

Komercyjne bazy danych 2

Język SQL w Oracle

doc. dr inż. Tomasz Traczyk

Instytut Automatyki i Informatyki Stosowanej
Politechnika Warszawska

Styczeń 2010

Rodzaje zdań języka SQL

DQL (*Data Query Language*) – zapytania SELECT, służące do pozyskiwania danych z bazy

DML (*Data Manipulation Language*) – dodawanie, usuwanie oraz modyfikacja danych

DDL (*Data Definition Language*) – tworzenie i modyfikacja struktur danych oraz do administrowania bazą danych

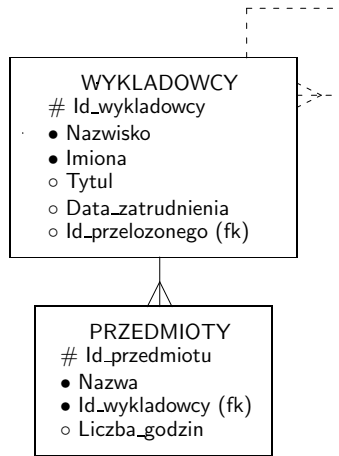
Środowiska wykonywania języka SQL

- ▶ Środowiska konwersacyjne
 - ▶ konsole SQL (SQL*Plus, SQL Developer)
 - ▶ programy administracyjne (Enterprise Manager)
 - ▶ programy do graficznego tworzenia zapytań (np. Oracle Data Query)
- ▶ Języki programowania
 - ▶ zanurzony SQL (*embedded SQL*)
 - ▶ OCI (*Oracle Call Interface*)
 - ▶ ODBC
 - ▶ JDBC
- ▶ Narzędzia czwartej generacji (Oracle Developer, HTML DB czyli Application Express)
- ▶ Programy powszechnego użytku (przez ODBC) np. Microsoft Office z Microsoft Query

Obsługa błędów

- ▶ Obsługa błędów w SQL
 - ▶ sygnalizowane komunikatem
 - ▶ obsługiwane:
 - ▶ błędy składniowe – instrukcja nie jest wykonywana
 - ▶ błędy wykonania – efekty błędnej instrukcji automatycznie wycofywane
 - ▶ transakcja jest kontynuowana
- ▶ Komunikaty o błędach
 - ▶ trójliterowy kod środowiska
 - ▶ numer błędu
 - ▶ komunikat o błędzie (w języku NLS)
- ▶ Propagacja informacji o błędzie
 - ▶ program który spowodował błąd może przechwycić informację (wyjątek)
 - ▶ nieprzechwycona informacja o błędzie ukazuje się na ekranie operatora

Przykładowa struktura danych



DQL¹ – specyficzne cechy Oracle

- ▶ Szczegóły składni
 - ▶ słowo AS niedozwolne przed aliasami tabel
 - ▶ słowo AS nieobowiązkowe przed aliasami kolumn
 - ▶ aliasy kolumn niedozwolone w klauzulach WHERE, GROUP BY i HAVING
 - ▶ dostępne operatory mnogościowe: UNION [ALL], MINUS, INTERSECT
- ▶ Tabela-zaślepka DUAL, np. SELECT SYSDATE FROM DUAL;

¹Data Query Language

Złączenia wg SQL'99

- ▶ CROSS JOIN – produkt kartezjański
- ▶ LEFT|RIGHT|FULL OUTER JOIN – złączenia zewnętrzne
- ▶ JOIN ... ON – złączenie z jawnym podaniem warunku złączenia w klauzuli ON
- ▶ NATURAL JOIN – złączenie naturalne z automatycznym doбором kolumn
 - ▶ wg zgodności nazw kolumn
 - ▶ nieco niebezpieczne!
- ▶ JOIN ... USING – złączenie równościowe ze wskazaniem kolumn
 - ▶ wymienia się nazwy kolumn (zgodne w obu tabelach)
 - ▶ na kolumny z klauzuli USING nie można powoływać się z użyciem aliasu tabeli

Podzapytanie w klauzuli FROM

```
SELECT w.id_wykladowcy, nazwisko, obciazenie,
       liczba_przedmiotow                alias w, by uniknąć niejednoznaczności

FROM wykladowcy w,
     (SELECT id_wykladowcy, SUM(liczba_godzin) AS obciazenie
      FROM przedmioty
      GROUP BY id_wykladowcy                podzapytanie oblicza obciążenia
                                             godzinowe
     ) o,                                    o jest aliasem nadanym wynikowi
                                             podzapytania

     (SELECT id_wykladowcy, COUNT(*) AS liczba_przedmiotow
      FROM przedmioty
      GROUP BY id_wykladowcy                podzapytanie oblicza liczbę prowa-
                                             dzonych przedmiotów
     ) l
WHERE w.id_wykladowcy = o.id_wykladowcy    warunki złączeń tabeli z wynikami
                                             podzapytań

AND w.id_wykladowcy = l.id_wykladowcy
ORDER BY nazwisko;
```

Podzapytanie w klauzuli SELECT

Podzapytanie w klauzuli SELECT musi zwracać wartość skalarną – udaje wyrażenie.

▶ Przykład [21]

Zapytanie hierarchiczne

```
SELECT LPAD(' ',2*(LEVEL-1)) || nazwisko as Nazwisko,  
       id_wykladowcy, id_przelozonego           LEVEL – poziom zagłębienia w hierarchii  
  
FROM wykladowcy  
CONNECT BY PRIOR id_wykladowcy = id_przelozonego  
  
START WITH id_przelozonego IS NULL  
ORDER SIBLINGS BY nazwisko;
```

CONNECT BY – związek między elementami w hierarchii, PRIOR – kierunek przeszukiwania
START WITH – punkt startowy
specjalna klauzula do sortowania w ramach „gałęzi”

Wyrażenia w SQL

- ▶ Budowa
 - ▶ stałe
 - ▶ nazwy kolumn i pseudokolumn
 - ▶ operatory
 - ▶ wywołania funkcji wbudowanych
 - ▶ wywołania funkcji użytkownika (wywoływana funkcja nie może modyfikować danych ani zmiennych pakietów)
- ▶ Stałe
 - ▶ znakowe – w pojedynczych apostrofach
 - ▶ czas i data – tak jak znakowe o ile format zgodny z ustawionym dla sesji albo jawna konwersja: `TO_DATE('1997-10-24', 'YYYY-MM-DD')`
 - ▶ numeryczne – notacja zwykła lub „naukowa”
 - ▶ NULL

Pseudokolumny

- ▶ Używane podobnie jak kolumny
- ▶ Określone jedynie w zapytaniach!

ROWID	Podaje wewnętrzny identyfikator wiersza
ROWNUM	Podaje numer kolejny pobranego wiersza (ale przed sortowaniem)
LEVEL	Podaje poziom zagłębienia w hierarchii, dostępna tylko w zapytaniu hierarchicznym
<code>nazwa_sekwencji.NEXTVAL</code>	Podaje nową wartość z sekwencji
<code>nazwa_sekwencji.CURRVAL</code>	Podaje ostatnią wartość zwróconą przez sekwencję

NULL

- ▶ Stała NULL
- ▶ Wyrażenie z operandem o wartości NULL całe jest równe NULL, wyjątki:
 - ▶ NULL AND fałsz – daje fałsz, NULL OR prawda – daje prawdę
 - ▶ tekst || NULL – daje tekst
 - ▶ NULL jest ignorowany przez funkcje grupujące
- ▶ Pusty napis typu VARCHAR2, czyli ' ', jest równoznaczny z NULL
- ▶ Testowanie wartości sensowne jedynie operatorami IS NULL lub IS NOT NULL
- ▶ W zapytaniu warunek o wartości NULL jest uważany za niespełniony
- ▶ W więzach kontrolnych warunek o wartości NULL jest uważany za spełniony
- ▶ Do zastępowania wartości NULL służą wyrażenia
 NVL(wartość_wejściowa, wartość_zastępcza)
 NVL2(wartość_wejściowa, wart_jeśli_nie_null, wart_jeśli_null)
 COALESCE(wartość_1, wartość_2, ...) – zwraca pierwszą wartość nienullową
- ▶ Wyrażenie zwracające NULL jeśli operandy są równe:
 NULLIF(wartość_1, wartość_2)

Operatory

Arytmetyczne	+ - * /	Operacje arytmetyczne
Logiczne	NOT, AND, OR	Operacje logiczne
Znakowe		Konkatenacja
Data i czas	+ - -	Przesunięcie daty/czasu: dla typu DATE 1 oznacza jedną dobę Odejmovanie dat/czasu: dla typu DATE różnica w dobach (i ich ułamkach)
Porównania	= != ^= <> < > <= >= LIKE BETWEEN a AND b IN(lista_wartości) IS NULL, IS NOT NULL	Równość Brak równości Nierówności Porównanie wartości znakowej z wzorcem (znaki specjalne % i _) Przynależność do przedziału Przynależność do zbioru wartości Test braku wartości

Konwersje

- ▶ Konwersje w SQL są w zasadzie robione automatycznie
- ▶ Funkcje konwersji TO_typ, np. TO_CHAR, TO_DATE, TO_NUMBER
- ▶ Format dla czasu i daty oraz liczb
 - ▶ domyślny – ustawiany dla sesji
 - ▶ wartość początkowa brana z ustawień NLS na stacji klienta
 - ▶ jawnie podany, np.: TO_CHAR(SYSDATE, 'YYYY-MM-DD HH24:MI:SS')

Wzorce formatów – wybór

Liczby	9	cyfra
	0	cyfra, z uzupełnieniem liczby zerem wiodącym lub końcowym
	D	znak dziesiętny (zależny od NLS)
	G	separator grup, np. tysięcy (zależny od NLS)
Data i czas	YYYY	rok (4 cyfry)
	MM	miesiąc cyframi
	MONTH	miesiąc słownie (w języku NLS)
	DD	dzień miesiąca
	DAY	dzień tygodnia słownie (w języku NLS)
	HH	godzina (1-12)
	HH24	godzina (0-24)
	MI	minuty
SS	sekundy	
RR	rok (2 cyfry) ze „zgadywaniem” wieku (Y2K)	

Funkcje wbudowane (wybór)

Znakowe	ASCII	Zamiana znaku na kod liczbowy
	LENGTH	Zwraca długość napisu w znakach
	CHR	Zamiana kodu liczbowego na znak
	LOWER, UPPER	Zmiana liter na małe/wielkie, działa prawidłowo na znakach narodowych
	INITCAP	Zmienia pierwsze litery wyrazów na wielkie, pozostałe na małe
	LPAD, RPAD	Dodaje podaną liczbę powtórzeń wzorca po lewej/prawej stronie napisu
LTRIM, RTRIM		Usuwa określone znaki (np. spacje) z lewej/prawej strony napisu
	INSTR	Bada wystąpienie wzorca w napisie
SUBSTR		Zwraca fragment napisu
	REPLACE	Zamienia wzorzec znaleziony w napisie na inny

Funkcje wbudowane (wybór), c.d.

Arytmetyczne	ABS, EXP, LN, LOG, POWER, SIGN, ROUND, TRUNC, MOD, SIN, COS, TAN, SQRT	Typowe funkcje arytmetyczne
Data i czas	SYSDATE ROUND TRUNC NEXT_DAY	Zwraca bieżący czas i datę Zaokrąglenie daty i czasu do określonej precyzji (miesiąca, kwartału itp.) Obcięcie daty i czasu do określonej precyzji Znalezienie daty najbliższego podanego dnia tygodnia
Konwersja	TO_CHAR, TO_DATE, TO_NUMBER, ROWIDTOCHAR, CHARTOROWID, HEXTORAW, RAWTOHEX	Konwersja do odpowiedniego typu Konwersja adresów wierszy Konwersja danych binarnych

Wyrażenie DECODE

DECODE(wyrażenie, klucz1, wynik1 [,klucz2, wynik2...] [,wartość_domyślna])

Przykłady:

```
SELECT DECODE(tytul, 'prof.', 'W.P.', 'dr', 'Pan', 'Ob.')      wybiera odpowiedni zwrot
       || ' ' || imiona || ' ' || INITCAP(nazwisko) AS      nazwisko jest zamieniane z wielkich
naglowek                                                    liter na zwykle przyjętą formę

FROM wykladowcy;
```

```
SELECT id_przedmiotu,
       DECODE( SIGN(liczba_godzin-2.1), -1, 'niewielkie', 1, 'znaczne') || '
obciążenie' AS obciazenie
FROM przedmioty;                                           użyto funkcji SIGN do porównania
                                                           dwóch liczb
```

Wyrażenie CASE

- ▶ Porównywanie wyrażenia z listą wartości:

```
CASE wyrażenie
WHEN klucz1 THEN wynik1
 [WHEN klucz2 THEN wynik2...]
 [ELSE wartość_domyślna]
```

- ▶ Sprawdzanie warunków (zwracających wartość logiczną):

```
CASE WHEN warunek1 THEN wynik1
 [WHEN warunek2 THEN wynik2...]
 [ELSE wartość_domyślna]
```

Klauzula WITH

- ▶ Pozwala zdefiniować podzapytanie „przed” zapytaniem – definiuje coś w rodzaju lokalnej/chwilowej perspektywy
- ▶ Umożliwia wielokrotne odwołanie się do tego podzapytania po nadanej nazwie
- ▶ Przydatne przy obliczeniach, zwłaszcza z agregacjami

```
WITH sumy AS (
SELECT
  (SELECT sum(lacznie) FROM sprawozdanie_A) AS suma_A,
  (SELECT sum(oblicz_zima) FROM sprawozdanie_B) AS suma_B_zima,
  (SELECT sum(oblicz_lato) FROM sprawozdanie_B) AS suma_B_lato
FROM dual
)
SELECT suma_A, suma_B_zima, suma_B_lato,
       suma_A-suma_B_zima AS roznica_zima,
       suma_A-suma_B_lato AS roznica_lato
FROM sumy;
```

przykład na podzapytania w klauzuli SELECT

łatwe obliczenia na wynikach zapytania z klauzuli WITH

odwołanie do nazwy zapytania z klauzuli WITH

DML² – specyficzne cechy Oracle

- ▶ Podzapytanie (modyfikowalne) dopuszczalne w klauzulach:
 - ▶ UPDATE zdania UPDATE
 - ▶ INTO zdania INSERT
 - ▶ FROM zdania DELETE
- ▶ TRUNCATE TABLE nazwa_tabeli [DROP STORAGE] – kasuje zawartość całej tabeli (i zwalnia pamięć)

UWAGA! TRUNCATE powoduje automatyczny COMMIT

²Data Manipulation Language

Sterowanie blokadami

Zapytanie z blokadą

- ▶ Zapytanie SELECT ... FOR UPDATE OF ... powoduje założenie blokad na pobierane wiersze
- ▶ Uwaga – kosztowne!

Sterowanie przeterminowaniem oczekiwania na blokadzie

- ▶ Przydatne dla SELECT ... FOR UPDATE oraz zdań DML
- ▶ Domyślnie oczekiwanie na blokadzie nie przeterminowuje się
- ▶ Klauzula WAIT n umożliwia kontrolowane oczekiwanie na zwolnienie blokady
- ▶ Klauzula NOWAIT powoduje natychmiastowy wyjątek, jeśli natrafiono na blokadę

Mało znane klauzule DML

Klauzula DEFAULT w zdaniu INSERT

- ▶ Może być użyta zamiast wartości dla kolumny w części VALUES
- ▶ Powoduje podstawienie wartości domyślnej zdefiniowanej dla tej kolumny

Klauzula RETURNING ... INTO ...

- ▶ Stosowana w zanurżonym SQL, np. w PL/SQL
- ▶ Pozwala zwrócić ze zdania DML zmienione przez nie dane do zmiennych języka macierzystego
- ▶ Używana np. do odświeżenia wartości wyświetlanych na formularzach
- ▶ Wynik wcale nie jest oczywisty – mogły zadziałać wyzwalacze!
- ▶ Zawartością klauzuli są dwie listy: kolumn i zmiennych

Mało znane instrukcje DML

Odświeżanie danych

```
MERGE ... WHEN MATCHED THEN UPDATE ...
      WHEN NOT MATCHED THEN INSERT ...
```

- ▶ Przydatna w ładowaniu/odświeżaniu danych, zwłaszcza w hurtowniach
- ▶ Szybsza od niezależnych instrukcji INSERT i UPDATE

Wstawianie warunkowe do wielu tabel

```
INSERT ALL|FIRST WHEN ... THEN INTO t1(...) VALUES ... WHEN ...
ELSE INTO tn(...) VALUES...
```

- ▶ Pozwala rozrzucać dane do wielu tabel w czasie wykonania jednego zdania INSERT
- ▶ Przyspiesza złożone wstawienia oparte na odczycie danych z zapytania, np. w procesach ETL
- ▶ Wykonuje wstawienie dla wszystkich spełnionych warunków lub tylko dla pierwszego

DDL³ – specyficzne cechy Oracle

- ▶ Składnia większości zdań DDL specyficzna dla dialektu Oracle
- ▶ Wszystkie zdania DDL automatycznie wykonują COMMIT!
- ▶ Nie można zmienić nazwy kolumny (w starszych wersjach nie można też usunąć kolumny)
- ▶ Możliwości zmiany typów kolumn są bardzo ograniczone
- ▶ Dla kluczy głównych i unikalnych automatycznie tworzone są indeksy unikalne

³Data Definiton Language

Tabele – tworzenie

```
CREATE TABLE wykladowcy (
  id_wykladowcy   NUMBER(10) NOT NULL      NOT NULL  oznacza kolumnę
                                                         obowiązkową
  ,nazwisko       VARCHAR2(25) NOT NULL
  ,imiona         VARCHAR2(25) NOT NULL
  ,tytul          VARCHAR2(20)
  ,data_zatrudnienia DATE DEFAULT SYSDATE  DEFAULT – wartość domyślna,
                                                         SYSDATE – bieżąca data i czas
  ,id_przelozonego NUMBER(10)
);

COMMENT ON TABLE  wykladowcy IS 'Dane wykładowców';
COMMENT ON COLUMN wykladowcy.id_wykladowcy IS 'Identyfikator
wykładowcy';
```

komentarze umieszczane w słowniku danych

Modyfikacje i usuwanie

- ▶ Usuwanie tabeli
DROP TABLE przedmioty CASCADE CONSTRAINTS;
- ▶ Dodawanie kolumn
ALTER TABLE wykladowcy ADD instytut VARCHAR2(50)
DEFAULT 'Instytut Ogólnych Teorii i Zarządzania' NOT NULL;
 - ▶ kolumny obowiązkowe można dodać tylko:
 - ▶ do pustej tabeli
 - ▶ z klauzulą DEFAULT
- ▶ Zmiana kolumn
ALTER TABLE wykladowcy MODIFY instytut NULL;
ALTER TABLE wykladowcy MODIFY nazwisko VARCHAR2(40);
 - ▶ ograniczone możliwości zmian:
 - ▶ bez ograniczeń dla pustych tabel lub kolumn
 - ▶ zmiana typu w ramach zgodnych typów
 - ▶ zmiana długości tylko w górę

Więzy deklaratywne – definiowanie

- ▶ Klucz główny

```
ALTER TABLE wykadowcy ADD CONSTRAINT wyk_pk  
PRIMARY KEY (id_wykadowcy);
```

Kolumny klucza głównego powinny być NOT NULL
- ▶ Klucze unikalne

```
ALTER TABLE przedmioty ADD CONSTRAINT prz_uk UNIQUE (nazwa);
```
- ▶ Klucze obce

```
ALTER TABLE wykadowcy ADD CONSTRAINT wyk_wyk_fk  
FOREIGN KEY (id_przelozonego)  
REFERENCES wykadowcy (id_wykadowcy);
```

Kolejność kolumn w kluczu obcym powinna być taka sama jak we wskazywanym kluczu głównym lub unikalnym
- ▶ Więzy kontrolne

```
ALTER TABLE wykadowcy ADD CONSTRAINT wyk_chk  
CHECK ( tytul IN ('prof.', 'dr', 'mgr') );
```

W warunku nie wolno używać funkcji niedeterministycznych (np. SYSDATE, UID, USER, USERENV) ani pseudokolumn

Modyfikowanie więzów

Usuwanie więzów

```
ALTER TABLE wykadowcy DROP CONSTRAINT wyk_chk;
```

Włączanie i wyłączanie więzów

```
ALTER TABLE wykadowcy DISABLE CONSTRAINT wyn_chk;
```

```
ALTER TABLE wykadowcy ENABLE CONSTRAINT wyk_chk EXCEPTIONS INTO exceptions;
```

- ▶ Ograniczenia dezaktywizuje się:
 - ▶ dla wykonania zmian struktur lub wartości kluczy
 - ▶ przed masowym ładowaniem danych
- ▶ Ograniczenia można aktywizować bez sprawdzania dotychczasowej zawartości: ENABLE NOVALIDATE
- ▶ Tabela exceptions
 - ▶ trafia do niej informacja o błędnych wierszach
 - ▶ trzeba ją utworzyć skryptem UTLEXCPY.SQL

Sterowanie chwilą sprawdzania więzów

- ▶ Klauzule w CREATE CONSTRAINT:
 - ▶ NOT DEFERRABLE
 - ▶ INITIALLY DEFERRED
 - ▶ INITIALLY IMMEDIATE
- ▶ Zmiana w czasie transakcji:

```
SET CONSTRAINTS ALL|lista IMMEDIATE|DEFERRED;
```
- ▶ Zmienione ustawienie jest ważne do końca transakcji

Sekwencje

Służą do generowania niepowtarzalnych wartości dla kluczy numerycznych.

- ▶ Tworzenie sekwencji

```
CREATE SEQUENCE wykadowcy_seq INCREMENT BY 10  
START WITH 50;
```

generowana sekwencja będzie zaczynać się od 50 i mieć krok 10
- ▶ Użycie sekwencji

```
INSERT INTO wykadowcy (id_wykadowcy, nazwisko, imiona )  
VALUES (wykadowcy_seq.NEXTVAL, 'KOWALSKI', 'Jan');
```

pobrano kolejną wartość z sekwencji i wstawiono do klucza głównego tabeli

 - ▶ pseudokolumna NEXTVAL – odwołanie do powoduje inkrementację sekwencji
 - ▶ pseudokolumna CURRVAL – odwołanie podaje bieżący numer dla danej sesji (bez inkrementacji)
 - ▶ od wersji 11g analogiczne funkcje wbudowane PL/SQL
 - ▶ działanie sekwencji jest niezależne od transakcji i nie wymaga blokad
 - ▶ w numeracji mogą wystąpić nieciągłości!

Indeksy drzewiaste

- ▶ Warunki efektywnego działania indeksów drzewiastych
 - ▶ klucz zawiera wiele wartości o w miarę równomiernej częstotliwości
 - ▶ wartości klucza rozłożone nierównomiernie, ale wyszukuje się te rzadko występujące
 - ▶ najczęściej używana zapytania wyszukują nie więcej niż 15% wierszy
 - ▶ tabela jest duża
 - ▶ klucz indeksowy jest używany w warunku złączenia
- ▶ Warunki, gdy indeksy nie działają efektywnie
 - ▶ tabela jest mała
 - ▶ wartości klucza są rozłożone nierównomiernie, a wyszukuje się te występujące często
 - ▶ wyszukuje się z warunkiem IS NULL albo IS NOT NULL
- ▶ Utrzymywanie indeksów
 - ▶ automatyczne
 - ▶ spowalnia DML!
- ▶ Wykorzystanie indeksów
 - ▶ decyduje optymalizator zapytań
 - ▶ wyszukiwanie przez dopasowanie do początkowego fragmentu klucza

Definiowanie indeksów drzewiastych

```
CREATE INDEX wykladowcy_idx ON wykladowcy (nazwisko, imiona)
                                     klucz indeksowy składa się z dwóch
                                     kolumn
                                     indeks jest umieszczony w odrębnej
                                     przestrzeni tabel
```

TABLESPACE user_index;

lub

```
CREATE UNIQUE INDEX...;
                                     indeks unikalny, nie dopuszcza po-
                                     twórceń klucza
```

Usuwanie

```
DROP INDEX nazwa_indeksu;
```

Indeks jest także automatycznie usuwany, gdy usuwana jest indeksowana tabela.

Grona (*clusters*)

- ▶ Przechowują wiersze (z jednej lub wielu tabel) o wspólnej wartości klucza w jednym obszarze fizycznym
- ▶ Przyspieszają dostęp wg tego klucza
- ▶ Mogą spowalniać inne rodzaje dostępu i DML
- ▶ Odmiany:
 - ▶ indeksowane
 - ▶ z funkcją mieszającą
 - ▶ zapytania z warunkami równościowymi
 - ▶ mała zmienność tabel

Perspektywy

- ▶ Zapamiętane w bazie danych zapytania, do których można odwoływać się tak jak do tabel
- ▶ Typowe zastosowania perspektyw
 - ▶ zapamiętanie często używanych przekrojów danych (np. tzw. *business views*)
 - ▶ ograniczanie dostępu do danych (perspektywa „widzi” dane z uprawnieniami swego twórcy)
 - ▶ budowa klarownych i łatwych do utrzymania interfejsów między podsystemami
 - ▶ umożliwienie szczególnie złożonych zapytań (przestarzałe!)
- ▶ Gdy zapytanie perspektywy przestaje być poprawne, perspektywa nie jest usuwana, ale odwołanie do niej powoduje błąd

Modyfikowalność perspektyw

- ▶ Zależy od wersji Oracle
- ▶ Modyfikacja musi umożliwiać jednoznaczne wprowadzenie zmian do tabel źródłowych
- ▶ Zapytanie nie zawiera: operatorów mnogościowych, funkcji grupujących ani klauzuli GROUP BY, DISTINCT, CONNECT BY, pseudokolumny ROWNUM
- ▶ Dla złączeń – modyfikowalne tylko niektóre kolumny i tylko z jednej tabeli
- ▶ Nie można modyfikować kolumn będących wynikiem wyrażeń
- ▶ Dla INSERT – inicjalizuje wartości kolumn obowiązkowych
- ▶ Klauzula WITH CHECK OPTION dopuszcza tylko modyfikacje dające „widzialne” rezultaty

Definiowanie perspektyw

CREATE OR REPLACE

VIEW profesorowie (id_profesora, nazwisko, imiona)

AS SELECT id_wykladowcy, nazwisko, imiona
FROM wykladowcy WHERE tytuł = 'prof.'
WITH CHECK OPTION;

OR REPLACE pozwala zmienić definicję już istniejącej perspektywy

w nawiasach podaje się tzw. aliasy określające nazwy kolumn wyników perspektywy

ta klauzula ogranicza możliwości zapisu przez perspektywę

Synonimy

- ▶ Pomocnicze nazwy, przez które można odwoływać się do obiektów bazy danych
- ▶ Synonimy publiczne — dostępne dla wszystkich użytkowników (co nie znaczy, że dostępne są same obiekty)
- ▶ Zastosowanie synonimów
 - ▶ ukrycie nazwy schematu (i ew. połączenia b.d.)
 - ▶ uproszczenie
 - ▶ uelastycznienie
- ▶ Tworzenie:
CREATE [PUBLIC] SYNONYM nazwa_synonimu FOR nazwa_obiektu;
- ▶ Gdy obiekt, który był wskazywany przez synonim, przestaje istnieć, synonim nie jest usuwany, ale odwołanie do niego powoduje błąd

Słownik systemowy

- ▶ Zawiera definicje wszystkich obiektów w bazie danych
- ▶ Dla użytkownika dostępny przez perspektywy

Użyteczne perspektywy słownika systemowego (wybór)

USER_TABLES	Tabele użytkownika
USER_TAB_COLUMNS	Kolumny w tabelach
USER_CONSTRAINTS	Ograniczenia deklaratywne
USER_CONS_COLUMNS	Kolumny w ograniczeniach
USER_INDEXES	Indeksy
USER_IND_COLUMNS	Kolumny w kluczach indeksowych
USER_SEQUENCES	Sekwencje
USER_VIEWS	Perspektywy
USER_UPDATABLE_COLUMNS	Modyfikowalne kolumny w perspektywach
USER_SYNONYMS	Synonimy użytkownika
ALL_SYNONYMS	Synonimy dostępne dla użytkownika (w tym publiczne)