



**Wydział Elektroniki
i Technik Informatycznych**

POLITECHNIKA WARSZAWSKA

Bazy Danych 1

edycja 21L

Laboratorium 7

Przebieg laboratorium

 Porównywanie wt. NULL

 Widoki

 Sekwencje

 Synonimy

 [VIEWS - Oracle docs](#)  [Sequences - Oracle docs](#)  [Synonyms - Oracle docs](#)

Porównywanie wt. NULL

- Jedynie operatory **IS NULL**, **IS NOT NULL** przy porównywaniu z wt. NULL zwracają wt. logiczne TRUE lub FALSE. **Zastosowanie pozostałych operatorów zwraca wt. logiczną UNKNOWN.**
- Przykład 1: założmy, że zmienna A ma wartość 10:
 - $A = \text{NULL} \rightarrow \text{UNKNOWN}$
 - $A > \text{NULL} \rightarrow \text{UNKNOWN}$
 - $A < \text{NULL} \rightarrow \text{UNKNOWN}$
 - $A \text{ IS NULL} \rightarrow \text{FALSE}$
 - $A \text{ IS NOT NULL} \rightarrow \text{TRUE}$
- Przykład 2: założmy, że zmienna A ma wartość NULL:
 - $A = \text{NULL} \rightarrow \text{UNKNOWN}$
 - $A \text{ IS NULL} \rightarrow \text{TRUE}$

Rezultaty operacji logicznych

	TRUE	FALSE	UNKNOWN
NOT	FALSE	TRUE	UNKNOWN

AND	TRUE	FALSE	UNKNOWN
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
UNKNOWN	UNKNOWN	FALSE	UNKNOWN

OR	TRUE	FALSE	UNKNOWN
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
UNKNOWN	TRUE	UNKNOWN	UNKNOWN

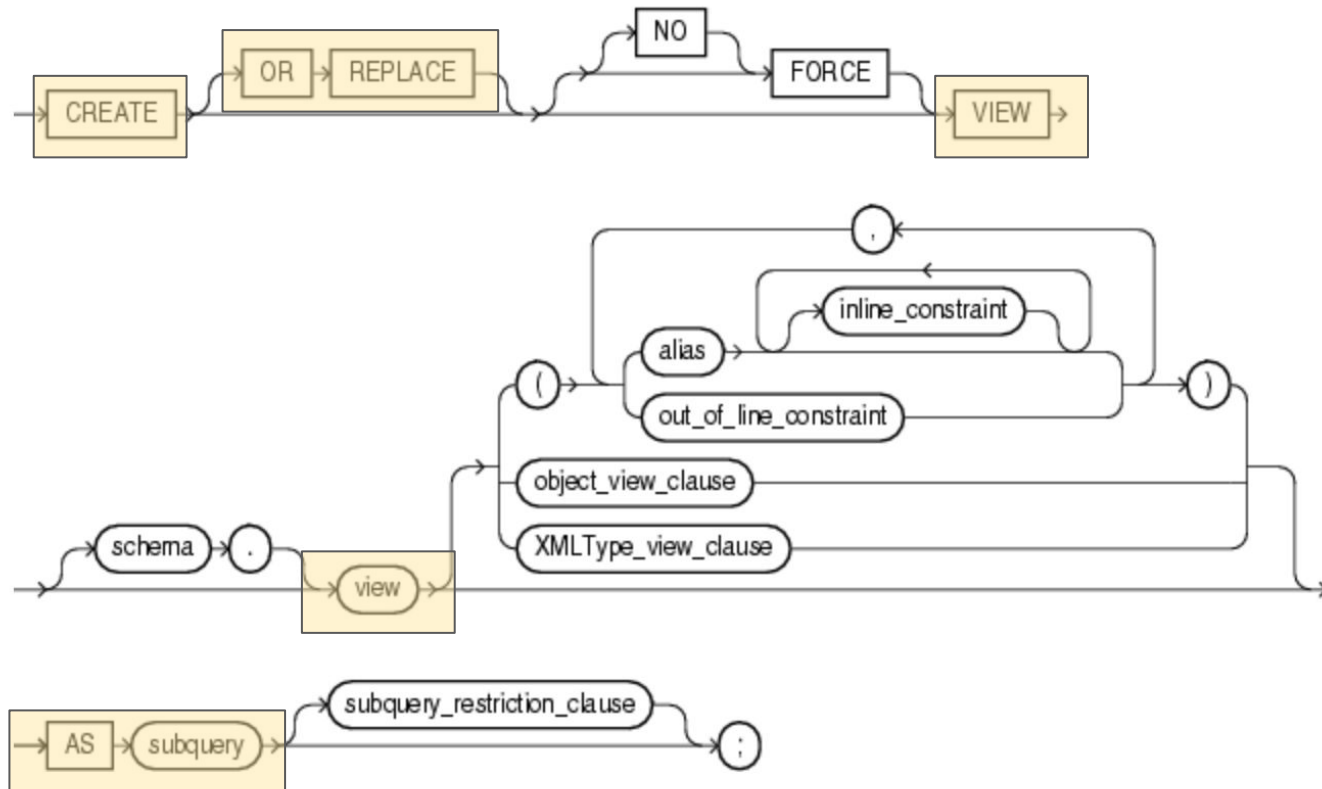
Widoki

- Widoki (perspektywy) są logicznymi reprezentacjami tabeli lub złączeń wielu tabel (tzw. tabel bazowych).
 - Tabele bazowe mogą być zarówno rzeczywistymi tabelami, jak i widokami.
- Dlaczego stosujemy widoki:
 - ze względów bezpieczeństwa (nie udostępniamy całego schematu bazy danych).
 - bo nie chcemy powtarzać skomplikowanych zapytań do BD wiele razy.
 - widoki (zmaterializowane) mogą skrócić czas oczekiwania na złożone zapytania zwracające wiele wierszy.

Widoki - dozwolone operacje

- Do widoków można zawsze wydawać zapytania SELECT.
- Za pomocą niektórych widoków można wprowadzać, modyfikować i usuwać dane w tabelach bazowych poleceniami INSERT, UPDATE i DELETE.
- Modyfikowanie danych za pomocą widoków nie jest możliwe w następujących przypadkach:
 - Widok jest zdefiniowany jako WITH READ ONLY.
 - Dla widoków zdefiniowany jako WITH CHECK OPTION możliwe jest modyfikowanie jedynie tych wierszy, które spełniają warunki klauzuli WHERE z definicji widoku.
 - Definicja widoku nie uwzględnia kolumn NOT NULL z tabeli bazowej.
 - Widok jest zdefiniowany na podstawie niektórych zapytań SELECT np. posiadającego klauzulę GROUP BY.
 - ...

Widoki - składnia



Widoki - przykład

- W definicji widoku powinno pojawić się zapytanie, które odwoła się to tabel lub innych widoków.
- Przykład: *Utwórz widok, który będzie zawierać imię, nazwisko, zarobki oraz nazwę zajmowanego stanowiska dla wszystkich pracowników.*

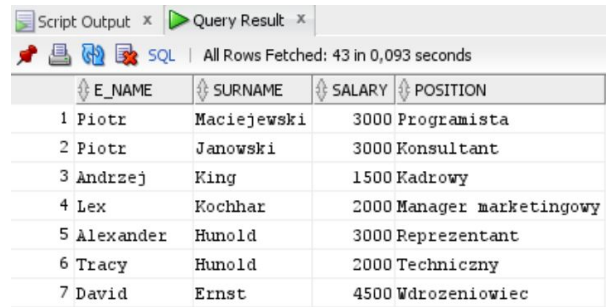
```
CREATE VIEW emp_pos_names  
AS
```

Utworzenie widoku

```
SELECT e.name as name, surname, salary, p.name as position  
FROM employees e JOIN positions p USING (position_id);
```

```
SELECT position, COUNT(*)  
FROM emp_pos_names  
WHERE salary > 2100  
GROUP BY position  
HAVING COUNT (*) >= 2  
ORDER BY 2 DESC;
```

Wydanie zapytania
z wykorzystaniem
widoku



Script Output x Query Result x

All Rows Fetched: 43 in 0,093 seconds

E_NAME	SURNAME	SALARY	POSITION
1 Piotr	Maciejewski	3000	Programista
2 Piotr	Janowski	3000	Konsultant
3 Andrzej	King	1500	Kadrowy
4 Lex	Kochhar	2000	Manager marketingowy
5 Alexander	Hunold	3000	Reprezentant
6 Tracy	Hunold	2000	Techniczny
7 David	Ernst	4500	Wdrozeniowiec

Widoki - modyfikowanie danych

→ Przykład: Widok `emp_view` przechowuje `id`, `imie`, `nazwisko` oraz `datę urodzenia` danego pracownika. Wykorzystaj ten widok do dodania nowego pracownika do tabeli `employees`.

```
CREATE VIEW emp_view AS
    SELECT employee_id, name, surname, birth_date
    FROM employees;
```

```
INSERT INTO emp_view VALUES (301, 'Piotr', 'Kowalski', SYSDATE);
UPDATE emp_view SET name = 'PIOTR' WHERE name LIKE 'Piotr';
```

Widoki - ograniczenie na modyfikowanie danych

- Oprócz tego, że można tworzyć widoki tylko “do odczytu”, to możliwe jest także definiowanie widoków, które pozwolą na modyfikowanie tych wierszy w tabeli bazowej, do których odwołuje się definicja widoku.
- Przykład: *Widok emp_view_names udostępnia kolumny employee_id, name, surname dla pracowników, których imiona rozpoczynają się od litery ‘P’, ‘R’, ‘M’:*

```
CREATE VIEW emp_view_names AS
  SELECT employee_id, name, surname, birth_date FROM employees
  WHERE name like 'P%' OR name like 'R%' OR name like 'M%'
  WITH CHECK OPTION;
```

- Ponieważ widok został stworzony z opcją **WITH CHECK OPTION**, to możliwe jest za jego pomocą dodawanie, modyfikowanie i usuwanie tylko tych wierszy tabeli Employees, które spełniają warunek:

```
WHERE name like 'P%' OR name like 'R%' OR name like 'M%'
```

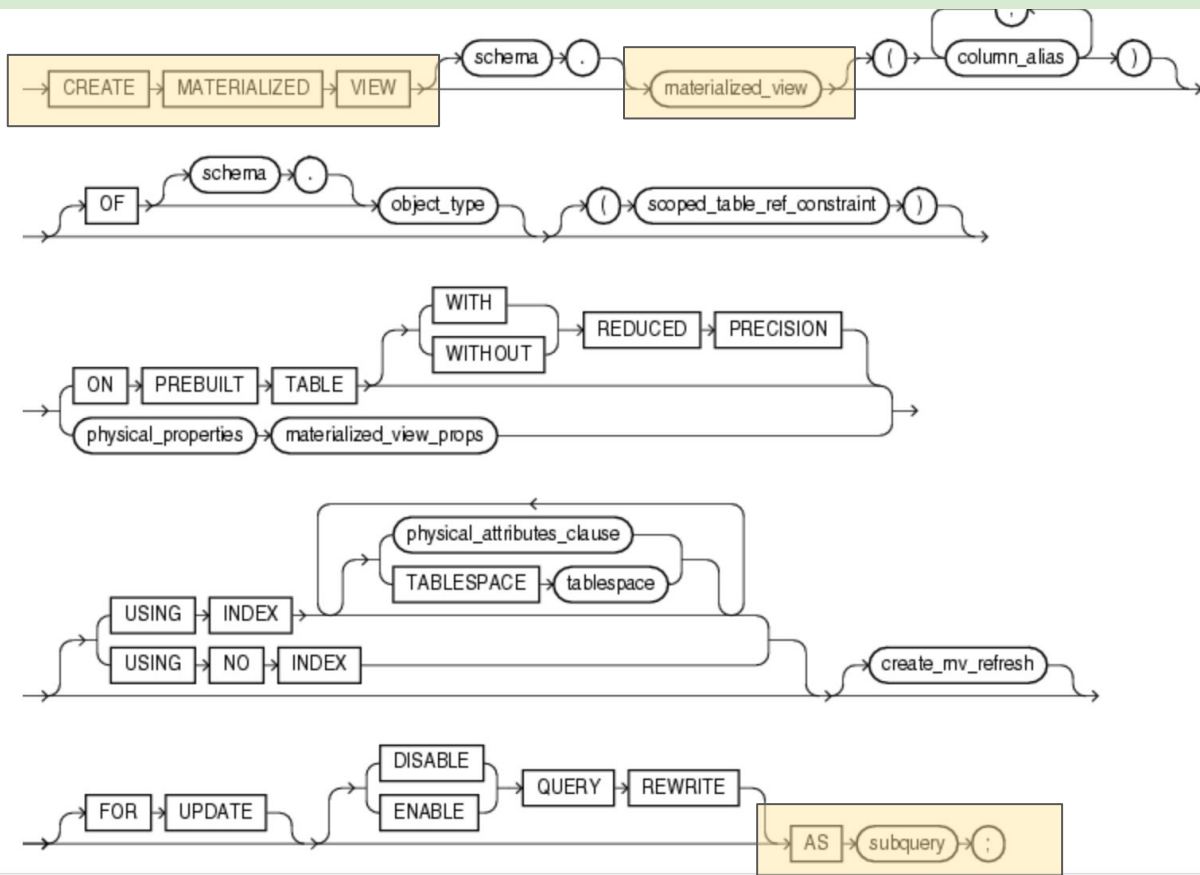
Widoki - ćwiczenia

1. Zdefiniuj widok, który będzie zwracać imię, nazwisko, zarobki, nazwę stanowiska, nazwę departamentu oraz imię i nazwisko managera dla wszystkich pracowników w tabeli employees.
2. Zdefiniuj widok typu WITH CHECK OPTION przechowujący id stanowisk (position_id) oraz nazwę, minimalne zarobki wszystkich stanowisk rozpoczynających się od litery 'P', 'K' lub 'M'. Następnie spróbuj zwiększyć minimalne zarobki dla stanowiska 'Rekruter' o 1000 i przeanalizuj komunikat błędu.
3. Wykonaj polecenia DROP VIEW, aby usunąć jeden z wcześniej utworzonych widoków.

Widoki zmaterializowane

- Rezultaty zapytań definiujących widoki w poprzednich przykładach nie są na stałe przechowywane w bazie danych. Oznacza to, że każdorazowe odwołanie do widoku powoduje na nowo wykonanie zapytania definiującego ten widok.
- W przypadku widoków zmaterializowanych rezultaty zapytań definiujących widoki są na stałe przechowywane w bazie danych.
- Należy odświeżać widoki zmaterializowane, aby zobaczyć aktualne dane w tabelach bazowych (automatycznie lub na żądanie).

Widoki zmaterializowane - składnia



Widoki zmaterializowane - przykład

→ Przykład: Widok zmaterializowany `emp_pos_dept_MView` przechowuje imię, nazwisko pracownika, jego zarobki, nazwę stanowiska oraz nazwę departamentu w którym pracuje:

```
CREATE MATERIALIZED VIEW emp_pos_dept_mview AS
```

```
SELECT e.name name, e.surname, e.salary, p.name position, d.name department  
FROM employees e JOIN positions p USING(position_id)  
JOIN departments d USING (department_id);
```

```
SELECT * FROM emp_pos_dept_mview;
```


All Rows Fetched: 43 in 0,047 seconds

	NAME	SURNAME	SALARY	POSITION	DEPARTMENT
1	Piotr	Maciejewski	3000	Programista	Administracja
2	Jose Manuel	Sciarra	3036	Grafik	Administracja
3	Piotr	Janowski	3000	Konsultant	Marketing
4	Andrzej	King	1500	Kadrowy	Marketing

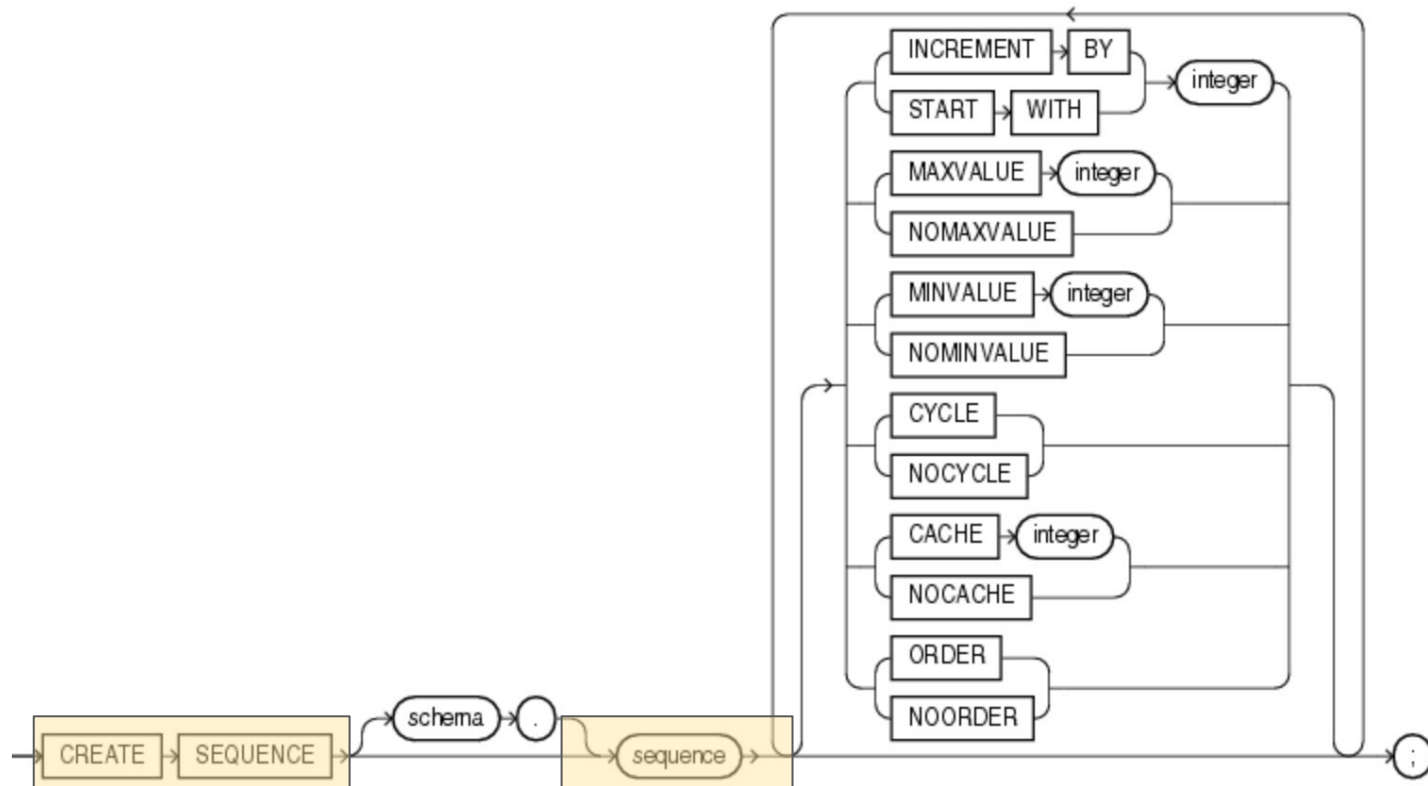
Widoki zmaterializowane - ćwiczenia

1. Zdefiniuj widok, zmaterializowany, który będzie przechowywał imię i nazwisko kierowników i liczbę jego podwładnych.
2. Zdefiniuj widok zmaterializowany przechowujący informacje o sumie budżetów (`estimated_budget`) projektów prowadzonych przez dany departament.
3. Do widoku stworzonego w powyższym poleceniu dodaj kolumnę z informacją o procentowym udziale sumy budżetów projektów w rocznym budżecie danego departamentu. Zaokrąglaj procentowy udział do 2 miejsc po przecinku. Posortuj malejąco względem tego procentowego udziału.
4. Wykonaj polecenia `DROP MATERIALIZED VIEW`, aby usunąć jeden z wcześniej utworzonych widoków.

Sekwencje

- Sekwencja jest obiektem bazy danych, który pozwala na iteracyjne uzyskiwanie liczb całkowitych.
- Sekwencje są na ogół wykorzystywane do generowania wartości kluczy głównych.

Sekwencje - składnia



Sekwencje - opcje definiowania

- Istnieje szereg parametrów pozwalających określić charakterystykę sekwencji. Niektóre z nich zostały pokazane w tabeli:

INCREMENT BY liczba1	Określa, o ile mają być inkrementowane lub dekrementowane kolejne wartości sekwencji.
START WITH liczba1	Określa początkową wartość sekwencji.
MAXVALUE liczba1	Określa maksymalną wartość sekwencji.
NOMAXVALUE	Brak maksymalnej wartości sekwencji (w rzeczywistości maks. to liczba: $10^{28} - 1$)
MINVALUE liczba1	Określa minimalną wartość sekwencji.
NOMINVALUE	Ustawia wartość minimalną wartość 1 dla sekwencji rosnących i $-10^{28} - 1$ dla sek. malejących.
CYCLE	Wskazuje, że po osiągnięciu maksymalnej (minimalnej) wartości sekwencja zacznie generować liczby począwszy od minimalnej (maksymalnej) wartości.
NOCYCLE	Po osiągnięciu wartości maksymalnej (minimalnej) sekwencja przestaje generować wartości.

Sekwencje - przykład

- *Zdefiniuj sekwencję o nazwie `int_seq1`, która będzie generowała wartości całkowite z przedziału $100 \div 1000$:*

```
CREATE SEQUENCE int_seq1 start with 100 maxvalue 1000;
```

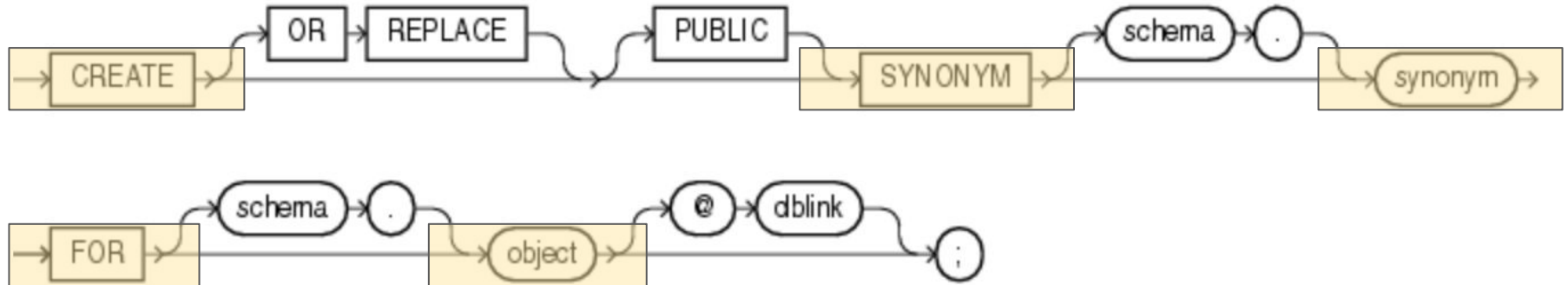
- Po zdefiniowaniu sekwencji możliwe jest odwoływanie się do jej aktualnej wartości (CURRVAL) oraz następnej wartości (NEXTVAL):

```
SELECT int_seq1.NEXTVAL FROM dual;  
SELECT int_seq1.CURRVAL FROM dual;
```

Synonimy

- Synonimy są alternatywnymi nazwami nadawanymi obiektom bazy danych.
- Możliwe jest tworzenie synonimów m.in. dla następujących obiektów BD:
 - tabele, widoki, inne synonimy, pakiety, procedury i funkcje języka PL/SQL, widoki zmaterializowane, sekwencje, obiekty zdefiniowane przez użytkownika.
- Można zdefiniować synonimy:
 - *prywatne* - dostępnych jedynie dla właściciela oraz użytkowników, którzy mają uprawnienia do tabel bazowych
 - *publiczne* - widoczne dla wszystkich użytkowników bazy danych (jednakże wymagane jest nadanie dodatkowych uprawnień do takich obiektów).
- Korzystając z synonimu można wykonywać wszystkie operacje DML.
- Dlaczego stosujemy synonimy:
 - maskowanie schematu i nazwy tabeli (wygoda, czytelność)
 - wprowadzenie niezależności między aplikacją kliencką (konsument danych z tabeli) a fizyczną lokalizacją danych

Synonimy - składnia



Synonimy - przykład

- Przykład: zdefiniuj synonim prywatny o nazwie *emps* dla tabeli *employees*:

```
CREATE SYNONYM emps FOR employees;
```

- Przykład: zdefiniuj synonim prywatny o nazwie *seq_syn* dla sekwencji zdefiniowanej na slajdzie 16:

```
CREATE SYNONYM seq_syn FOR int_seq1;
```

- Przykład: zdefiniuj synonim prywatny o nazwie *MView_1* dla widoku zmaterializowanego zdefiniowanego na slajdzie 11:

```
CREATE SYNONYM MView_1 FOR emp_pos_dept_mview;
```

Sekwencje, synonimy - ćwiczenia

1. Zdefiniuj sekwencję, która: (i) będzie posiadała minimalną wartość 10; (ii) rozpocznie generowanie wartości od 12; (iii) będzie posiadała maksymalną wartość 17; (iv) będzie cykliczna. Następnie wygeneruj kilkanaście wartości za pomocą tej sekwencji i obserwuj rezultaty.
2. Zdefiniuj sekwencję, która będzie generowała malejąco liczby parzyste z przedziału $100 \div 0$.
3. Nadaj synonim dla dowolnej z dwóch poprzednio zdefiniowanych sekwencji i pobierz z niej wartość za pomocą synonimu.



Praca domowa

1. Utwórz widok zawierający złączenia kilku tabel (np. employees, positions, departments, addresses) i spróbuj go wykorzystać do wprowadzenia nowych danych do tych tabel. Jaki komunikat błędu zaobserwowałeś?
Wyszukaj w dokumentacji możliwe rozwiązanie tego problemu.
2. Zdefiniuj widok łączący tabele countries, regions i addresses. Następnie wykorzystując złączenie pomiędzy tym widokiem, a tabelą departments wyświetl kraje oraz regiony położenia wszystkich departamentów.
3. Wyszukaj informacje w Internecie lub w dokumentacji bazy Oracle, w jaki sposób możliwe jest automatyczne wykorzystanie sekwencji do generowania wartości klucza głównego przy dodawaniu danych do tabeli.