

Instrukcja laboratoryjna 3	Bazy Danych
	Temat: Wprowadzenie do języka SQL, tworzenie, modyfikacja, wypełnianie tabel
	Przygotowali: mgr inż. Barbara Łukawska, mgr inż. Maciej Lasota

1) Wprowadzenie

SQL (Structured Query Language – Strukturalny Język Zapytań), umożliwia dostęp i przetwarzanie danych w bazie danych na poziomie obiektów modelu relacyjnego, tj. dwuwymiarowych tabel. Pierwszy standard języka SQL powstał w 1987 roku. Najczęściej stosowanym standardem jest standard z 1992 roku SQL92. Jest on do tej pory wykorzystywany przez wszystkie darmowe i komercyjne DBMS. Z technicznego punktu widzenia, SQL jest pod językiem danych. Oznacza to, że jest on wykorzystywany wyłącznie do komunikacji z bazą danych. Nie posiada on cech pozwalających na tworzenie *kompletnych programów*. Użycie SQL, zgodnie z jego nazwą, polega na zadawaniu zapytań do bazy danych. Zapytania można zaliczyć do jednego z trzech głównych podzbiorów:

- ❖ **SQL DML** (ang. *Data Manipulation Language*, czyli Język Manipulacji Danymi),
- ❖ **SQL DDL** (ang. *Data Definition Language*, czyli Język Definicji Danych).
- ❖ **SQL DCL** (ang. *Data Control Language*, czyli Język Kontroli nad Danymi).

Obecnie najnowszymi standardami języka SQL są SQL:2003, SQL:2006 oraz SQL:2008.

2) Tworzenie tabel

```
CREATE TABLE nazwa_tabeli
    (nazwa_kolumny typ [wartość domyślna] [ograniczenia na kolumnie]
    {,nazwa_kolumny typ [wartość domyślna] [ograniczenia na kolumnie]}
    [ograniczenia na tabeli]
    ...
    );
```

Ograniczenia ogólne:

[PRIMARY KEY]	- klucz główny,
[UNIQUE]	- klucz jednoznaczny (wartość unikatowa),
[NOT] NULL	- wartość NULL,
[DEFAULT] wartość_domyślna]	- wartość domyślna,

Ograniczenia na kolumnie:

```
kolumna [CONSTRAINT nazwa_wiezu_pk PRIMARY KEY]  
kolumna [CONSTRAINT nazwa_wiezu UNIQUE]  
kolumna [CONSTRAINT nazwa_wiezu [NOT] NULL]  
kolumna [CONSTRAINT nazwa_wiezu CHECK (kolumna > wyrażenie)]  
kolumna [CONSTRAINT nazwa_wiezu_fk REFERENCES  
nazwa_tabeli_2(kolumna1,[kolumna2]...)] [ON DELETE CASCADE]
```

Ograniczenia na tabeli:

```
[CONSTRAINT nazwa_wiezu_pk PRIMARY KEY (kolumna1,[kolumna2]...)]  
[CONSTRAINT nazwa_wiezu UNIQUE (kolumna1,[kolumna2]...)]  
[CONSTRAINT nazwa_wiezu CHECK(kolumna > wyrażenie)]  
[CONSTRAINT fk_kolumna_tabeli FOREIGN KEY (kolumna1,[kolumna2]...)  
REFERENCES nazwa_tabeli_2(kolumna1,[kolumna2]...)] [ON DELETE CASCADE]
```

Przykład: Tworzenie tabel

```
CREATE TABLE tab1  
    (kol1 NUMBER(5) CONSTRAINT pk PRIMARY KEY,  
    kol2 VARCHAR2(30) CONSTRAINT nn NOT NULL,  
    kol3 DATE DEFAULT(SYSDATE),  
    kol4 NUMBER(8) CONSTRAINT w CHECK (kol4>100)  
    );  
  
CREATE TABLE tab2  
    (kol_a NUMBER(3) CONSTRAINT nn2 NOT NULL,  
    kol_b VARCHAR2(30) CONSTRAINT up CHECK(kol_b = UPPER(kol_b)),  
    kol_c NUMBER(3) CONSTRAINT wm CHECK (kol_c>1 AND kol_c<200),  
    kol_d NUMBER(5) CONSTRAINT fk REFERENCES tab1(kol1),  
    kol_e NUMBER(5) CONSTRAINT c1 NOT NULL  
    CONSTRAINT c2 CHECK (kol_e < 10000),  
    CONSTRAINT kz PRIMARY KEY (kol_a, kol_c)  
    );
```

3) Modyfikowanie tabel

```
ALTER TABLE nazwa_tabeli [rodzaj modyfikacji];
```

Modyfikacja kolumny:

```
ALTER TABLE nazwa_tabeli MODIFY (nazwa_kolumny typ_danych);
```

```
ALTER TABLE nazwa_tabeli MODIFY (nazwa_kolumny NOT NULL);
```

```
ALTER TABLE nazwa_tabeli RENAME COLUMN nazwa_kolumny TO nowa_nazwa_kolumny;
```

Dodanie kolumny:

```
ALTER TABLE nazwa_tabeli ADD (kolumna typ_danych [ograniczenia]);
```

Usunięcie kolumny:

```
ALTER TABLE nazwa_tabeli DROP (kolumna);
```

Dodanie ograniczeń:

```
ALTER TABLE nazwa_tabeli ADD ograniczenie (kolumna);
```

```
ALTER TABLE nazwa_tabeli ADD CONSTRAINT nazwa_wiezu FOREIGN KEY  
(kolumna1, [kolumna2]...) REFERENCES nazwa_tabeli2 (kolumna1, [kolumna2]...);
```

Włączenie i wyłączenie więzów integralności:

```
ALTER TABLE nazwa_tabeli ENABLE CONSTRAINT nazwa_wiezu;
```

```
ALTER TABLE nazwa_tabeli DISABLE CONSTRAINT nazwa_wiezu;
```

Usunięcie więzów integralności:

```
ALTER TABLE nazwa_tabeli DROP CONSTRAINT nazwa_wiezu;
```

Zmiana nazwy tabeli:

```
ALTER TABLE nazwa_tabeli RENAME TO nowa_nazwa_tabeli;
```

4) Usuwanie tabel

```
DROP TABLE nazwa_tabeli;  
DROP TABLE nazwa_tabeli CASCADE CONSTRAINTS;
```

5) Wstawianie i modyfikowanie danych w tabeli

```
INSERT INTO nazwa_tabeli [(nazwa_kolumny{, nazwa kolumny})] VALUES (wartości);
```

Przykład: Wstawianie danych do tabeli

```
INSERT INTO tab1 (kol1, kol2) VALUES (1, 'PIERWSZY');  
INSERT INTO tab1 VALUES (2, 'DRUGI', SYSDATE, 101);
```

Zmiana wartości:

```
UPDATE nazwa_tabeli SET nazwa_kolumny = wartość{, nazwa_kolumny = wartość}  
[WHERE warunek];
```

Przykład: Modyfikacja danych w tabeli

```
UPDATE tab1 SET kol4 = 102 WHERE kol1=1;
```

Usuwanie wartości:

```
DELETE FROM nazwa_tabeli [WHERE (warunek)];
```

6) Transakcje (COMMIT, ROLLBACK, SAVEPOINT)

Transakcje są logiczną jednostką działań w bazie danych, której nie można podzielić. Stanowią one zbiór logicznych zmian w bazie danych, które należy **wykonać wszystkie** lub **nie wykonywać żadnej**.

COMMIT – instrukcja SQL zatwierdzająca wszystkie zmiany w bazie danych bez możliwości późniejszego ich wycofania.

ROLLBACK – instrukcja wycofania zmian i zakończenia transakcji, instrukcja ROLLBACK umożliwi również wycofanie części ostatnio wykonanych instrukcji pod warunkiem, że zaznaczyliśmy tzw. *punkt zachowania (wycofania)* za pomocą instrukcji SAVEPOINT.

SAVEPOINT – instrukcja SQL tworząca punkt zachowania.

Przykład:

```
[instrukcje SQL]
[instrukcje SQL]
...
SAVEPOINT moj_punkt_zachowania;
[instrukcje SQL]
[instrukcje SQL]
...
ROLLBACK TO SAVEPOINT moj_punkt_zachowania
```