

Standardy i składnia języka SQL.

Historia¹

Rozwój relacyjnych baz danych, który miał miejsce w latach 70-tych ubiegłego wieku uwarunkował konieczność opracowania języka do manipulacji, wyciągania i obsługi danych w bazach.

Pierwszym oficjalnym językiem relacyjnych baz danych, był **SEQUEL** (Structured English Query Language), opracowany przez pracowników firmy IBM (Raymond F. Boyce oraz Donald Chamberline). Zaimplementowany w 1973 roku w SYSTEM R – pierwszym silniku bazodanowym opartym o model relacyjny (jednak pierwszy komercyjny system RDBMS to wdrożenie firmy ORACLE w 1979 r.).

Jak sama nazwa wskazuje, SEQUEL to język w domyśle przyjazny dla użytkownika, służący odpytywaniu baz. Jednym z założeń była łatwość tworzenia zapytań, operacji na zbiorach za pomocą słów kluczowych w języku angielskim. Język miał być intuicyjny i prostoty. Te cechy to także założenia samego modelu relacyjnego i chyba właśnie dlatego, systemy baz danych oparte o model relacyjny podbiły świat i są do dziś dominującymi środowiskami bazodanowymi.

Nazwa ewoluowała – SEQUEL, okazała się być nazwą zastrzeżoną przez brytyjską firmę przemysłu lotniczego. Stąd została skrócona do znanej obecnie formy czyli SQL (Structured Query Language).

Najważniejszymi systemami RDBMS (Relational DataBase Management System), w których podstawowym językiem jest SQL, to oczywiście : MS SQL Server, Oracle, DB2, MySQL, PostgreSQL, Sybase.

Obejrzyj: <https://pl.khanacademy.org/computing/computer-programming/sql/sql-basics/v/s-q-l-or-sequel>

Standardy, dialekty

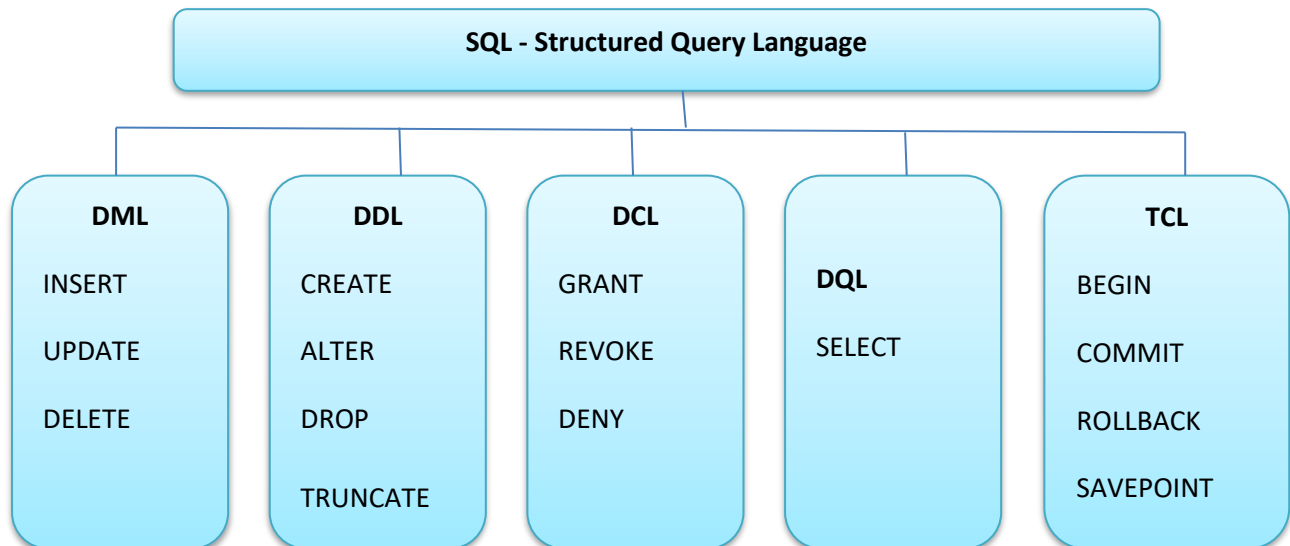
Konkurencyjność rynku spowodowała konieczność ustandaryzowania języka SQL i na szczęście stało się to już w roku 1986, kiedy został opracowany przez ANSI pierwszy standard określany jako SQL:86. Podkreślam, że na szczęście tak szybko, bo choć istnieją istotne różnice np. w nazwach implementowanych funkcji, to ogólne zasady dla relacyjnych baz danych, różnych producentów, są spójne. Ma to znaczenie szczególnie podczas integracji platform i dla nas, pracujących w różnych środowiskach. Powstały, więc dialekty językowe. **Transact-SQL** (T-SQL) – historycznie wprowadzony przez Sybase, rozwijany do dziś przez Microsoft w SQL Server. Inne dialekty, mające duże znaczenie na rynku to oczywiście PL/SQL (firmy ORACLE) oraz SQL/PSM (najpopularniejszy silnik relacyjny w serwisach WWW – MySQL).

Pomimo różnic w dialektach, nazwach funkcji, typach danych – istnieje szeroki wspólny mianownik – relacyjny model oparty o teorię zbiorów. Standardy ANSI są regularnie aktualizowane. Od 1986 roku zostało opublikowanych szereg wersji (aktualnie obowiązująca to ANSI SQL:2011/2011), wprowadzających porządek w nowych funkcjonalnościach. Np. w SQL:2003 zostały wprowadzone standardy związane z obsługą XML.

¹ <http://www.sqlpedia.pl/jezyk-sql-historia-standardy/>

Kategorie SQL

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



Użycie SQL, zgodnie z jego nazwą, polega na zadawaniu zapytań do bazy danych. Zapytania można zaliczyć do jednego z czterech głównych podzbiorów:

- **SQL DML** (ang. *Data Manipulation Language* – „język manipulacji danymi”),
- **SQL DDL** (ang. *Data Definition Language* – „język definicji danych”),
- **SQL DCL** (ang. *Data Control Language* – „język kontroli nad danymi”).
- **SQL DQL** (ang. *Data Query Language* – „język definiowania zapytań”).
- **SQL TCL** (ang. *Transactional Control Language*) – „język kontroli transakcji”

DML

DML (*Data Manipulation Language*) służy do wykonywania operacji na danych – do ich umieszczania w bazie, kasowania, przeglądania oraz dokonywania zmian. Najważniejsze polecenia z tego zbioru to:

- **INSERT** – umieszczenie danych w bazie,
- **UPDATE** – zmiana danych,
- **DELETE** – usunięcie danych z bazy.

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

DDL

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** (np. `CREATE TABLE`, `CREATE DATABASE`, ...) – utworzenie struktury (bazy, tabeli, indeksu itp.),
- **DROP** (np. `DROP TABLE`, `DROP DATABASE`, ...) – usunięcie struktury,

- **ALTER** (np. ALTER TABLE ADD COLUMN ...) – zmiana struktury (dodanie kolumny do tabeli, zmiana typu danych w kolumnie tabeli).
- **TRUNCATE** - powoduje usunięcie danych z tabeli (np. TRUNCATE TABLE klienci).

DCL

DCL (Data Control Language) ma zastosowanie do nadawania uprawnień do obiektów bazodanowych. Najważniejsze polecenia w tej grupie to:

- **GRANT** – pozwala użytkownikowi lub roli na wykonywanie operacji określonej przez nadane uprawnienie (np. 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).
- **REVOKE** – usuwa uprzednio nadane lub odmówione uprawnienie (np. REVOKE ALL PRIVILEGES ON EMPLOYEE FROM PIOTR – odebranie użytkownikowi wszystkich praw do tabeli EMPLOYEE).
- **DENY** - odmawia użytkownikowi lub roli określonego uprawnienia i zapobiega odziedziczeniu go po innych rolach

DQL

DQL (Data Query Language) - to instrukcje, za pomocą których możesz otrzymać z bazy określone dane. Najważniejszym poleceniem jest tutaj **SELECT**.

TCL (język kontroli transakcji)

SQL **TCL** (ang. Transactional Control Language) – „język kontroli transakcji”

- **BEGIN** - - otwiera transakcję
- **COMMIT** -zatwierdza transakcję
- **ROLLBACK**- cofnij transakcję
- **SAVEPOINT** – zapis punkt transakcji

Najbardziej znane dialekty języka SQL:

- **T-SQL** (Transact-SQL)– stosowany na serwerach Microsoft SQL Serwer
- **PL/SQL** (Procedural Language /sql)- stosowany na serwerach ORACLE
- **SQL PL**- wersja używana na serwerach firmy IBM
- **SQL/PSM**- (najpopularniejszy silnik relacyjny w serwisach WWW –MySQL).
- **PL/pgSQL**- (na serwerach PostgreSQL) -Procedural Language /PostgreSQL

Transact-SQL, T-SQL oznacza transakcyjny SQL, czyli rozszerzenie języka SQL umożliwiające tworzenie konstrukcji takich jak pętle, instrukcje warunkowe oraz zmienne. Jest używany do tworzenia wyzwalaczy, procedur i funkcji składowanych w bazie

Podstawy składni zapytań SQL

Składnia DQL (Data Querying Language) – składnia zapytań.

Rozdział ten przedstawia opis kroków i składni przetwarzania zapytań, zgodnie z ich faktyczną kolejnością (która jest ściśle określona).

```
(Step 5)    SELECT      -- określanie kształtu wyniku, selekcja pionowa  
(kolumn)  
(Step 1)    FROM        -- określenie źródła (źródeł) i relacji między nimi  
(Step 2)    WHERE       -- filtracja rekordów  
(Step 3)    GROUP BY   -- grupowanie rekordów  
(Step 4)    HAVING     -- filtrowanie grup  
(Step 6)    ORDER BY   -- sortowanie wyniku
```

Domyślnie, sortowanie odbywa się w sposób rosnący (ASC ang. ascending). Dla każdej z kolumn możemy określić właściwy kierunek sortowania rekordów :

- rosnąco **ASC** ascending – jest to domyślny sposób i nie trzeba go jawnie określać.
- malejąco **DESC** descending – konieczne jest jawne określenie przy atrybucie, po którym chcemy w ten sposób sortować.

Przykład:

Wyświetlenie informacji o liczbie pracowników, pracujących w zespołach wieloosobowych (więcej niż jeden pracownik) w ramach Działu IT. Końcowy wynik posortujemy rosnąco po liczbie pracowników.

Kwerenda realizująca to zadanie będzie wyglądała następująco :

```
SELECT Zespól , COUNT( IdPrac ) AS LiczbaPracowników  
FROM Pracownicy  
WHERE Dział = 'IT'  
GROUP BY Zespól  
HAVING COUNT ( IdPrac ) > 1  
ORDER BY LiczbaPracowników
```

Inne przydatne polecenia:

- **TOP n** – gdzie n określa ilość wyświetlanych wierszy, np: `SELECT TOP 2 * FROM pracownik` (wyświetli dwa pierwsze rekordy z tabeli pracownik W MS SQL)
- **LIMIT n** – gdzie n określa ilość wyświetlanych wierszy, np: `SELECT * FROM pracownik LIMIT 2` (w MySQL)
- **DISTINCT** służy do pobrania z tabeli unikatowych danych, które nie będą się powtarzać.