

# Bazy danych

## Wykład 4\_1

**Temat:** Język SQL i T-SQL – wprowadzenie

**Sławomir Świętoniowski**

slawomir-swietoniowski@wp.pl

# Plan wykładu

---

1. Standard języka SQL i jego historia.
2. Podzbiory funkcjonalne języka SQL.
3. Język Transact-SQL w MS SQL Server 2008 R2.

# Co to jest język SQL?

---

## SQL – Structured Query Language

(wymowa: `es-`kju-`el lub `si:kuel):

- Najbardziej rozpowszechniony, ustandaryzowany język baz danych.
- Pozwala na:
  - **definiowanie struktury bazy danych** (podzbiór DDL – Data Definition Language), np. CREATE TABLE, ALTER VIEW, DROP PROCEDURE;
  - **operowanie na danych**: dodawanie, udostępnianie, modyfikowanie i usuwanie (podzbiór DML – Data Manipulation Language), np. INSERT, SELECT, UPDATE, DELETE.
  - **zarządzanie dostępem do danych** (podzbiór DCL – Data Control Language), np. GRANT, DENY, REVOKE;
  - **definiowanie więzów integralności** (podzbiór DIL – Data Integrity Language), np. CREATE RULE.

# SQL – historia (1)

---

- 1970–1980 Edgar F. Codd (IBM Research Laboratory, CA, USA) opublikował teorię relacyjnego modelu danych, który stał się podstawą relacyjnych baz danych. Spowodowało to liczne badania naukowe i rozwój wielu produktów DBMS.
- 1974 Pojawił się język relacyjny *Structured English Query Language* – SEQUEL (twórca: Donald Chamberline, IBM). Prototypowa implementacja nosiła nazwę SEQUEL-XRM.
- 1976–1977 Została opracowana poprawiona wersja języka, nazwana SEQUEL/2 (SQL). Na jej podstawie IBM opracował ulepszony język *System R*, wdrożony u wielu klientów firmy IBM.

# SQL – historia (2)

---

- 1980–1983     IBM zaprezentował nowe wersje produktów SQL: SQL/DS dla środowiska VSE (1981 r.) i środowiska VM (1982 r.) oraz DB2 dla środowiska MVS (1983 r.).
- 1977–1990     Po sukcesie i akceptacji produktów IBM, inne firmy zaczęły rozwijać własne produkty bazodanowe, oparte na modelu relacyjnym i języku SQL: ORACLE (1977 r.), DG/SQL (Data General Corporation, 1984 r.), SYBASE (Sybase Inc., 1986 r.), INGRES (Relational Technology Inc., 1981-1985 r.), IDM (Britton-Lee Inc, 1982-1985 r.).
- 1982            Ustanowienie pierwszego standardu języka SQL przez ANSI (American National Standards Institute).

# SQL – historia (3)

---

- 1987 Akceptacja standardu ANSI przez ISO (International Organisation for Standardization) jako standardu międzynarodowego: „SQL-86” (lub SQL1).
- 1989 Zatwierdzenie rozszerzonego standardu SQL IEF (Integrity Enhancement Feature): „SQL-89” (SQL2).
- 1992 Rozszerzony standard międzynarodowy: „ISO/IEC 9075”, zwany powszechnie jako „SQL-92” (lub SQL3). Ta wersja jest najczęściej implementowana w produktach komercyjnych.

# SQL – historia (4)

---

- 1999/2003      Standard ANSI SQL:2003 (wersja rozwojowa SQL:1999). Dodano nowe typy danych, w tym obsługę XML, pojawiło się nowe polecenie MERGE i jeszcze kilka innych elementów (SQL4/SQL5).
- 2008            Standard ANSI SQL:2008. Dalsza aktualizacja języka SQL (SQL6).
- 2011            Standard ANSI SQL:2011. Najaktualniejsza wersja standardu (SQL7).

# Standard SQL – korzyści i wady

---

## Zalety:

- redukcja kosztów szkoleń,
- przenośność aplikacji,
- długowieczność aplikacji,
- ułatwienie komunikacji międzysystemowej,
- ułatwienie użytkownikowi wyboru systemu DBMS.

## Wady:

- ograniczenie kreatywności,
- język SQL jest niedoskonały jako język relacyjny,
- standard SQL ma wiele braków, m.in. nie określa wielu elementów potrzebnych w praktyce (np. dane typu logicznego, nazewnictwo obiektów).



# Plan wykładu

---

1. Standard języka SQL i jego historia.

**2. Podzbiory funkcjonalne języka SQL.**

3. Język Transact-SQL w MS SQL Server 2008 R2.

# Podzbiory języka SQL

---

- Język definiowania danych (ang. *data definition language*, DDL).
- Język operowania danymi (ang. *data manipulation language*, DML).
- Język kontroli danych (ang. *data control language*, DCL).
- Język integralności danych (ang. *data integrity language*, DIL).

# Jezyk definiowania danych, DDL

---

Służy do tworzenia i modyfikowania struktury bazy danych: tabel i ograniczeń deklaratywnych, relacji, innych obiektów bazy danych (np. widoków).

# Język operowania danymi, DML

---

Służy do implementacji tzw. operacji CRUD: wstawiania, odczytywania, modyfikowania i usuwania danych.

# Język kontroli danych, DCL

---

Wyodrębniony podzbiór języka SQL, służący do definiowania systemu bezpieczeństwa: użytkowników, ról oraz ich uprawnień na serwerze i bazach danych.

# Język integralności danych, DIL

---

Wykorzystywany do definiowania ograniczeń deklaratywnych i reguł, które zapewniają integralność danych. Ponieważ podzbiór ten częściowo nachodzi na DDL, rzadko jest wyodrębniany.

# Plan wykładu

---

1. Standard języka SQL i jego historia.
2. Podzbiory funkcjonalne języka SQL.
- 3. Język Transact-SQL w MS SQL Server 2008 R2.**

# Język SQL w MS SQL Server 2008 R2

**Transact-SQL (T-SQL)** – transakcyjny język SQL, wbudowany w systemie bazodanowym Microsoft SQL Server 2008 R2:

- jest zgodny ze standardem ANSI-92 na poziomie podstawowym (np. jeśli chodzi o polecenia INSERT, SELECT, UPDATE, DELETE);
- udostępnia pewne funkcje ANSI-99, a także jego rozszerzenia np.:
  - programy przechowywane na serwerze (procedury przechowywane, wyzwalacze, funkcje użytkownika);
  - dodatkowe typy danych (także definiowane przez użytkownika) np. table;
  - role w bazach danych;
  - wartości domyślne pól tabel;
  - duży zbiór funkcji wbudowanych (np. GETDATE( )).
- jest podstawą implementacji ogromnie licznych, wbudowanych procedur systemowych (np. *sp\_helpdb*), które umożliwiają wykonywanie wielu zaawansowanych czynności administracyjnych i programistycznych z poziomu konsoli SQL (bez narzędzi wizualnych, kreatorów itp.).



# Transact-SQL – przykład

Język T-SQL pozwala na pisanie zaawansowanych programów w obrębie warstwy bazodanowej, dzięki złożonym konstrukcjom kodu strukturalnego (przykład poniżej). Brakuje z nim natomiast elementów typowo obiektowych.

```
IF EXISTS -- Sprawdzenie, czy ten sam student nie jest zapisywany drugi raz.
    (SELECT *
      FROM Student
      WHERE PESEL = @PESEL)
BEGIN
    RAISERROR('W bazie jest już student o tym numerze PESEL!', 16, 1)
    RETURN
END

DECLARE @IndeksMin int
DECLARE @IndeksMax int
-- Pobranie dopuszczalnego zakresu numeru indeksu.
SELECT @IndeksMin = KluczeMin, @IndeksMax = KluczeMax
FROM OddzialUczelni
WHERE OddzialMacierzysty = 1

SET ANSI_NULLS ON
IF (@NrIndeksu IS NOT NULL) -- Jeśli numer indeksu został podany jako parametr.
BEGIN
    -- Sprawdzenie, czy numer mieści się w zakresie ustalonym dla tego oddziału.
    IF (@NrIndeksu NOT BETWEEN @IndeksMin AND @IndeksMax)
    BEGIN
        RAISERROR('Podany numer indeksu jest błędny!', 16, 1)
        RETURN
    END
    ELSE
        INSERT Student (NrIndeksu, Nazwisko, Imie, PESEL, NIP, Adres, Miejscowosc,
                        KodPocztowy, Telefon, Email, DataZapisania, DataWypisania, IdGrupy,
```

# Literatura

---

1. BEYNON-DAVIES P., *Systemy baz danych – nowe wydanie*, WNT, Warszawa 2003.
2. DATE C. J., DARWEN H., *SQL. Omówienie standardu języka*, WNT, Warszawa 2000 (książka dostępna w bibliotece WSIZ “Copernicus”).
3. MICROSOFT, *Books On-Line* – dokumentacja systemu *MS SQL Server*, Microsoft Corp. 1988 – 2000.
4. RANKINS R., JENSEN P., BERTUCCI P., *Microsoft SQL Server 2000. Księga eksperta.*, HELION, Gliwice 2003 (książka dostępna w bibliotece WSIZ „Copernicus”).
5. Strona MSDN: <http://msdn.microsoft.com>.
6. WAYMIRE R., SAWTELL R., *MS SQL Server 2000 dla każdego*, HELION, Gliwice 2002 (książka dostępna w bibliotece WSIZ “Copernicus”).

# Bazy danych

Wykład 4\_1

Dziękuję za uwagę !