

Utworzenie sekwencji i użycie jej do wstawiania danych do tabeli

Tworzymy sekwencję o nazwie seq_id_pracownika z wartością początkową 4 i wartością przyrostu 1.

Używamy tej sekwencji do wstawienia nowego rekordu do tabeli pracownicy, gdzie id_pracownika jest generowany przez sekwencję.

```
CREATE SEQUENCE seq_id_pracownika
  START WITH 4
  INCREMENT BY 1;

INSERT INTO pracownicy(id_pracownika, imie, nazwisko, pensja)
VALUES(seq_id_pracownika.NEXTVAL, 'Marek', 'Zieliński', 65000);
```

Modyfikacja istniejącej sekwencji i obserwacja zmian

Zmienimy wartość przyrostu sekwencji seq_id_pracownika na 2.

Wstaw kolejny rekord do tabeli pracownicy i sprawdźmy, jak zmieniła się wartość id_pracownika.

```
ALTER SEQUENCE seq_id_pracownika  
    INCREMENT BY 2;  
  
INSERT INTO pracownicy(id_pracownika, imie, nazwisko, pensja)  
VALUES(seq_id_pracownika.NEXTVAL, 'Ewa', 'Kowalczyk', 70000);
```

Usunięcie sekwencji i próba jej użycia

Usuńmy sekwencję seq_id_pracownika.

Spróbujmy wstawić kolejny rekord do tabeli pracownicy używając usuniętej sekwencji.

Powinien zostać zwrócony błąd.

```
DROP SEQUENCE seq_id_pracownika;  
  
INSERT INTO pracownicy(id_pracownika, imie, nazwisko, pensja)  
VALUES(seq_id_pracownika.NEXTVAL, 'Krzysztof', 'Jankowski',  
75000);
```

PODSUMOWANIE

- **Sekwencje w PL/SQL** są obiektami bazy danych, które generują unikalne liczby w określonym porządku. Są one często używane do tworzenia unikalnych identyfikatorów dla rekordów w tabeli.
- **Tworzymy sekwencję** za pomocą polecenia `CREATE SEQUENCE`, gdzie można określić różne atrybuty, takie jak wartość początkową, wartość minimalną, wartość maksymalną, wartość przyrostu i wartość bufora.
- **Używanie sekwencji:** Sekwencje są używane za pomocą `NEXTVAL` i `CURRVAL`. `NEXTVAL` jest używane do generowania następnej wartości w sekwencji, a `CURRVAL` jest używane do pobrania bieżącej wartości sekwencji.
- **Modyfikowanie sekwencji** za pomocą polecenia `ALTER SEQUENCE`, gdzie można zmienić różne atrybuty sekwencji.
- **Usuwanie sekwencji** za pomocą polecenia `DROP SEQUENCE`.
Usunięcie sekwencji jest trwałe i nie można go cofnąć.
- **Sekwencje a równoczesność** - Sekwencje są bezpieczne dla wielu użytkowników, co oznacza, że wiele sesji może jednocześnie generować wartości z tej samej sekwencji bez konfliktów.

Zadania:

1. Utwórz sekwencję o nazwie `seq_klient_id` z wartością początkową 100 i wartością przyrostu 10. Następnie wygeneruj pierwsze 5 wartości tej sekwencji.
2. Zmodyfikuj sekwencję `seq_klient_id`, tak aby wartość przyrostu wynosiła 5. Wygeneruj następne 5 wartości tej sekwencji i zobacz, jak zmieniły się wartości.
3. Utwórz tabelę `klienci` z kolumnami `id_klienta`, `imie`, `nazwisko` i `email`. Użyj sekwencji `seq_klient_id` do generowania wartości dla `id_klienta`.
4. Wstaw 5 rekordów do tabeli `klienci`, używając sekwencji `seq_klient_id` do generowania wartości `id_klienta`.
5. Usuń sekwencję `seq_klient_id`. Spróbuj wstawić kolejny rekord do tabeli `klienci` używając usuniętej sekwencji. Jaki błąd otrzymujesz?