Bazy danych
Laboratorium 7
Info

Konta indywidualne

Username – 3 pierwsze litery nazwiska następnie 3 pierwsze litery imienia a na końcu liczba 18
np. Polak Andrzej → username: POLAND18
Password – **BD2018**

Polecenia

Polecenie **CREATE TABLE** - tworzenie tabeli.

```
CREATE TABLE nazwa_tabeli
(       kolumna1 typ[(rozmiar)]  [DEFAULT wyrazenie1] [wiezy_kolumny1],
        kolumna2 typ[(rozmiar)]  [DEFAULT wyrazenie2] [wiezy_kolumny2], ...,
        [wiezy_tabeli], ...
);
```

Typy danych:

- CHAR(w) - łańcuchy znaków, do 255 elementów (w<=255), stałej długości, uzupełniany spacjami przy porównaniach
- VARCHAR(w) - łańcuchy znaków, do 255 elementów, zmiennej długości
- VARCHAR2(w) - zmienna znakowa, do 2 000 znaków,
- NUMBER - liczby zmiennoprzecinkowe, maks. 38 cyfr znaczących,
- NUMBER(w) - jak NUMBER, ale ograniczone w zakresie w cyfr znaczących (w <= 38),
- NUMBER(w,d) - jak NUMBER(w) ale dodatkowo określamy ilość cyfr po kropce dziesiętnej,
- DATE - data i czas,
- RAW(L) - typ zachowujący się jak CHAR(w), jednak służy do przechowywania danych binarnych,
- TIMESTAMP(w) - przechowuje informacje o dacie i czasie dodatkowo do *w* miejsc po przecinku sekundy (*w* nie może być większe niż 9),
- BLOB, CLOB, BFILE - służą do przechowywania bardzo dużych plików,
- LONG - tylko jedno pole tego typu może występować w rekordzie, ograniczone użycie - nie można używać takiego pola jako indeksu, nie można według niego sortować, nie działają na nim funkcje znakowe. Używane do przechowywania grafiki i dźwięku (multimedia).
- LONG RAW - odpowiednik typu LONG, tylko do przechowywania danych binarnych.

Typy ograniczeń:

- NOT NULL - wartość obowiązkowa,
- CHECK (warunek_logiczny) - wartość spełniająca narzucony warunek; wartości kolumny dla każdego wiersza spełnia zadany warunek (warunek nie może zawierać podzapytań, ani funkcji zmiennych w czasie),
- UNIQUE - wartość niepowtarzalna, jednoznaczna,
- PRIMARY KEY - klucz główny; więzy klucza głównego oznaczają, że kolumna (kolumny) przyjmuje wartości jednoznaczne i niepuste; dla klucza głównego tworzony jest automatycznie indeks,
- FOREIGN KEY, REFERENCES - klucz obcy; wartości z kolumn klucza istnieją we wcześniej zdefiniowanym kluczu kandydującym innej lub tej samej tabeli (kolumna może przyjmować wartość puste).

Definicja ograniczeń (CONSTRAINT) „*inline*”:

```
CREATE TABLE nazwa_tabeli
(   kol1 typ[(rozmiar)]   [DEFAULT wyrazenie1]   [[CONSTRAINT nazwa_ogr_kolumny]
        {   [NOT] NULL
            | UNIQUE
            | PRIMARY KEY
            | REFERENCES tabela(kol1 [, kol2]...)   [ON DELETE { CASCADE | SET NULL }]
            | CHECK (warunek_logiczny)
        } [DISABLE|ENABLE]],
    ...
);
```

Definicja ograniczeń (CONSTRAINT) „*out of line*”:

```
CREATE TABLE nazwa_tabeli
(  ...,
   [[CONSTRAINT nazwa_ogr_tabeli]
            {   UNIQUE ( kol1 [, kol2]... )
                | PRIMARY KEY ( kol1 [, kol2]...)
                | FOREIGN KEY ( kol1 [, kol2]...)   REFERENCES tabela(kol1 [,kol2]...)
                                                    [ON DELETE { CASCADE | SET NULL }]
                | CHECK (warunek_logiczny)
            } [DISABLE|ENABLE]],
   ...
);
```

Przykład

Utwórz tabelę Obywatele składającą się z następujących kolumn:

- pesel – 11 pozycyjna liczba całkowita; kolumna stanowi klucz główny tabeli,
- nazwisko – przechowuje nazwiska do 20 znaków; nazwisko zawsze musi być określone,
- plec – pole jednoznakowe określające płeć, przyjmujące jeden z dwóch znaków K lub M ,
- data_ur – kolumna przechowująca datę urodzenia o domyślnej wartości 01.01.2017 oraz z ograniczeniem dolnej (najstarszej) daty na poziomie 01.01.1950,
- NIP – 10 pozycyjna unikatowa (niepowtarzalna) liczba całkowita, który nie może kończyć się cyfrą parzystą,
- waga – liczba 5-cio cyfrowa z dokładnością dwóch miejsc po przecinku o wartości mieszczącej się w przedziale [15.25, 175.75]
- adres – kolumna przechowująca do 100 znaków,

```
CREATE TABLE Obywatele
( pesel NUMBER(11) constraint obywatele_pk PRIMARY KEY,
  nazwisko VARCHAR2(20) constraint obywatele_naz_nn NOT NULL,
  plec CHAR(1) constraint obywatele_plec_ch CHECK (plec in ('M','K')),
  data_ur DATE DEFAULT TO_DATE('01-01-2017', 'DD-MM-YYYY')
     constraint obywatele_du_ch CHECK(data_ur>TO_DATE('01-01-1950', 'DD-MM-YYYY')),
  NIP NUMBER(10) constraint obywatele_nip_un UNIQUE
    constraint obywatele_nip_ch CHECK( mod(nip,2) <>0),
  waga NUMBER(5,2) constraint obywatele_wag_ch CHECK(waga between 15.25 and 175.75),
  adres VARCHAR2(100)
)
```

Polecenie **INSERT** – polecenie wykorzystywane do wprowadzania danych do tabel.

INSERT INTO {tabela | perspektywa} [(atrybut1 [,...])] {DEFAULT VALUES |
VALUES (wartosc1 [,...]) | SELECT... };

Przykład

```
INSERT INTO obywatele VALUES
 (12345678901, 'Kowalski', 'M', TO_DATE('01-11-1990', 'DD-MM-YYYY'),
  1231231231, 72.25, 'Testowo ul Testowa 1' );

INSERT INTO obywatele (nazwisko, plec, pesel, nip, waga)
     VALUES ('Polak', 'K', 23456782345, 5432312301, 65);

SELECT * FROM obywatele;
```

	PESEL	NAZWISKO	PLEC	DATA_UR	NIP	WAGA	ADRES
1	12345678901	Kowalski	M	90/11/01	1231231231	72,25	Testowo ul Testowa 1
2	23456782345	Polak	K	17/01/01	5432312301	65	(null)

Rys.1 Wynik

Polecenie **UPDATE** – polecenie wykorzystywanie do aktualizacji danych zgromadzonych w tabelach.

UPDATE {tabela | perspektywa}
SET {{atrybut = wartosc | DEFAULT | NULL},
{atrybut = wartosc | DEFAULT | NULL},
...|
(atr1, atr2,...) = (SELECT...)}
[**WHERE** warunek];

Przykład

W tabeli Obywatel osobie o nazwisku Polak zmień datę urodzenia na datę przyjęcia ostatniego pracownika do firmy (z tabeli Pracownicy).

```
UPDATE obywatele SET data_ur=(SELECT max(data_zatr) FROM pracownicy),
                     adres='Anonimowo ul. Anonimowa 0'
WHERE nazwisko LIKE 'Polak';
```

Polecenie **DELETE** – polecenia stosowany do usuwania danych z tabel.

DELETE FROM {tabela | perspektywa} [**WHERE** warunek];

Przykład

Usuń z tabeli dane wszystkich mężczyzn.

```
DELETE FROM obywatele WHERE plec LIKE 'M';
```

DROP TABLE - usuwanie tabeli

DROP TABLE [schemat.]nazwa_tabeli [CASCADE CONSTRAINTS];

```
DROP TABLE obywatele  cascade constraints;
```

Polecenie **COMMIT** – polecenie zatwierdzające wprowadzone dane. Dane, które nie zostały zatwierdzone, są widoczne wyłącznie w ramach sesji (w wypadku awarii zmiany zostaną wycofane).

Polecenie **ROLLBACK** - umożliwia wycofanie niezatwierdzonych zmian w bazie danych.