

Bazy danych
Laboratorium 9
Info

Sekwencja jest obiektem podającym kolejne wartości wg ustalonych kryteriów. Sekwencje stosujemy często w celu tworzenia kluczy głównych tabel.

```
CREATE SEQUENCE nazwa_sekwencji  
    minvalue 0  
    maxvalue 9999  
    start with 0  
    cycle  
    increment by 1;
```

gdzie:

- *minvalue* - minimalna wartość jaką może przybrać sekwencja,
- *Maxvalue* - maksymalna wartość jaką może przybrać sekwencja,
- *start with* - określa wartość od jakiej ma się rozpoczynać sekwencja,
- *increment by* - określa wartość o jaką ma się zmieniać aktualny stan sekwencji po każdym pobraniu danych (jako parametr tego polecenia możemy również podać liczbę ujemną, wartość sekwencji będzie wtedy maleć),
- *cycle / nocycle* - parametr ten określa czy sekwencja może się „przekreślić” i rozpocząć naliczanie od początku.

Inne możliwe opcje:

- *Cache / nocache* - klauzula CACHE włącza wykonywanie pre-alokacji numerów sekwencji i przechowywanie ich w pamięci, co skutkuje zwiększeniem szybkości generacji kolejnych liczb. Klauzula NOCACHE wyłącza tę możliwość. Domyślnie przyjmowane jest CACHE 20. Wartość podana w CACHE musi być mniejsza niż MAXVALUE – MINVALUE
- *Order / noorder* - klauzula ORDER gwarantuje, że kolejne liczby będą generowane w porządku jakim otrzymane zostały przez system polecenia ich generacji. Klauzula NOORDER wyłącza tę własność.

Pobieranie wartości z sekwencji

```
select nazwa_sekwencji.currval from dual;  
select nazwa_sekwencji.nextval from dual;  
insert into nazwa_tabeli values (nazwa_sekwencji.nextval, 'Nowak', 'Anna', ...);
```

gdzie

- *Nextval* - służy do pobierania następnej wartości z sekwencji,
- *Currval* - Służy do pobierania aktualnej wartości sekwencji.

Usuwanie sekwencji

```
DROP SEQUENCE nazwa_sekwencji;
```

Modyfikacja sekwencji

```
ALTER SEQUENCE nazwa_sekwencji nazwa_parametru {wartość}
```

np.

```
alter sequence moja_sekwencja increment by 10;
```

Widok (perspektywa) jest predefiniowanym, zapisanym po stronie serwera zapytaniem, którego wynik może być wielokrotnie odczytywany

```
CREATE [OR REPLACE] VIEW Nazwa_widoku  
    [(lista_kolumn)]  
AS SELECT zapytanie;
```

- Opcjonalny parametr *lista kolumn* definiuje nazwy kolumn dla perspektywy. Lista kolumn tworzonej perspektywy musi zawierać dokładnie tyle pozycji, ile kolumn zwraca instrukcja SELECT. Jeżeli w klauzuli SELECT użyliśmy wyrażeń, należy jawnie określić nazwy kolumn widoku.
- Zapytanie jest dowolną — z kilkoma wyjątkami — poprawnie sformułowaną instrukcją SELECT języka SQL.
- Jeżeli chcemy zmienić wcześniej utworzony widok, należy po słowie kluczowym CREATE użyć opcji OR REPLACE.

Modyfikowanie danych poprzez widoki

Ponieważ widok nie przechowuje kopii danych, ale jest jedynie „pryzmatem”, przez który odczytujemy dane zapisane w tabelach, modyfikacja danych widoku jest w rzeczywistości modyfikacją danych zapisanych w tabelach. Widoki, w odniesieniu do których mogą być wykonywane instrukcje INSERT, DELETE i UPDATE, nazywane są widokami modyfikowalnymi.

Modyfikowanie danych jest możliwe, o ile każdy pojedynczy wiersz widoku odpowiada dokładnie jednemu wierszowi tabeli (widok jest powiązany z tabelami bazowymi związkiem jedno-jednoznaczny). Konsekwencją tego ograniczenia jest to, że widoki, przy których definicji użyto:

- funkcji grupujących w klauzuli SELECT,
- operatora DISTINCT,
- klauzul GROUP BY lub HAVING,
- wyrażeń matematycznych na liście argumentów klauzuli SELECT,

nie mogą być wykorzystane do modyfikowania danych w tabeli.

Warto do wszystkich modyfikowalnych widoków dodać klauzulę **WITH CHECK OPTION** w celu poprawienia spójności danych (przy wykonywaniu INSERT i UPDATE następuje sprawdzenie, czy wstawiany bądź modyfikowany wiersz spełnia warunek określony w klauzuli WHERE)

WITH READ ONLY – opcja, w ramach której uniemożliwia się dokonywanie zmian poprzez perspektywę w tabelach „źródłowych”

Modyfikowanie widoków

ALTER VIEW nazwa_widoku AS Select ... ;

Usuwanie widoków

DROP VIEW nazwa_widoku;

Ponieważ usunięcie tabeli bazowej nie spowoduje automatycznego usunięcia powiązanych z nią widoków, z reguły instrukcje DROP TABLE i DROP VIEW wykonywane są łącznie.

Indeksy

CREATE INDEX nazwa_indeksu ON nazwa_tabeli(nazwa_kolumny);

CREATE INDEX nazwa_indeksu ON nazwa_tabeli(nazwa_kolumny1, nazwa_kolumny2);

Definiowanie porządku kolumn

Dość powszechną praktyką jest definiowanie w indeksie porządku, w jakim klucze będą posortowane. Służą do tego dwa słowa kluczowe *ASC* i *DESC*.

CREATE INDEX nazwa_indeksu ON nazwa_tabeli(nazwa_kolumny1 ASC, nazwa_kolumny2 DESC);

Czasami zachodzi taka potrzeba, aby zapewnić unikalność wartości w danej kolumnie. W takim przypadku należy na niej założyć indeks unikalny. Aby dało się założyć taki indeks, wartości w danej kolumnie (lub w wielu kolumnach, gdy zakładamy indeks na wielu kolumnach) muszą być różne. W zdecydowanej większości przypadków pola muszą być zdefiniowane jako *NOT NULL*, ponieważ dwie wartości NULL interpretowane są jako te same wartości.

CREATE UNIQUE INDEX nazwa_indeksu ON nazwa_tabeli(nazwa_kolumny);