Spis treści

WYKLAD 1

1974 – Structured English Query Languafe – SEQUEL

Structured Query Language – SQL

SQL – jezyk zapytan kierowanych do bazy danych \. Kierujemy do bazy POLECENIA wykonania pewnych operacji

SQL – jezyk deklaratywny – operowanie danymi i ich przechowania decyduje SZBD

Prezentacja zapisu składni – notacja Backusa – Naura (BNF)

- Slowa zarezerwowane (nazwa klauzuli, typ danych, operatory logiczne itp.) WIELKIMI LITERAMI
- Nawiasy kwadratowe […] oznacza opcjonalność – zapis [ORDER BY] oznacza, ze klauzula sortowania może wystapic w poleceniu, ale nie jest wymagana
- Pionowa kreska – możliwość wyboru – zapis[ASC | DESC] oznacza wybor. DESC – malejaca, ASC – rosnaca
- Nawiasy klamrowe – element wymagany – zapis {AS | IS} oznacza, ze musi wystapic dokładnie jeden napis AS albo IS
- Nawias okragly – oznacza możliwość powtorzenia elementu

Standard jezyka przewiduje kończenie każdego polecenia znakiem ; (średnik). MS SQL SERVER tego nie wymaga

Dane tekstowe zawsze musza zostać ujęte w pojedyncze cudzyslowia

KOMENTARZE

- Blokowy - /* komentarz */
- Jednoliniowy – komentarz

STRUKTURA JEZYKA

Zapytania – polecenia

- SQL DML – (ang. Data Manipulation Language) „jezyk manipulacji danymi”
    - Odpowiedzialny za operowanie danymi
        - INSERT – wpisywanie nowych danych (rekordow) do bazy danych
        - UPDATE – aktualizacja danych już istniejących
        - DELETE – usuwanie danych (rekordow) z bazy
- SQL DDL – (ang.Data Definition Language) „jezyk definicji danych”
    - Odpowiedzialny za operacje na obiektach danych
        - CREATE – utworzenie nowego obiektu bazy danych
        - ALTER - zmiana struktury obiektu już istniejącego
        - DROP – usuniecie z bazy istniejącego obiektu
- SQL DCL – (ang.Data Control Language) „jezyk kontroli nad danymi”
    - Odpowiedzialny za nadawanie uprawnien do operacji na bazie danych
        - GRANT – przyznanie uprawnien do operacji na obiektach
        - REVOKE – odebranie uprawnien do operacji na obiektach
        - DENY – zabrania operacji na obiektach (ale nie w ORACLE)
- SQL DQL – (ang. Data Query Language) „jezyk definiowania zapytan”
    - Jedno polecenie zaczynające się od SELECT, ale o najbardziej rozbudowanej strukturze. Wykonanie polecen DQL nie ma wpływu na stan danych ani na strukturę obiektow bazy danych

TYPY DANYCH

- Character(n)
- Character Varying(n)
- Numeric(p,q) – liczba dziesietna, zkiziba z o cyfr u zbajy
- Integer

TYPY DANYCH ORACLE

- Char(n)
- Varchar2(n)

- nChar(n)
- nVarchar2(n)
- Number(p,s)
- Integer, Int
- Date
- Timestamp(s)

WYKLAD 2

SELECT

Zwrócenia wartości wyrazen, które mogą być stalymi funkcjami, mogą tez zawierac odwołania do wartości zapisanych w tabelach bazy danych

Pierwsza klauzula rozpoczynajaca się od słowa kluczowego SELECT oraz druga rozpoczynajca się od FROM

Pelna struktura:

SELECT [DISTINCT] wyrażenie (, ...)

FROM nazwa_tabeli (, ...)

[WHERE warunek]

[GROUP BY wyrażenie (, ...)]

[HAVING warunek]

(...)

[ORDER BY wyrażenie (, ...)];

DISTINCT – eliminuje z wyniku powtrzajace się rekordy

Uwaga

SELECT DISTINCT * FROM… nie ma sensu, bo rekordy w tabeli nie mogą się powtarzac

KLAUZA SELECT – WYRAZENIA tworzone według zasad

- Wyrażenia mogą zawierac napisy, liczby, wyrażenia arytmetyczne. W skład mogą wchodzić nazwy kolumn
- Wyrazenia bedace stalymi (liczbami lub napisami) nosza nazwe „literal”

- W skład wyrażenia mogą wchodzić funkcje, które mogą operować na nazwach kolumn
- W wyrażeniach można uzywac operatorow algebraicznych: dodawania, odejmowania, dzielenia, mnożenia, konkatenacji czyli laczenia wyrazen tekstowych

TRIM – obcina spacje

CONCAT – konkatenacja, zwraca wynik jej argumentu polaczone w wartość tekstowa

CAST – funkcja konwertujaca

ALIAS – może być poprzedzony słowem AS

PSEUDOWARTOSC NULL

Jeden z postulatow Codd'a jest koniecznośc wprowadzenie trzeciej wartości logicznej NULL, oznaczajaca brak informacji. Null oznacza albo TRUE albo FALSE

Kazda operacja porównania, arytmetyczna, konkatenacja, w której jeden z operatorow to NULL daje w wyniku NULL

5*NULL = NULL

'Ala' + NULL = NULL

NULL = NULL daje NULL(!)

Operacje logiczne z NULL

NULL OR TRUE jest TRUE

NULL AND FALSE jest FALSE

NOT NULL jest NULL

Jeżeli w defiincji wyrażenia pojawi się składnik, który zwroci pseudowartosc NULL, wówczas cale wyrażenie przyjmie wartość NULL

NVL(wyrazenie1, wyrazenie2) – NULL na wartość znaczace

Alternatywa dla funkcji Isnull i NVl może być funkcja COALESCE

KLAUZULA SELECT – OPCJA DISTINCT

Oznacza eliminacje powtarzających się wierszy(rekordow) Wymaga to posrotowania wierszy wynikowych, aby można było wyznaczyć grupy powtarzających się wierszy, co stanowi znaczy „koszt" obciążenia serwera.

KLAUZULA ORDER BY

W celu umożliwienia sortowania wierszy wynikowych, wprowadza się klauzule sortujaca ORDER BY dewfiniujaca, według której kolumny wynikowej dane maja zostać posortowane, oraz jaki ma być porządek sortowania – rosnący (ASCENDING, ASC) czy malejący (DESCENDING, DESC)

Klauzula ORDER BY może się pojawić w składni polecenia tylko jeden raz, zawsze jako ostatnia

Slowo ASC wskazuje na domyślny kierunek sortowania, może zostać opuszczone w składni polecenia

Wykonanie w jej rezultacie sortowanie wierszy jest „kosztowne”, mocno obciaza serwer

KLAUZULA WHERE, odczytywanie danych z wielu tabel

KLAUZULA WHERE

Jest trzecia w kolejności klauzula w składni instrukcji SELECT. Jest opcjonalna. Pozwala zdefiniować warunek logiczny, ograniczający rekordy zwracane w wyniku działania instrukcji SELECT do tych tylko, dla których przyjmuje on wartość logiczna TRUE. Rekordy, dla których warunek przyjmuje wartość FALSE lub NULL sa z wyniku eliminowane

Warunek WHERE może być koniunkcja(AND), alternatywa(OR) bądź negacja(NOT) innych warunków logicznych. Hierarchia operatorow NOT, AND, OR może zostać zmieniona przy uzyciu nawiasow

OPERATORY W KLAUZULI WHERE

- Arytmetyczne +,-,*,/
- Konkatenacji (łączenia napisow || (w ORCALE)
- Porownan =, <> lub !=, <, <=, >, =>
- Trstujacy Null x IS[NOT]NULL
- Logiczne NOT, AND, OR (z taka hierarchia jak teraz wypiasane)
- Przynaleznosc do listy wartości x[NOT] in (x1,…) np. Kolor IN(‘Czarny’, ‘Bialy’) albo Group IN(201,202)

Wyrazenia musza być tego samego typu(liczbowe, tekstowe, daty)

- Operator zawierania w przedziale domknietym x[NOT]BETWEEN z AND np. Sal BETWEEN 1000 AND 2000

OPERATOR LIKE w implementacji może zostać uzyty do poszukiwania wzorca tekstowego w odniesieniu do wyrazen typu liczbowego i daty

WARUNEK ZLACZENIA DEFINIOWANY W KLAUZLI WHERE

Każdy rekord w tablie Dept zawiera znaczaca wartość deptno – klucz glowny. Jeżeli jakiś pracownik jest zatrudniony w jakims departamencie, to jego rekord w tabeli Emp zawiera w kolumnie deptno wartość – klucz obcy, wskazujący na numer departamentu, w którym jest zatrudniony. Należy do polecenia dodac sformułowany powyżej warunku rownosc wartość deptno

Emp.deptno = Dept.deptno

Dzieki takiej operacji odrzucimy te rekordy, w których Emp.deptno != Dept.deptno, albo Emp.deptno jest NULL

WARUNEK ZLACZENIA DEFINIOWANY W KLAUZULI FROM

Skladnia wygląda nastepujaco

Tab1 JOIN Tab2 On Tab1.Kol1 = Tab2.Kol2

Zlaczenie tabel definiowane przez JOIN może wystapic w następujących wariantach:

[INNER] JOIN

LEFT [OUTER] JOIN

RIGHT [OUTER] JOIN

FULL [OUTER] JOIN

Najczesciej używanym wariantem laczenia tabel jest INNER JOIN. Odczyt realizowany jest według poniższej logiki:

Odczytaj z bazy i wlacz do rozwiązania wszystkie rekordy ze wskazanych w JOIN tabel, dla których warunek określony w ON przyjmuje wartość TRUE

WARUNEK ZLACZENIA OUTER JOIN

Rozszerza wynik o rekordy niepolaczone

OUTER JOIN wystepujae w trzech wariantach

LEFT [OUTER] JOIN

RIGHT [OUTER] JOIN

FULL [OUTER] JOIN

W definicji zlaczen słowa INNER i OUTER sa opcjonalne

ZLACZENIA TABEL LEFT [OUTER] JOIN

Lewostronne zlaczenie zewnętrzne – do wyniku wlaczane sa wszystkie rekordy ze wskazanych JOIN tabel, dla których warunek określony w ON przyjmuje wartość TRUE (tak samo jak w INNER JOIN) oraz

dodatkowo wszystkie pozostale rekordy z tabeli wymienionej po LEWEJ stronie LEFT JOIN, dla których ten warunek przybiera wartość FALSE lub NULL

ZLACZENIE TABEL RIGHT [OUTER] JOIN

Prawostronne zlaczenie zewnętrze, symetryczne dla zlaczenia lewstronnego – do wyniku wlaczane sa wszystkie rekordy ze wskazanych w JOIN tabel, dla których warunek określony w ON przyjmuje wartość TRUE (tak samo jak w INNER JOIN), oraz dodatkowo wszystkie pozostale rekordy w ztabel wyienionej po PRAWEJ stronie RIGHT JOIN, dla których ten warunek przybiera wartość FALSE lub NULL

ZLACZENIA TABEL FULL [OUTER] JOIN

Pelne zlaczenie zewnętrze – jest suma bez powtorzen wynikow LEFT i RIGHT JOIN. Do wyniku wlaczane sa wszystkie rekordy ze wskazanych w JOIN tabel, dla których warunek określony w ON przyjmuje wartość TRUE, (czylu tak, jak w INNER JOIN) oraz dodatkowo wszystkie pozostale rekordy z tabel wymienionych po LEWEJ i PRAWEJ stronie FULL JOIN, dla których ten warunek przybiera wartość FALSE lub NULL

ZLACZENIE OUTER JOIN – alternatywny zapis

Realizuacy zlacznie jednostronne

ZLACZENIE CROSS IN

Iloczyn kartezjański na zbiorach rekordow tabel przywołanych w klauzuli FROM

OPERATORY ALGEBRAICZNE

Wyniki instrukcji SELECT mogą być traktowane, jako zbiory rekordow. Można wykonywać operacje dopuszczalne na zbiorach – sumy, roznicy i przecięcia (iloczynu) zbiorow. Wrunkiem jest jednakowa struktura wszystkich zbiorow – identyczna liczba wynikowych kolumn, identycznie uporządkowanych, a kazda kolumna zawiera wartości o takim samym typie danych

Operacje na zbiorach wynikow realizowane sa przez operatory

UNION[ALL] suma zbiorow wynikow

INTERSECT iloczyn (przecięcie) zbiorow wynikow

EXCEPT/MINUS roznica zbiorow wynikow (minus w Oracle)

Krótkie przypomnienie

- Suma (unia) zbiorow, to zbior zlozony ze wszystkich elementow należących do ktoregokolwiem z sumowanych zbiorow

- Iloczyn (przecięcie) zbiorow A i B to czesc wspolna tych zbiorow, czyli zbior zawierający tylko te elementy, które jednocześnie naleza do zbioru A i do zbioru B
- Roznica zbiorow A i B to zbior zlozony z tych elementow zbioru A, które nie naleza do zbioru B

W wyniku uzycia operatora UNION zwracany jest zbior wynikowych rekordow bez powtorzen, czyli wynik scisle odpowiadający definicji sumy zbiorow. Natomiast uzycie opcjonalnego rozszerzeniea ALL wstrzymuje eliminiacje powtorzen

FUNKCJE AGREGUJACE – KLAUZULE GROUP BY I HAVING

FUNKCJE PODSUMOWUJACE (AGREGUJACE)

Dane zwracane w wyniku zapytania operującego na jednym lub kilku zrodlach rekordow, mogą zostać przeliczone przy uzyciu jednej z funkcji sumarycznych (agregujących)

Count – liczba rekordow lub wartości

Avg – wartość srednia argumentu

Sum – suma

Max – maksymlna wartość

Min – minimalna wartość

Pseudowartosc Null nie sa brane pod uwagę przy obliczaniu wartości funkcji

FUNKCJA COUNT

Zwraca liczbe znaczych wystapien jej argumentu.

KLAUZULA GROUP BY

Tworzy grupy rekordow o jednakowych wartościach zdefiniowanych w niej wyrazen. Na liscie klauzuli może znajdować się więcej niż jedno wyrażenie. W takim przypadku grupowanie odbywa się według unikalnych wartości swekwencji wyrazen

Zasada grupowania jest nastepujaca: dla GROUP BY wyrażenie powstanie tyle grup, ile jest roznych wartości wyrazeenia

Funkcje agregujące nie mogą być umieszczone w klauzuli WHERE

W klauzuli GROUP BY nie wolno odwolywac się do aliasow wyrazen zdefiniowanych w klauzuli WHERE

Utworzenie grup rekordow wskazanych w klauzuli GROUP BY wymaga ich posrtowania (ORDER BY)

KLAUZULA HAVING

Rekordy wynikowe, po operacji grupowania i obliczeniach funkcji agregujących, mogą zostać poddane weryfikacji, poprzez zastosowanie warunku logicznego umieszczonego w klauzuli HAVING.

W klauzuli HAVING można się odwolywac tylko do wyrazen zdefiniowanych w klauzuli SELECT oraz do funkcji agregujących operujących na grupach rekordow utworzonych zgodnie ze wskazaniami klauzuli GROUP BY.

KOLEJNOSC WYSTEPOWANIA KLAUZUL

Kolejnosc występowania klauzul w poleceniu SELECT jest scisle okreslona (SELECT – FROM – WHERE – GROUP BY – HAVING – ORDER BY)

W Oracle GROUP BY i HAVING naprzemiennie

ZAPYTANIA Z UZYCIEM FUNKCJI AGREGUJACYCH – ZASADY OGOLNE

Elementami listy SELECT, klauzuli HAVING i ORDER BY (w przypadku jej uzycia wraz z klauzula GROUP BY) mogą być tylko:

- Stała
- Funkcja podsumowujaca (agregujaca)
- Nazwa kolumny wystepujaca w klauzuli GROUP BY
- Wyrażenie wystepujace w klauzuli GROUP BY, może ono zawierac w sobie nazwy kolumn

Jeżeli wśród wartości wyliczanych przez wyrażenia grupujące pojawia się wartość NULL, jest tworzona dla niej oddzielna grupa

KLAUZULA HAVING VS KLAUZULA WHERE

Wskazania zawarte w klauzuli where realizowane sa w trakcie odczytu rekordow ze wskazanych tabel. Dopóki rekordy nie zostaną odczytane, nie można obliczac wartości funkcji operujących na danych odczytanych, nie można ich również grupować. Najpierw odczyt, a potem operacje

Wskazania zawarte w klauzuli HAVING realizowane sa po odczytaniu wlasciwytch rekordow, pogrupowaniu ich zgodnie ze wskazówkami klauzuli GROUP BY i obliczeniu funkcji agregujących

Przez to co wyżej uniemozliwia uzycia funkcji agregujących w klauzuli WHERE

ALGORYTM WYKONANIA ZAPYTANIA GRUPUJACEGO

1. Powtorz kroki 2-7 dla każdego składnika operatora algebraicznego
2. Rozwaz kolejno wszystkie kombinacje wierszy tabel występujących w klauzuli FROM
3. Do każdej kombinacji zastosuj warunek WHERE. Pozostaw tylko kombinacje dające wartość TRUE

4. Podziel pozostajce kombinacje na grupy
5. Dla każdego pozostającego wiersza reprezentującego grupe oblicz wartość wyrazen na liscie SELECT
6. Do każdej grupy zastosuj warunek w klauzuli HAVING. Pozostaw tylko grupy, dla których wartość warunku jest TRUE
7. Jeśli po SELECT wystepuje DISTINCT, usun duplikaty wśród wynikowych wierszy
8. Jestli trzeba, zastosuj odpowiedni operator algebraiczny
9. Jeśli wystepuje klauzula ORDER BY, wykonaj sortowanie wierszy

WYKLAD 5 – PODZAPYTANIA

PODZAPYTANIA NIESKORELOWANE („ZWYKLE")

Wewnatrz klauzuli WHERE, HAVING i FROm mogą wystapic podzapytania, majace postac zapytania SELECT, ujętego w nawiasy i niekończącego srednikiem wewnątrz nawiasu. W klauzulach tych może znajdować się więcej niż jedno podzapytanie

W podzapytaniu nie można uzywac klauzul ORDER BY

W podzapytaniu zwykłym zbior wynikowych wierszy podzapytania nie zależy od wieszy w glownym zapytaniu – jest ono wykonywane niezależnie i moglo by być samodzielnie realizowanym poleceniem

W podzaptaniu można odwolywac się do nazw kolumn występujących w glownym zapytaniu. Jest to wykorzystywane w podzapytaniach skorelowanych

KILKA PODZAPYTAN W JEDNEJ KLAUZULI

W jednej kaluzuli WHERE, FROM, HAVING może pojawić się więcej niż jedno podzapytanie

ZAGNIEZDZENIE PODZAPYTAN

Podzapytania mogą mieć więcej niż jeden poziom – mogą być zagniezdzone

OPERATORY IN, NOI IN

Gdy używamy operatorow porówna =, >, => … podzapytanie musi zwracać dokaldnie jedna wartość z jednego wiersza

Gdy pozdzapytanie zwraca wiele wierszy, należy uzyc IN lub NOT IN. Operator IN zwraca w wyniku te rekordy, dla których wyrażenie zdefiniowane po WHERE ma wartość rowna któremukolwiek elementowi listy.

WYKLAD 6 – POLECENIA DML

DATA MANIPULATION LANGUAGE – WIADOMOSC OGOLNE

Podzbior polecen SQL odpowiedzialnych za wykonywanie operacji na danych to DML.

Instrukcje DML – INSERT, UPDATE, DELETE wykonują operacje zawsze na JEDNEJ tabeli

INSTRUKCJA INSERT

Instrukcja INSERT jest poleceniem dopisania nowego wiersza do wskazanej tabeli.

WSTAWIENIE POJEDYNCZEGO WIERSZA

INSERT INTO nazwa_tabeli (lista nazw kolumn)

VALUES (lista wartości);

Dla składni pelnej liczba, kolejność i typy danych kolumn wymienionych w klauzuli INSERT musi odpowiadać układowi listy wartości w klauzuli VALUES. Mogą zostać pominięte kolumny dopuszczające NULL, posiadające domyslna wartość DEFAULT

W przypadku składni uproszczonej lista wartości musi odpowiadać kolejności i typom danych WSZYSTKICH kolumn tabeli. Nie można pominąć kolumn dopuszczających NULL

INSTRUKCJA SELECT JAKO ZRODLO DANYCH DLA INSTRUKCJI INSERT

W miejscu klauzuli VALUES może pojawić się polecenie INSERT, które może odczytac z innych tabel wartości

INSTRUKCJA SELECT TWORZACA NOWA TABELE

CREATE TABLE nazwa_nowej_tabeli

AS

(SELECT… FROM…);

INSTRUKCJA UPDATE

UPDATE nazwa_tabeli

SET nazwa_kolumny = wyrazenie1, …

[WHERE warunek];

INSTRUKCJA DELETE

DELETE FROM nazwa_tabeli

[WHERE warunek];

Usuwane sa cale wiersze, dla których warunek w klauzuli WHERE przyjmuje wartość TRUE

INSTRUKCJE COMMIT I ROLLBACK

W bazach danych operujemy pojęciem TRANSAKCJA pod którym rozumiemy pewien zbior polecen DML, które powinny zostać zrealizowane na zasadzie: jeżeli nie wszystkie mogą zostać wykonane poprawnie, to nie jest realizowana żadna,

COMMIT – zatwierdz zmiany, które zostały wprowadzone od ostatniego polecenia Commit lub ROLLBACK

ROLLBACK – wycofaj zmiany, które zostały wprowdzaone od ostatniego polecenia Commit lub ROLLBACL

Zestaw polecn DML który pojawia się pomiędzy kolejnymi poleceniami COMMIT i/lub ROLLBACK nazywamy transakcja

Realizacja transakcji może być realizowana na dwa sposoby

- Kazda poprawnie wykonana operacj DML jest automatycznie zatwierdzana a zmiany w bazie wynikłe z jej dzialnia sa trwale beposrednio
    - Jeżeli chcemy żeby w skład transkacji wchodzila więcej niż jedna instrukcja DML, musimy transakcje jawnie rozpocząć poleceniem BEGIN TRANSCATION i zakonczyc instrukcja COMMIT
- Zmiany wykonane przez wszystkie polecenia DML poczawszy od ostatniego polecenia COMMIT sa nietrwale, a zostan utrwalone dopiero po ich potwierdzeniu poleceniem COMMIT, albo wycofane poleceniem ROLLBACK

INSTRUKCJA TRUNCATE

Stanowi alternatywę dla instrukcji DELETE

TRUNCATE nazwa_tabeli;

W jej składni nie ma klauzuli WHERE, wiec jej wynikiem będzie usuniecie wszystkich wiersyz z tabeli, oczywiście pod warunkiem nie naruszania wiezow integralności

TRUNCATE działa szybciej niż DELETE

INSTRUKCJA GRANT I REVOKE

GRANT – przyznaj uprawnienia wskazanemu użytkownikowi do wykonywania określonych operacji na wskazanym obiekcie bazy danych

REVOKE – odbierz wskazanemu użytkownikowi posiadane uprawnienia do wykonywania określonych operacji na wskazanym obiekcie bazy danych

WYKLAD 7 – POLECENIA DDL

Data Definiton Language odpowiedzialny za operacje na obiektach bazy danych – tworzenie, usuwanie oraz zmiany ich struktury. W DDl wchodzą trzy zestawy polecen, rozpoczynajce się od slow:

CREATE – utworz nowy obiekt

DROP – usun istniejący obiekt

ALTER – zmien sturkture istniejącego obiektu

Pod pojęciem obiektu rozumiemy:

- Tabele
- Widoki (perspektywy)
- Indeksy
- Procedury
- Wyzwalacze
- Bazy danych
- Inne obiekty

Ogolnie postac polecenia DDL ma skladnie:

CREATE|ALTER|DROP Nazwa_klasyObiketuNazwa_obiektu

UTWORZENIE TABELI

CREATE TABLE nazwa_tabeli(

Nazwa_kolumny_1 Typ_danych [więzy_integralnosci]

, Nazwa_kolumny_2 …

);

Nazwa tabeli w ORACLE nie może przekraczaj 30 znakow

Liczba kolumn – max. 1000

WIEZY SPOJNOSCI (INTEGRALNOSCI)

To zespol regul, ktopre gwarantują logiczna spojnosc i / lub poprawność danych wprowdezonych i przechowanych w bazie. Ich zadaniem jest to aby odzwierciedlaly swiat rzeczywisty

Podstawpwymi metodami określania i realizacji wiezow spojnosci na serwerze sa:

- Deklaratywne wiezy spojnosc zawarte w definicjach tabel (instrukcje CREATE TABLE i ALTER TABLE)
- Operacje realizowane przez wyzwalacze bazy danych
- Utworzenie osobnego interfejsu programistycznego API, stanowiącego wartswe posrednia pomiędzy serwerem a aplikacjami klienckimi

GENERALNA ZASADA: tworząc baze danych MY narzucamy reguly jej działania (zatem również wiezy), a za ich przestrzeganie odpowiada SZBD

RODZAJE WIEZOW SPOKNOSCI

- NOT NULL – związane z kolumna, niedopuszcajace do wystąpienia w niej pseudowartości NULL
- PRIMARY KEY – deklaracja klucza glownego w skład którego wchodzi jedna lub kilka kolumn
- FOREIGN KEY REFERENCES nazwa_tabeli – deklaracja klucza obcego, odwołującego się do klucza glownego naszej tabeli
- UNIQUE – deklracaja klucza jednoznacznego, niedopuszcajacego powtarzania się wartości
- CHECK (warunek logiczny) – definicja warunku jaki ma być spełniony dla wartości wstawianych lub modyfkowanych
- DEFALUT wartość – wartość domyslna dla kolumn

SPOSÓB DEKLAROWANIA WIEZOW SPOJNOSCI

```
CREATE TABLE nazwa_tabeli (
     Nazwa_kolumny_1 Typ_danych Więzy_integralności
(…)
);
```

Przykład VII.2.1

```
CREATE TABLE Osoba (
                  IdOsoba   Int          PRIMARY KEY,
                  Imie      Varchar(20)) NOT NULL,
                  Nazwisko  Varchar(30)
                  );
```

Wiezy deklarowane „poza linia” wraz z nadaniem im nazwy poprzez uzyci slowa kluczowego CONSTRAINT