

SQL - podstawy,

Niektóre popularne bazy danych SQL:

Microsoft SQL Server, MySQL, PostgreSQL, Oracle, Firebird i Fyrcle, IBM DB2, Microsoft Access
(m.in. na podstawie Wikipedii)

SQL (ang. **Structured Query Language**) – strukturalny język zapytań używany do tworzenia, modyfikowania baz danych oraz do umieszczania i pobierania danych z baz danych.

Język SQL jest językiem deklaratywnym.

Decyzję o sposobie przechowywania i pobrania danych pozostawia się systemowi zarządzania bazą danych (DBMS).

SQL został opracowany w latach 70. w firmie IBM.

Stał się standardem w komunikacji z serwerami relacyjnych baz danych.

Pierwszą firmą, która włączyła SQL do swojego produktu komercyjnego, był Oracle.

W 1986 SQL stał się oficjalnym standardem, wspieranym przez Międzynarodową Organizację Normalizacyjną (ISO) i jej członka, Amerykański Narodowy Instytut Normalizacji (ANSI)

Formy SQL

Z technicznego punktu widzenia, SQL jest **podjęzykiem** danych. Oznacza to, że jest on wykorzystywany wyłącznie do komunikacji z bazą danych.

Nie posiada on cech pozwalających na tworzenie kompletnych programów.

Jego wykorzystanie może być trojaki i z tego względu wyróżnia się trzy formy SQL-a:

1. **SQL interakcyjny** (autonomiczny) wykorzystywany jest przez użytkowników w celu bezpośredniego pobierania lub wprowadzania informacji do bazy.

Przykładem może być zapytanie prowadzące do uzyskania zestawienia aktywności kont w miesiącu.

Wynik jest wówczas przekazywany na ekran, z ewentualną opcją przekierowania go do pliku lub drukarki.

2. **Stacyjny** kod SQL (Static SQL) nie ulega zmianom i pisany jest wraz z całą aplikacją, podczas której pracy jest wykorzystywany.

Stacyjny SQL występuje w dwóch odmianach.

1) **Embedded SQL** (Osadzony SQL) oznacza włączenie kodu SQL do kodu źródłowego innego języka.

Większość aplikacji pisana jest w takich językach jak C++ czy Java, jedynie odwołania do bazy danych realizowane są w SQL.

W tej odmianie statycznego SQL-a do przenoszenia wartości wykorzystywane są zmienne.

2) **Język modułów**. W tym podejściu moduły SQL łączone są z modułami kodu w innym języku.

3. **Dynamiczny kod SQL** (Dynamic SQL) generowany jest w trakcie pracy aplikacji.

Wykorzystuje się go w miejsce podejścia statycznego, jeżeli w chwili pisania aplikacji nie jest możliwe określenie treści potrzebnych zapytań – powstaje ona w oparciu o decyzje użytkownika.

Tę formę SQL generują przede wszystkim takie narzędzia jak graficzne języki zapytań.

Utworzenie odpowiedniego zapytania jest tu odpowiedzią na działania użytkownika.

Składnia SQL

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** – „język manipulacji danymi”): **SELECT, INSERT, UPDATE, DELETE**

SQL DDL (ang. **Data Definition Language** – „język definicji danych”): **CREATE, DROP, ALTER**

SQL DCL (ang. **Data Control Language** – „język kontroli nad danymi”): **GRANT, REVOKE, DENY**

Instrukcje SQL w obrębie zapytań tradycyjnie zapisywane są wielkimi literami, jednak nie jest to wymóg.

Każde zapytanie w SQL-u musi kończyć się znakiem średnika (;).

Dodatkowo, niektóre programy do łączenia się z silnikiem bazy danych (np. psql w przypadku PostgreSQL), używają swoich własnych instrukcji, spoza standardu SQL,

które służą np. do połączenia się z bazą, wyświetlenia dokumentacji itp.

DML - Data Manipulation Language - Modyfikowanie zawartości relacji (tabel) bazy danych

DML (Data Manipulation Language) służy do wykonywania operacji na danych – do ich umieszczania w bazie, kasowania, przeglądania, zmiany.

Najważniejsze polecenia z tego zbioru to: **SELECT, INSERT, UPDATE, DELETE**

DML służy do operacji na danych - do ich umieszczania w bazie, kasowania, przeglądania, zmiany.

Najważniejsze polecenia z tego zbioru to:

- **SELECT** - pobranie z bazy danych,
- **INSERT** - umieszczenie danych w bazie,
- **UPDATE** - zmiana danych,
- **DELETE** - usunięcie danych z bazy

SELECT – pobranie danych z bazy,

np.

```
SELECT * FROM pracownicy WHERE pensja > 2000 ORDER BY staz DESC;
```

```
SELECT * INTO ETAT40 FROM ETAT;
```

INSERT – umieszczenie danych w bazie,

np.

```
INSERT INTO pracownicy (imie, nazwisko, pensja, staz) VALUES ('Jan', 'Kowalski', 5500, 1);
```

```
INSERT INTO Pracownicy ( nr, nazwisko, stanowisko, data_zatr )  
SELECT numer, nazwisko, etat, pracuje_od FROM pracownik WHERE etat in ("ADIUNKT", "ASYSTENT",  
"Profesor");
```

```
INSERT INTO Pracownicy ( nr, nazwisko, stanowisko, data_zatr ) VALUES ("3001", "Mleczko", "portier", "1997-  
01-01");
```

UPDATE – zmiana danych,

np.

```
UPDATE pracownicy SET pensja = pensja * 1.1 WHERE staz > 2;
```

```
UPDATE Pracownicy SET PlacaPod = 700.25;
```

```
UPDATE Pracownik SET PLACA_DOD = 0 WHERE Numer=1050;
```

```
UPDATE Pracownicy SET Pracownicy.PlacaPod = PlacaPod*1.25  
WHERE (stanowisko="Profesor" Or stanowisko="Adiunkt") AND Year(Data_zatr)<1980;
```

DELETE – usunięcie danych z bazy.

np.

```
DELETE FROM pracownicy WHERE imie = 'Jan' AND nazwisko = 'Kowalski';
```

```
DELETE * FROM Personel;
```

Dane tekstowe muszą być zawsze ujęte w znaki pojedynczego cudzysłowu (').

DDL - Data Definition Language - Definiowanie schematu bazy danych

Dzięki DDL (Data Definition Language) można operować na strukturach, w których dane są przechowywane – czyli np. dodawać, zmieniać i kasować tabele lub bazy.

Najważniejsze polecenia tej grupy to: **CREATE, DROP, ALTER**

Polecenia umożliwiają tworzenie i modyfikowanie podstawowych struktur relacyjnej bazy danych, takich jak relacje i perspektywy.

CREATE - utworzenie tablicy lub bazy danych

CREATE DATABASE i **CREATE TABLE**, czyli tworzenie baz i tabel – utworzenie struktury (bazy, tabeli, indeksu itp.),

CREATE DATABASE nazwa_bazy, np. **CREATE DATABASE firma**

CREATE TABLE nazwa_tabeli (kolumny tabeli)

np.

```
CREATE TABLE pracownik (  
id_pracownika INT(5) NOT NULL PRIMARY KEY,  
imie CHAR(20) NOT NULL,  
nazwisko CHAR(50) NOT NULL,  
pensja FLOAT (6,2) NOT NULL  
)
```

Przykłady:

```
CREATE TABLE pracownicy ( imie varchar(255), nazwisko varchar(255), pensja float, staz int );  
CREATE TABLE Pracownicy (nr integer , nazwisko text , stanowisko text, data_zatr date, constraint klucz  
primary key (nr) );
```

DROP - usunięcie tablic, baz danych, indeksów

Używane w zapytaniach definiujących dane

DROP pozwala na usunięcie tablic, baz danych, indeksów

DROP TABLE, DROP DATABASE, ... – usunięcie struktury,

MySQL

```
ALTER TABLE `nazwa_tabeli`
```

```
DROP INDEX nazwa_indexu
```

MS Access

```
DROP INDEX nazwa_indexu
```

```
ON `nazwa_tabeli`
```

MS SQL Server

```
DROP INDEX `nazwa_tabeli`.nazwa_indexu
```

DB2/Oracle

DROP INDEX nazwa_indexu

Przykłady:

DROP TABLE pracownicy;

ALTER TABLE Etat4 DROP COLUMN Salary;

ALTER - umożliwia wykonanie operacji na strukturze istniejącej tabeli takie jak: dodawanie, usuwanie, zmiana nazwy, zmiana typu danych kolumny, a także dodawanie, usuwanie indexu dla kolumny.

ALTER TABLE `nazwa_kolumny` specyfikacja_operacji_alter

ALTER TABLE `nazwa_tabeli` ADD `nazwa_kolumny` typ_danych

ALTER TABLE `nazwa_tablicy` DROP COLUMN `nazwa_kolumny`

ALTER TABLE `nazwa_tablicy` MODIFY `nazwa_kolumny` typ_danych

ALTER TABLE `nazwa_tablicy` CHANGE `stara_nazwa_kolumny` `nowa_nazwa_kolumny` typ_danych

np. *ALTER TABLE pracownicy ADD dzial varchar(255);*

DCL - Data Control Language

DCL (Data Control Language) ma zastosowanie do **nadawania uprawnień** do obiektów bazodanowych.

Najważniejsze polecenia w tej grupie to: **GRANT, REVOKE, DENY**

Istnieją trzy typy uprawnień serwera SQL:

- **uprawnienia do zadań,**
- **uprawnienia do obiektów**
- **uprawnienia niejawne.**

GRANT - nadaje uprawnienia

GRANT <lista praw> ON <element bazy danych > TO <lista użytkowników>

Przykłady:

GRANT ALL PRIVILEGES ON EMPLOYEE TO PIOTR WITH GRANT OPTION

– przyznanie wszystkich praw do tabeli EMPLOYEE użytkownikowi PIOTR z opcją pozwalającą mu nadawać prawa do tej tabeli.

GRANT SELECT, INSERT ON Osoby TO Tadek

GRANT INSERT(Nazwa) ON Produkty TO Tadek, Ania

GRANT SELECT, UPDATE ON Wyniki TO Piotr WITH GRANT OPTION

GRANT UPDATE(Suma) ON Wyniki TO Iwona

GRANT SELECT, UPDATE ON Wyniki TO Iwona

REVOKE – odbiera uprawnienia

Nie można wykonywać czynności. Uprawnienie może zostać zmienione w wyniku członkostwa w roli.

REVOKE <lista praw> ON <element bazy danych > FROM <lista użytkowników>

Przykłady:

REVOKE SELECT, INSERT ON Osoby FROM Tadek

REVOKE INSERT(Nazwa) ON Produkty FROM Tadek, Ania

REVOKE GRANT OPTION FOR INSERT ON Produkty FROM Tadek - odebranie praw nadawania uprawnień innym:

DENY - zabrania wykonywania operacji, jest silniejsze niż grant

Nie można wykonywać czynności. Uprawnienie nie może zostać zmienione w wyniku członkostwa w roli.

np. *DENY SELECT ON Tabela TO Uzytkownik*

Przykładowe zapytania

Przykłady użycia wyżej wymienionych rodzajów zapytań:

SELECT *

FROM pracownicy

WHERE pensja > 2000

ORDER BY staz DESC;

Zwraca tabelę (listę) utworzoną ze wszystkich kolumn (*) tabeli „pracownicy” (FROM pracownicy) zawierającą pracowników, których pensja jest większa niż 2000 (WHERE pensja > 2000) i sortuje wynik malejąco według parametru staz (ORDER BY staz DESC).

INSERT INTO pracownicy
(imie, nazwisko, pensja, staz)
VALUES

('Jan', 'Kowalski', 5500, 1);
Dodaje do tabeli „pracownicy” (INTO pracownicy) wiersz (rekord) zawierający dane pojedynczego pracownika.

UPDATE pracownicy
SET pensja = pensja * 1.1
WHERE staz > 2;

Podwyższa o 10% pensję (SET pensja = pensja * 1.1) pracownikom, których staż jest większy niż 2 (np. lata).

DELETE FROM pracownicy
WHERE imie = 'Jan' AND nazwisko = 'Kowalski';

Usuwa z tabeli „pracownicy” wszystkie wiersze (rekordy) dotyczące pracownika o imieniu „Jan” i nazwisku „Kowalski” (czyli takie, w których pole "imię" ma wartość Jan, a pole "nazwisko" – Kowalski).

CREATE TABLE pracownicy
(
imie varchar(255),
nazwisko varchar(255),
pensja float,
staz int
);

Tworzy tabelę „pracownicy” zawierającą pola tekstowe zmiennej długości (varchar) o nazwach „imie” (imię) i „nazwisko”, o maksymalnej długości 255 znaków, zapisaną za pomocą liczby rzeczywistej (float od ang. floating point) pensję oraz zapisany za pomocą liczby całkowitej (int od ang. integer) staż.

DROP TABLE pracownicy;
Usuwa z bazy tabelę „pracownicy”.
ALTER TABLE pracownicy
ADD dzial varchar(255);

Dodaje do struktury tabeli „pracownicy” kolumnę „dzial” (dział), jako pole tekstowe o długości maks. 255 znaków.

Bezpieczeństwo

Jako, że SQL jest **językiem interpretowanym**, istnieje możliwość nadużyć w przypadku konstruowania zapytań z wykorzystaniem parametrów pochodzących z zewnątrz aplikacji.
Szczególnie podatne na ten typ ataku są tworzone dynamicznie w oparciu o SQL-ową bazę danych serwisy internetowe.
Jeśli twórca aplikacji nie zadba o sprawdzenie poprawności (tzw. walidację) danych wejściowych stanowiących część zapytania, atakujący może być w stanie dopisać do zapytania („wstrzyknąć”) dodatkowe komendy lub zmienić ich sposób działania. Atak taki nosi nazwę SQL injection (wstrzyknięcie kodu za pomocą SQL).

Niektóre systemy bazodanowe używające SQL

- Apache Derby
- Caché
- DATAlegro
- **DB2**
- Firebird
- First SQL
- Greenplum
- HSQL
- Ingres
- Informix
- InterBase SQL
- MaxDB (dawniej SAP DB)
- Microsoft Access
- Microsoft Jet
- Microsoft SQL Server
- Mimer SQL
- MySQL
- mSQL

- Netezza
- **Oracle**
- Oracle Rdb
- **PostgreSQL**
- Pervasive
- SQL/DS
- SQLite
- **Sybase** (Sybase Adaptive Server Enterprise, Sybase SQL Anywhere, Sybase IQ)
- Teradata