

# SQL

## Structured Query Language

Karina Burnicka   Viktoria Ługowska

19.04.2021

## Część I

### Omównienie języków zapytań

# Omównienie języków zapytań

Czym jest język zapytań?

## Definicja

Język zapytań (ang.) query language – język stosowany do formułowania zapytań w odniesieniu do baz danych. W odpowiedzi uzyskuje się zestawienia danych, zwane raportami. Najbardziej znane języki zapytań to SQL oraz xBase

# Omównienie języków zapytań

Najpopularniejszy język zapytań

## SQL

Strukturalny język zapytań używany do tworzenia, modyfikowania baz danych oraz do umieszczania i pobierania danych z baz danych. Opracowany przez IBM, wydany w 1974r. Od 1986 stał się oficjalnym standardem wspieranym przez ISO.

## Inne języki zapytań

- 1 Language INtegrated Query (LINQ)
- 2 Query By Example (QBE)
- 3 XQuery (ang. XML query language)
- 4 xBase
- 5 POSTQUEL
- 6 Informix-4GL
- 7 System zarządzania relacyjną bazą danych (RDBMS)

## Inne języki zapytań

- 1 Language INtegrated Query (LINQ)
- 2 Query By Example (QBE)
- 3 XQuery (ang. XML query language)
- 4 xBase
- 5 POSTQUEL
- 6 Informix-4GL
- 7 System zarządzania relacyjną bazą danych (RDBMS)

## Inne języki zapytań

- 1 Language INtegrated Query (LINQ)
- 2 Query By Example (QBE)
- 3 XQuery (ang. XML query language)
- 4 xBase
- 5 POSTQUEL
- 6 Informix-4GL
- 7 System zarządzania relacyjną bazą danych (RDBMS)

## Inne języki zapytań

- 1 Language INtegrated Query (LINQ)
- 2 Query By Example (QBE)
- 3 XQuery (ang. XML query language)
- 4 xBase
- 5 POSTQUEL
- 6 Informix-4GL
- 7 System zarządzania relacyjną bazą danych (RDBMS)



## Inne języki zapytań

- 1 Language INtegrated Query (LINQ)
- 2 Query By Example (QBE)
- 3 XQuery (ang. XML query language)
- 4 xBase
- 5 POSTQUEL
- 6 Informix-4GL
- 7 System zarządzania relacyjną bazą danych (RDBMS)

## Inne języki zapytań

- 1 Language INtegrated Query (LINQ)
- 2 Query By Example (QBE)
- 3 XQuery (ang. XML query language)
- 4 xBase
- 5 POSTQUEL
- 6 Informix-4GL
- 7 System zarządzania relacyjną bazą danych (RDBMS)

## Inne języki zapytań

- 1 Language INtegrated Query (LINQ)
- 2 Query By Example (QBE)
- 3 XQuery (ang. XML query language)
- 4 xBase
- 5 POSTQUEL
- 6 Informix-4GL
- 7 System zarządzania relacyjną bazą danych (RDBMS)

## Część II

### Omównienie standardów ISO

## 2 Standardy ISO SQL

- 1 SQL:86 – pierwszy oficjalny standard
- 2 SQL:89 – ograniczenia integralności
- 3 SQL:92 – typy danych, operatory, tworzenie tabel tymczasowych

- 1 SQL:86 – pierwszy oficjalny standard
- 2 SQL:89 – ograniczenia integralności
- 3 SQL:92 – typy danych, operatory, tworzenie tabel tymczasowych

- ❶ SQL:86 – pierwszy oficjalny standard
- ❷ SQL:89 – ograniczenia integralności
- ❸ SQL:92 – typy danych, operatory, tworzenie tabel tymczasowych



- ❶ SQL:99 – obiektowość, typy tablicowe i strukturalne, zapytania rekurencyjne, wyrażenia regularne, osadzenie w Javie
- ❷ SQL:2003, SQL:2006 – obsługa danych XML
- ❸ SQL:2008 – klauzula FETCH, TRUNCATE TABLE
- ❹ SQL:2011 – nowe opcjonalne elementy składniowe SQL
- ❺ SQL:2016 - obsługa plików JSON, polimorficznych funkcji tabelowych
- ❻ SQL:2019 – tablice wielowymiarowe SQL/MDA

- 1 SQL:99 – obiektowość, typy tablicowe i strukturalne, zapytania rekurencyjne, wyrażenia regularne, osadzenie w Javie
- 2 SQL:2003, SQL:2006 – obsługa danych XML
- 3 SQL:2008 – klauzula FETCH, TRUNCATE TABLE
- 4 SQL:2011 – nowe opcjonalne elementy składniowe SQL
- 5 SQL:2016 - obsługa plików JSON, polimorficznych funkcji tabelowych
- 6 SQL:2019 – tablice wielowymiarowe SQL/MDA

- 1 SQL:99 – obiektowość, typy tablicowe i strukturalne, zapytania rekurencyjne, wyrażenia regularne, osadzenie w Java
- 2 SQL:2003, SQL:2006 – obsługa danych XML
- 3 SQL:2008 – klauzula FETCH, TRUNCATE TABLE
- 4 SQL:2011 – nowe opcjonalne elementy składniowe SQL
- 5 SQL:2016 - obsługa plików JSON, polimorficznych funkcji tabelowych
- 6 SQL:2019 – tablice wielowymiarowe SQL/MDA

- 1 SQL:99 – obiektowość, typy tablicowe i strukturalne, zapytania rekurencyjne, wyrażenia regularne, osadzenie w Javie
- 2 SQL:2003, SQL:2006 – obsługa danych XML
- 3 SQL:2008 – klauzula FETCH, TRUNCATE TABLE
- 4 SQL:2011 – nowe opcjonalne elementy składniowe SQL
- 5 SQL:2016 - obsługa plików JSON, polimorficznych funkcji tabelowych
- 6 SQL:2019 – tablice wielowymiarowe SQL/MDA

- 1 SQL:99 – obiektowość, typy tablicowe i strukturalne, zapytania rekurencyjne, wyrażenia regularne, osadzenie w Javie
- 2 SQL:2003, SQL:2006 – obsługa danych XML
- 3 SQL:2008 – klauzula FETCH, TRUNCATE TABLE
- 4 SQL:2011 – nowe opcjonalne elementy składniowe SQL
- 5 SQL:2016 - obsługa plików JSON, polimorficznych funkcji tabelowych
- 6 SQL:2019 – tablice wielowymiarowe SQL/MDA

- 1 SQL:99 – obiektowość, typy tablicowe i strukturalne, zapytania rekurencyjne, wyrażenia regularne, osadzenie w Javie
- 2 SQL:2003, SQL:2006 – obsługa danych XML
- 3 SQL:2008 – klauzula FETCH, TRUNCATE TABLE
- 4 SQL:2011 – nowe opcjonalne elementy składniowe SQL
- 5 SQL:2016 - obsługa plików JSON, polimorficznych funkcji tabelowych
- 6 SQL:2019 – tablice wielowymiarowe SQL/MDA

## Część III

### Podstawowe zasady i postaci języka SQL

## DDL

Język Definicji Danych – DDL (ang. Data Definition Language)

## DML

Język Manipulacji Danymi – DML (ang. Data Manipulation Language)

## DCL

Język Kontroli Danych - DCL (ang. Data Control Language)

## DQL

Język Zapytań do Danych - DQL (ang. Data Query Language)



# Język Definicji Danych - DDL

Podstawowe zasady i postaci języka SQL

## CREATE

definiowanie obiektów w bazie danych

## ALTER

modyfikowanie obiektów w bazie danych

## DROP

usuwanie obiektów z bazy danych

# Język Manipulacji Danymi - DML

Podstawowe zasady i postaci języka SQL

## INSERT

wstawianie do tabeli nowych wierszy

## UPDATE

modyfikowanie wierszy w tabeli

## DELETE

usuwanie wierszy z tabeli

### GRANT

przydzielenie prawa do danych

### REVOKE

pozbawienie prawa do danych

### DENY

bezwarunkowe pozbawienie prawa do danych

## SELECT

pobieranie danych z tabel

- 1 Zapytania/Instrukcje
- 2 Klauzule
- 3 Identyfikatory
- 4 Wyrażenia
- 5 Operatory
- 6 Predykaty

- 1 Zapytania/Instrukcje
- 2 Klauzule
- 3 Identyfikatory
- 4 Wyrażenia
- 5 Operatory
- 6 Predykaty

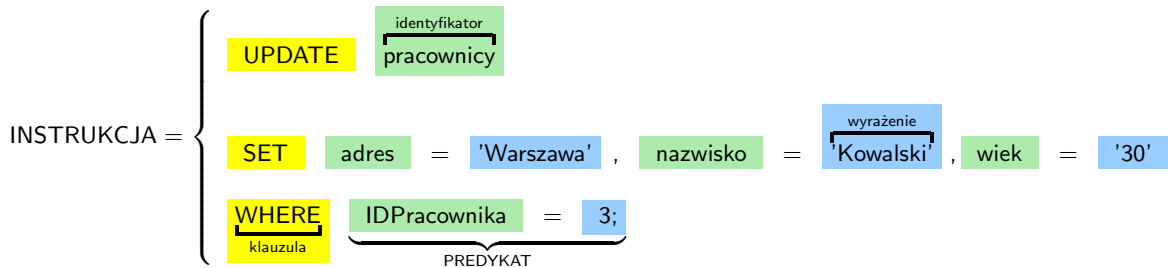
- 1 Zapytania/Instrukcje
- 2 Klauzule
- 3 Identyfikatory
- 4 Wyrażenia
- 5 Operatory
- 6 Predykaty

- 1 Zapytania/Instrukcje
- 2 Klauzule
- 3 Identyfikatory
- 4 Wyrażenia
- 5 Operatory
- 6 Predykaty



- 1 Zapytania/Instrukcje
- 2 Klauzule
- 3 Identyfikatory
- 4 Wyrażenia
- 5 Operatory
- 6 Predykaty

- 1 Zapytania/Instrukcje
- 2 Klauzule
- 3 Identyfikatory
- 4 Wyrażenia
- 5 Operatory
- 6 Predykaty



## Część IV

### Omówienie typów danych SQL

### Rodzaj dopuszczalnych danych

Każda kolumna tabeli posiada ograniczenie co do typu wpisywanych do niej danych (zapobiega to umieszczeniu danych tekstowych w kolumnie numerycznej). Ułatwia to również poprawne sortowanie i pełni ważną rolę w optymalizacji przestrzeni zajmowanej na dysku. Dlatego bardzo ważne jest staranne dobieranie typów kolumn

### Należy pamiętać !

Nie istnieją dwa identyczne SZBD

## Typy danych

Typy danych znacznie różnią się między poszczególnymi SZBD (systemami zarządzania bazą danych). Dochodzi nawet do tego iż typ o takiej samej nazwie ma różne znaczenia w SZBD różnych producentów. Szczegółów zawsze należy szukać w dokumentacji bazy danych.

### Tekstowe typy danych

Tekstowe typy danych nadają się do przechowywania informacji takich jak : nazwy, adresy, numery telefonów i kody pocztowe można podzielić je na typy o stałej długości i o zmiennej długości

- 1 CHAR – tekst o stałej długości od 1 do 255 znaków (rozmiar musi zostać określony w trakcie definiowania tablicy)
- 2 NCHAR – specjalna postać CHAR zaprojektowana dla znaków wielobajtowych lub unicode
- 3 NVARCHAR – specjalna postać typu TEXT dla znaków wielobajtowych lub unicode
- 4 TEXT – nazywany również: VARCHAR, LONG lub MEMO, tekst o zmiennej długości

### Tekstowe typy danych

Tekstowe typy danych nadają się do przechowywania informacji takich jak : nazwy, adresy, numery telefonów i kody pocztowe można podzielić je na typy o stałej długości i o zmiennej długości

- 1 CHAR – tekst o stałej długości od 1 do 255 znaków (rozmiar musi zostać określony w trakcie definiowania tablicy)
- 2 NCHAR – specjalna postać CHAR zaprojektowana dla znaków wielobajtowych lub unicode
- 3 NVARCHAR – specjalna postać typu TEXT dla znaków wielobajtowych lub unicode
- 4 TEXT – nazywany również: VARCHAR, LONG lub MEMO, tekst o zmiennej długości



### Tekstowe typy danych

Tekstowe typy danych nadają się do przechowywania informacji takich jak : nazwy, adresy, numery telefonów i kody pocztowe można podzielić je na typy o stałej długości i o zmiennej długości

- 1 CHAR – tekst o stałej długości od 1 do 255 znaków (rozmiar musi zostać określony w trakcie definiowania tablicy)
- 2 NCHAR – specjalna postać CHAR zaprojektowana dla znaków wielobajtowych lub unicode
- 3 NVARCHAR – specjalna postać typu TEXT dla znaków wielobajtowych lub unicode
- 4 TEXT – nazywany również: VARCHAR, LONG lub MEMO, tekst o zmiennej długości

### Tekstowe typy danych

Tekstowe typy danych nadają się do przechowywania informacji takich jak : nazwy, adresy, numery telefonów i kody pocztowe można podzielić je na typy o stałej długości i o zmiennej długości

- 1 CHAR – tekst o stałej długości od 1 do 255 znaków (rozmiar musi zostać określony w trakcie definiowania tablicy)
- 2 NCHAR – specjalna postać CHAR zaprojektowana dla znaków wielobajtowych lub unicode
- 3 NVARCHAR – specjalna postać typu TEXT dla znaków wielobajtowych lub unicode
- 4 TEXT – nazywany również: VARCHAR, LONG lub MEMO, tekst o zmiennej długości

## Numeryczne typy danych

### Numeryczne typy danych przechowują liczby

- 1 BIT – wartość jednobitowa (0 lub 1)
- 2 DECIMAL/NUMERIC - wartość stała- lub zmiennoprzecinkowa z różnym poziomem precyzji
- 3 FLOAT/ NUMBER – wartość zmiennoprzecinkowa
- 4 INT/ INTIGER – czterobajtowa wartość całkowita obsługująca liczby z przedziału od -2147483648 do 2147483647
- 5 REAL – czterobajtowa wartość zmiennoprzecinkowa
- 6 SMALLINT – dwubajtowa wartość całkowita obsługująca liczby z przedziału -32768 do 32767
- 7 TINYINT – jednobajtowa wartość całkowita obsługująca liczby z przedziału od 0 do 255

## Numeryczne typy danych

### Numeryczne typy danych przechowują liczby

- 1 BIT – wartość jednobitowa (0 lub 1)
- 2 DECIMAL/NUMERIC - wartość stała- lub zmiennoprzecinkowa z różnym poziomem precyzji
- 3 FLOAT/ NUMBER – wartość zmiennoprzecinkowa
- 4 INT/ INTIGER – czterobajtowa wartość całkowita obsługująca liczby z przedziału od -2147483648 do 2147483647
- 5 REAL – czterobajtowa wartość zmiennoprzecinkowa
- 6 SMALLINT – dwubajtowa wartość całkowita obsługująca liczby z przedziału -32768 do 32767
- 7 TINYINT – jednobajtowa wartość całkowita obsługująca liczby z przedziału od 0 do 255

## Numeryczne typy danych

Numeryczne typy danych przechowują liczby

- ❶ BIT – wartość jednobitowa (0 lub 1)
- ❷ DECIMAL/NUMERIC - wartość stała- lub zmiennoprzecinkowa z różnym poziomem precyzji
- ❸ FLOAT/ NUMBER – wartość zmiennoprzecinkowa
- ❹ INT/ INTIGER – czterobajtowa wartość całkowita obsługująca liczby z przedziału od -2147483648 do 2147483647
- ❺ REAL – czterobajtowa wartość zmiennoprzecinkowa
- ❻ SMALLINT – dwubajtowa wartość całkowita obsługująca liczby z przedziału -32768 do 32767
- ❼ TINYINT – jednobajtowa wartość całkowita obsługująca liczby z przedziału od 0 do 255

## Numeryczne typy danych

Numeryczne typy danych przechowują liczby

- ❶ BIT – wartość jednobitowa (0 lub 1)
- ❷ DECIMAL/NUMERIC - wartość stała- lub zmiennoprzecinkowa z różnym poziomem precyzji
- ❸ FLOAT/ NUMBER – wartość zmiennoprzecinkowa
- ❹ INT/ INTIGER – czterobajtowa wartość całkowita obsługująca liczby z przedziału od -2147483648 do 2147483647
- ❺ REAL – czterobajtowa wartość zmiennoprzecinkowa
- ❻ SMALLINT – dwubajtowa wartość całkowita obsługująca liczby z przedziału -32768 do 32767
- ❼ TINYINT – jednobajtowa wartość całkowita obsługująca liczby z przedziału od 0 do 255

## Numeryczne typy danych

Numeryczne typy danych przechowują liczby

- ❶ BIT – wartość jednobitowa (0 lub 1)
- ❷ DECIMAL/NUMERIC - wartość stała- lub zmiennoprzecinkowa z różnym poziomem precyzji
- ❸ FLOAT/ NUMBER – wartość zmiennoprzecinkowa
- ❹ INT/ INTIGER – czterobajtowa wartość całkowita obsługująca liczby z przedziału od -2147483648 do 2147483647
- ❺ REAL – czterobajtowa wartość zmiennoprzecinkowa
- ❻ SMALLINT – dwubajtowa wartość całkowita obsługująca liczby z przedziału -32768 do 32767
- ❼ TINYINT – jednobajtowa wartość całkowita obsługująca liczby z przedziału od 0 do 255

## Numeryczne typy danych

Numeryczne typy danych przechowują liczby

- ❶ BIT – wartość jednobitowa (0 lub 1)
- ❷ DECIMAL/NUMERIC - wartość stała- lub zmiennoprzecinkowa z różnym poziomem precyzji
- ❸ FLOAT/ NUMBER – wartość zmiennoprzecinkowa
- ❹ INT/ INTIGER – czterobajtowa wartość całkowita obsługująca liczby z przedziału od -2147483648 do 2147483647
- ❺ REAL – czterobajtowa wartość zmiennoprzecinkowa
- ❻ SMALLINT – dwubajtowa wartość całkowita obsługująca liczby z przedziału -32768 do 32767
- ❼ TINYINT – jednobajtowa wartość całkowita obsługująca liczby z przedziału od 0 do 255



## Numeryczne typy danych

Numeryczne typy danych przechowują liczby

- ❶ BIT – wartość jednobitowa (0 lub 1)
- ❷ DECIMAL/NUMERIC - wartość stała- lub zmiennoprzecinkowa z różnym poziomem precyzji
- ❸ FLOAT/ NUMBER – wartość zmiennoprzecinkowa
- ❹ INT/ INTIGER – czterobajtowa wartość całkowita obsługująca liczby z przedziału od -2147483648 do 2147483647
- ❺ REAL – czterobajtowa wartość zmiennoprzecinkowa
- ❻ SMALLINT – dwubajtowa wartość całkowita obsługująca liczby z przedziału -32768 do 32767
- ❼ TINYINT – jednobajtowa wartość całkowita obsługująca liczby z przedziału od 0 do 255

## Typy danych daty i czasu

- 1 DATE – wartość daty
- 2 DATETIME/ TIMESTAMP – wartość daty i czasu
- 3 SMALLDATETIME – wartość daty i czasu z dokładnością co do minuty
- 4 TIME – wartość czasu

## Typy danych daty i czasu

- 1 DATE – wartość daty
- 2 DATETIME/ TIMESTAMP – wartość daty i czasu
- 3 SMALLDATETIME – wartość daty i czasu z dokładnością co do minuty
- 4 TIME – wartość czasu

## Typy danych daty i czasu

- 1 DATE – wartość daty
- 2 DATETIME/ TIMESTAMP – wartość daty i czasu
- 3 SMALLDATETIME – wartość daty i czasu z dokładnością co do minuty
- 4 TIME – wartość czasu

### Typy danych daty i czasu

- 1 DATE – wartość daty
- 2 DATETIME/ TIMESTAMP – wartość daty i czasu
- 3 SMALLDATETIME – wartość daty i czasu z dokładnością co do minuty
- 4 TIME – wartość czasu

### Binarne typy danych

(dowolne dane takie jak grafika, multimedia czy dokumenty) są one najmniej przenośne między SZBD i nie korzysta się z nich często

- 1 BINARY – dane binarne o stałej długości od 255 lub 8000 bajtów w zależności od implementacji
- 2 VARBINARY – dane binarne o zmiennej długości od 255 do 8000 bajtów w zależności od implementacji
- 3 RAW – dane binarne o stałej długości maksymalnie do 255 bajtów
- 4 LONG RAW – dane binarne o zmiennej długości maksymalnie do *2GB*

### Binarne typy danych

(dowolne dane takie jak grafika, multimedia czy dokumenty) są one najmniej przenośne między SZBD i nie korzysta się z nich często

- ❶ BINARY – dane binarne o stałej długości od 255 lub 8000 bajtów w zależności od implementacji
- ❷ VARBINARY – dane binarne o zmiennej długości od 255 do 8000 bajtów w zależności od implementacji
- ❸ RAW – dane binarne o stałej długości maksymalnie do 255 bajtów
- ❹ LONG RAW – dane binarne o zmiennej długości maksymalnie do *2GB*

### Binarne typy danych

(dowolne dane takie jak grafika, multimedia czy dokumenty) są one najmniej przenośne między SZBD i nie korzysta się z nich często

- ❶ BINARY – dane binarne o stałej długości od 255 lub 8000 bajtów w zależności od implementacji
- ❷ VARBINARY – dane binarne o zmiennej długości od 255 do 8000 bajtów w zależności od implementacji
- ❸ RAW – dane binarne o stałej długości maksymalnie do 255 bajtów
- ❹ LONG RAW – dane binarne o zmiennej długości maksymalnie do *2GB*



### Binarne typy danych

(dowolne dane takie jak grafika, multimedia czy dokumenty) są one najmniej przenośne między SZBD i nie korzysta się z nich często

- 1 BINARY – dane binarne o stałej długości od 255 lub 8000 bajtów w zależności od implementacji
- 2 VARBINARY – dane binarne o zmiennej długości od 255 do 8000 bajtów w zależności od implementacji
- 3 RAW – dane binarne o stałej długości maksymalnie do 255 bajtów
- 4 LONG RAW – dane binarne o zmiennej długości maksymalnie do *2GB*

### Table: produkty

#### Columns:

prod_id	char(10)
dost_id	char(10)
prod_nazwa	char(255)
prod_cena	decimal(8,2)
prod_opis	text

## Część V

Tworzenie, usuwanie i modyfikacja schematu tabel i danych

# CREATE

Query x

Limit to 1000 rows

```
1 CREATE TABLE czytelnicy ( IDCzytelnika int,  
2                               Nazwisko VARCHAR(200),  
3                               Imie VARCHAR(200),  
4                               PRIMARY KEY(IDCzytelnika));
```

**SCHEMAS**

Filter objects

▼ biblioteka

▼ Tables

▼ czytelnicy

- Columns
- Indexes
- Foreign Keys
- Triggers

Views

Stored Procedures

Functions

**Table: czytelnicy**

**Columns:**

<u>IDCzytelnika</u>	int PK
Nazwisko	varchar(200)
Imie	varchar(200)

# CREATE

Query 1 x

Limit to 1000 rows

```
1 CREATE TABLE ksiazki (sygnatura smallint PRIMARY KEY,  
2 autor VARCHAR(100),  
3 tytul VARCHAR(100),  
4 rodzaj_literacki VARCHAR(50));
```

**SCHEMAS**

Filter objects

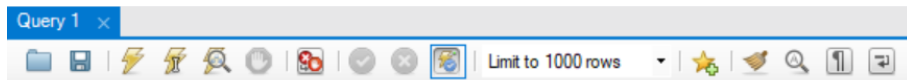
- biblioteka
  - Tables
    - czytelnicy
    - ksiazki
      - Columns
      - Indexes
      - Foreign Keys
      - Triggers
  - Views
  - Stored Procedures
  - Functions

**Table: ksiazki**

**Columns:**

<u>sygnatura</u>	smallint PK
autor	varchar(100)
tytul	varchar(100)
rodzaj_literacki	varchar(50)

# INSERT



```
1 • INSERT INTO czytelnicy(IDCzytelnika, Nazwisko, Imie)
2   VALUES ( 1, 'Kowal', 'Piotr'),
3           ( 2, 'Czajka', 'Karolina'),
4           ( 3, 'Kot', 'Tomasz');
```

# INSERT

Query 1

Limit to 1000 rows

```
1 • INSERT INTO ksiazki
2   VALUES (1,'H. Sienkiewicz','Potop','powieść historyczna'),
3           (2, 'J. Słowacki', 'Balladyna', 'dramat'),
4           (3, 'A. Mickiewicz', 'Pan Tadeusz', NULL),
5           (4, NULL, 'Ferdynand', 'powieść awangardowa');
```

	sygnatura	autor	tytul	rodzaj_literacki
▶	1	H. Sienkiewicz	Potop	powieść historyczna
	2	J. Słowacki	Balladyna	dramat
	3	A. Mickiewicz	Pan Tadeusz	NULL
	4	NULL	Ferdynand	powieść awangardowa
*	NULL	NULL	NULL	NULL

# SELECT

Query x

Limit to 1000 rows















1 • SELECT \* FROM czytelnicy;

	IDCzytelnika	Nazwisko	Imie
▶	1	Kowal	Piotr
	2	Czajka	Karolina
	3	Kot	Tomasz
●	NULL	NULL	NULL



# SELECT

Query 1 ×

          | Limit to 1000 rows ▼     

```
1 • SELECT * FROM ksiazki
2   WHERE autor IN ('J. Słowacki', 'H. Sienkiewicz');
```

	sygnatura	autor	tytuł	rodzaj_literacki
▶	1	H. Sienkiewicz	Potop	powieść historyczna
	2	J. Słowacki	Balladyna	dramat
•	NULL	NULL	NULL	NULL

# UPDATE

Query 1 x



```
1 • UPDATE czytelnicy
2   SET Nazwisko = 'Wróbel'
3   WHERE IDCzytelnika = 2;
```

	IDCzytelnika	Nazwisko	Imie
▶	1	Kowal	Piotr
	2	Wróbel	Karolina
	3	Kot	Tomasz
✱	NULL	NULL	NULL

# UPDATE



```
1 • UPDATE książki
2   SET autor = 'W. Gombrowicz'
3   WHERE sygnatura > 3;
```

# DELETE

Query x

Limit to 1000 rows

```
1 • DELETE FROM czytelnicy
2   WHERE IDCzytelnika < 2;
```

	IDCzytelnika	Nazwisko	Imie
▶	2	Wróbel	Karolina
	3	Kot	Tomasz
•	NULL	NULL	NULL

# DELETE

Query 1 ×

Limit to 1000 rows ▾

```
1 • DELETE FROM ksiazki
2 WHERE sygnatura BETWEEN 2 AND 4;
```

	sygnatura	autor	tytul	rodzaj_literacki
▶	1	H. Sienkiewicz	Potop	powieść historyczna
●	NULL	NULL	NULL	NULL

# ALTER

```
Query x
1 • ALTER TABLE czytelnicy
2   ADD data_urodzenia date;
3
4 • ALTER TABLE czytelnicy
5   ADD adres VARCHAR(50);
```

Table: czytelnicy

Columns:

<u>IDCzytelnika</u>	int PK
Nazwisko	varchar(200)
Imie	varchar(200)
data_urodzenia	date
adres	varchar(50)

# ALTER



```
1 • ALTER TABLE wydawnictwa
2 ADD (CONSTRAINT ID PRIMARY KEY(ID_Wydawnictwa));
```

Table: **wydawnictwa**

Columns:

<b>ID_Wydawnictwa</b>	tinyint PK
Nazwa_wydawnictwa	varchar(100)
Kod_pocztowy	varchar(5)
miasto	varchar(50)
ulica	varchar(50)

# DROP

Query x

Limit to 1000 rows

1 • DROP TABLE czytelnicy;

Action Output

#	Time	Action	Message	Duration / Fetch
1	10:28:49	DROP TABLE czytelnicy	0 row(s) affected	0.016 sec



## Część VI

### Zakładanie i usuwanie indeksów

## Co to są indeksy i do czego służą?

<i><b>Zapytanie</b></i>	<i><b>Czas (ms)</b></i>	<i><b>Data Output</b></i>
SELECT * FROM person;	3562	10000000000
SELECT * FROM person WHERE last_name = 'Smith';	4261	100520
SELECT * FROM person WHERE first_name = 'Emma';	4066	49767
SELECT * FROM person WHERE first_name = 'Julie' AND last_name = 'Andrews';	514	46

## Co to są indeksy i do czego służą?

<i>Zapytanie</i>	<i>Czas (ms)</i>	<i>INDEKS</i>
SELECT * FROM person WHERE first_name = 'Emma';	508	CREATE INDEX person_first_name_idx ON person (first_name);
SELECT * FROM person WHERE first_name = 'Julie' AND last_name = 'Andrews';	26	CREATE INDEX person_first_name_last_name_idx ON person (first_name, last_name);

# Zakładanie indeksów

Query x

Limit to 1000 rows

```
1 • CREATE INDEX identyfikator ON czytelnicy(IDCzytelnika);
```

**SCHEMAS**

Filter objects

- biblioteka
  - Tables
    - czytelnicy
      - Columns
      - Indexes
        - PRIMARY
        - identyfikator**
      - Foreign Keys
      - Triggers
    - Views
    - Stored Procedures
    - Functions

**Index: identyfikator**

**Definition:**

Type	BTREE
Unique	No
Visible	Yes
Columns	IDCzytelnika

# Zakładanie unikalnych indeksów

Query x

Limit to 1000 rows

```
1 • CREATE UNIQUE INDEX dane_osobowe ON czytelnicy(Nazwisko, Imie);
```

**SCHEMAS**

Filter objects

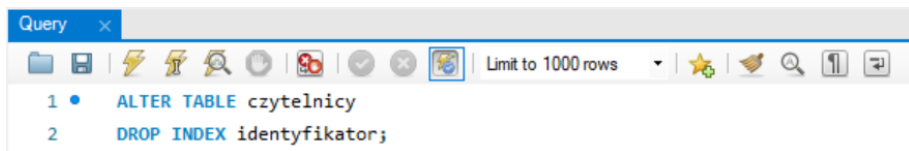
- biblioteka
  - Tables
    - czytelnicy
      - Columns
      - Indexes
        - PRIMARY
        - dane\_osobowe
        - identyfikator
      - Foreign Keys
      - Triggers
    - Views
    - Stored Procedures
    - Functions

**Index: dane\_osobowe**

**Definition:**

Type	BTREE
Unique	Yes
Visible	Yes
Columns	Nazwisko Imie

## Usuwanie indeksów



## Część VII

### Omówienie więzów spójności

### Więzy spójności

dzięki więzom spójności nie można tak zmodyfikować danych by straciły one spójność. Są zbiorem zasad nałożonych na tabele w bazie danych



1 Primary Key

2 Unique

3 NOT NULL

4 Check

5 Foreign key

6 DEFAULT

7 INDEX

- 1 Primary Key
- 2 Unique
- 3 NOT NULL
- 4 Check
- 5 Foreign key
- 6 DEFAULT
- 7 INDEX

- 1 Primary Key
- 2 Unique
- 3 NOT NULL
- 4 Check
- 5 Foreign key
- 6 DEFAULT
- 7 INDEX

- 1 Primary Key
- 2 Unique
- 3 NOT NULL
- 4 Check
- 5 Foreign key
- 6 DEFAULT
- 7 INDEX

- 1 Primary Key
- 2 Unique
- 3 NOT NULL
- 4 Check
- 5 Foreign key
- 6 DEFAULT
- 7 INDEX

- 1 Primary Key
- 2 Unique
- 3 NOT NULL
- 4 Check
- 5 Foreign key
- 6 DEFAULT
- 7 INDEX

- 1 Primary Key
- 2 Unique
- 3 NOT NULL
- 4 Check
- 5 Foreign key
- 6 DEFAULT
- 7 INDEX

## Przykład więzów spójności

```
CREATE TABLE Pracownicy(  
    ID_prac NUMERIC(7,0) PRIMARY KEY,  
    pesel NUMERIC(11,0) UNIQUE NOT NULL,  
    Imię VARCHAR(25) NOT NULL,  
    Nazwisko VARCHAR(25) NOT NULL,  
    Szef NUMERIC(7,0)  
        CONSTRAINT Szef_ko REFERENCES Pracownicy,  
    Nazwa_działu VARCHAR(35),  
    Miejsce VARCHAR(35),  
    Zarobki NUMERIC(2,8),  
    Premia NUMERIC(8,2),  
    CONSTRAINT Dz_ko FOREIGN KEY (Nazwa_działu, Miejsce)  
        REFERENCES Działy (Nazwa, Miasto),  
    CONSTRAINT P_ck CHECK(0.1*Zarobki <= Premia AND Premia <=0.5*Zarobki)  
);
```



## Część VIII

### Omówienie zatwierdzania zmian w bazie

### Commit

Commit – ostateczne zatwierdzenie tymczasowo dokonanych zmian. Najpopularniejszym użyciem jest zakończenie transakcji.

### ROLLBACK

W czasie trwania transakcji, w przeciwieństwie do commit poleceniem służącym do anulowania niezatwierdzonych zmian jest ROLLBACK

Commit używany jest też w systemach kontroli wersji. Oznacza wtedy zatwierdzenie tymczasowo wprowadzonych zmian w kodzie źródłowym i wprowadzenie ich do systemu.

## Część IX

### Literatura



Sebastian Lachecinski

Składowanie i przetwarzanie danych temporalnych w świetle wymagań standardu języka SQL ISO

<https://www.researchgate.net/publication/344692829>



Konrad Zdanowski

Bazy danych – wykład ósmy Indeksy

[https://www.impan.pl/~kz/DB/KZ\\_BD\\_w08.pdf](https://www.impan.pl/~kz/DB/KZ_BD_w08.pdf)



Ben Forta

SQL Server i T-SQL w mgnieniu oka

Wydawnictwo Helion



Internet

<http://dev.cdur.pl/Artykuly/Zakladanie-indeksu-w-SQL-Server>

<https://jsystems.pl/blog/artykul.html?id=427>



SQL Index — Indexes in SQL — Database Index

<https://www.youtube.com/watch?v=fsG1XaZEa78&t=65s>

# Dziękujemy za uwagę!

Karina Burnicka   Viktoria Ługowska

