

Bazy danych - SQL

Składowe SZBD

- Jądro SZBD realizuje podstawowe funkcje związane z przechowywaniem danych, kontrolą współbieżności itp.
- Interfejs zawiera na ogół pewien język baz danych, np. SQL
- Zbiór dodatkowych narzędzi oprogramowania do tworzenia systemów informacyjnych

Języki baz danych

- język definicji danych (ang. Data Definition Language, DDL)
- języki selektywnego dostępu do danych – języki zapytań
- języki operowania na danych (ang. Data Manipulation Language, DML)

Języki zapytań

- Interfejsy typu zapytanie przez przykład (ang. Query by Example - QBE), szablony (formularze, strony WWW)
- Structured Query Language (SQL), języki algebraiczne
- języki predykatowe (o zmiennych atrybutowych i krotkowych)
- DATALOG (język zbliżony do PROLOGu ale nieproceduralny i bez termów)

SQL – historia

- 1974: Chamberlain, IBM, San Jose – Structured English Query Language SEQEL
- koniec lat 70-tych: ORACLE (Relational Software Inc.) – pierwsza implementacja komercyjna
- 1982: ANSI* – RDL (Relational Data Language)
- 1983: ISO** – definicja SQL
- 1986: ANSI – pierwszy standard SQL (SQL-86)

*American National Standards Committee

**International Standards Organisation

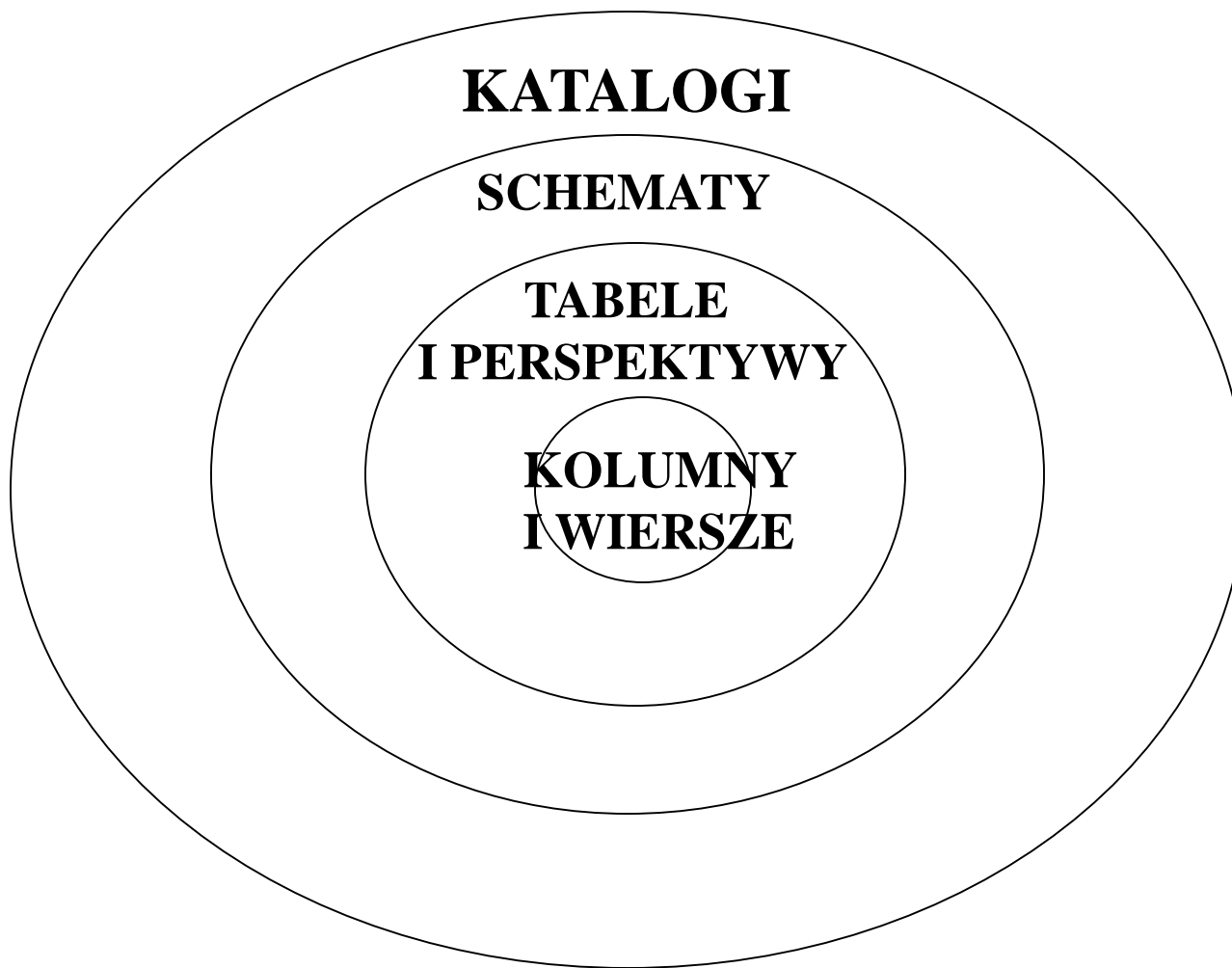
SQL – historia ciąg dalszy

- 1987: ISO – pierwszy standard SQL (ISO 9075)
- 1989: ISO – następny standard SQL (ISO 9076 – SQL-89)
- 1992: ISO – wzbogacona wersja ISO 9075 (SQL 2)
- 1999: ANSI/ISO – ostatnia wersja ISO 9075 (SQL-99 lub SQL 3)
- 2003: ANSI/ISO – SQL2011/2011

Środowisko SQL

- Interaktywny SQL – użytkownik wpisuje polecenie i wysyła bezpośrednio do interpretatora zapytań, który wykonuje odpowiednie działania w jądrze SZBD i ewentualnie zwraca wirtualną tabelę z odpowiedzią
- Osadzony SQL – jest nakładką na język proceduralny, którego rozkazy mogą uruchamiać w sposób statyczny lub dynamiczny zapytania

Hierarchia obiektów w SQL 2



Schematy

- Tworzenie schematu

```
CREATE SCHEMA nazwa_schematu  
AUTHORIZATION ID_wlasciciela
```

```
CREATE SCHEMA magazyn AUTHORIZATION dbo
```

- Określanie używanego schematu

```
SET SCHEMA nazwa_schematu
```

```
SET SCHEMA magazyn
```

Tworzenie bazy danych w MS SQL

```
create database Dydaktyka
```

```
Use Dydaktyka
```

Domeny

- Standard SQL 2 dopuszcza tworzenie własnych zbiorów dopuszczalnych wartości pewnych kolumn w tabelach (dziedzin atrybutów)

```
CREATE DOMAIN nazwa_domeny AS typ_danych  
DEFAULT wartosc_domyslna  
CHECK warunek_kontrolny
```

```
CREATE DOMAIN kontrola AS CHAR(1)  
DEFAULT 'T'  
CHECK (UPPER(VALUE) = 'T' OR UPPER(VALUE) = 'N' )
```

Tabele

- Tworzenie tabeli trwałych

```
CREATE TABLE nazwa_tabeli (  
    nazwa_kolumny1 typ_danych1  
    DEFAULT wartosc_domyslna1  
    CHECK warunek_kontrolny1,  
    nazwa_kolumny1 typ_danych1  
    DEFAULT wartosc_domyslna1  
    CHECK warunek_kontrolny1,  
    ..... ) ;
```

Tworzenie tabeli w MS SQL

```
Create table [Dydaktyka]
(
    [Id] Bigint Identity NOT NULL,
    [Test] Char(10) NULL,
    Primary Key ([Id])
)
```

Typy danych

- Teksty o stałej lub zmiennej długości: CHAR(n) – tekst o ustalonej długości n , VARCHAR(n) – tekst o zmiennej długości nie większej niż n ;
- Ciągi binarne o stałej lub zmiennej długości: BIT(n) – ciąg bitów o ustalonej długości n , BIT VARYING(n) – ciąg bitów o długości co najwyżej n
- Typowe wartości całkowite INT (INTEGER) lub SHORTINT (SMALLINT)

Typy danych

- Liczby zmiennopozycyjne: REAL - liczba zapisana w postaci wykładniczej o pojedynczej precyzji, FLOAT – podobna do REAL ale, w niektórych implementacjach można określić precyzję, DOUBLE PRECISION – jak REAL ale zwiększona precyzja
- Liczby stałopozycyjne: NUMERIC(n,m) – liczba składająca się z n znaków w tym m po przecinku, DECIMAL(n,m) – podobny do NUMERIC ale dopuszczający większą dokładność

Typy danych

- Daty i czas: DATE – data zapisana w formacie określonym przez implementację, TIME – czas zapisany w dostępnym standardzie, TIMESTAMP – połączenie daty i czasu
- Inne typy specyficzne dla implementacji: np. BOOL w MySQL czy INTERVAL YEAR TO MONTH (n) w Oracle 9i

Obsługa wartości NULL

- wartość NULL nie może być umieszczona w kolumnie NOT NULL,
- porównywanie dwóch kolumn zawierających NULL jest nieskuteczne (wartości NULL można identyfikować w klauzuli WHERE przy użyciu wyrażeń IS NULL IS NOT NULL)
- kolumna zawierająca NULL jest ignorowana podczas obliczania wartości agregujących natomiast jest uwzględniana w klauzuli GROUP BY
- jeżeli w warunku złączenia pojawi się kolumna z wartościami NULL to złączenie traktowane jest jako zewnętrzne

Ograniczenia

- Ograniczenie można zadać poprzez zdefiniowanie warunku logicznego, w tym także takiego, które sięga do innych tabel lub poprzez standardowego ograniczenia: NOT NULL lub UNIQUE

```
CREATE TABLE towar (  
  symbol_towaru CHAR(5)  
  NOT NULL  
  UNIQUE  
  DEFAULT 'SSSSS'  
  CHECK (SUBSTRING(VALUE, FROM 1 FOR 1) = 'S'),  
  . . . .
```

Ograniczenia – MS SQL

```
CREATE TABLE [dbo].[Studenci](  
    [StudentId] [bigint] IDENTITY(1,1) NOT FOR REPLICATION NOT NULL,  
    [Nazwisko] [nvarchar](100) NULL,  
    [Imie] [nvarchar](100) NULL,  
    [PESEL] [nvarchar](11) NULL,  
    [Kod] [nvarchar](6) NULL,  
    CONSTRAINT [PK_Studenci] PRIMARY KEY CLUSTERED  
    (  
        [StudentId] ASC  
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS  
    = ON) ON [PRIMARY]  
 ) ON [PRIMARY]
```

Ograniczenia – MS SQL

```
ALTER TABLE [dbo].[Studenci] WITH CHECK ADD  
CONSTRAINT [CK_Studenci] CHECK ((([Kod] like '[0-9][0-9]-[0-9][0-9][0-9]'))
```

```
ALTER TABLE [dbo].[Studenci] CHECK CONSTRAINT  
[CK_Studenci]
```

Indeksy

- Indeks jest strukturą danych umożliwiającą szybki dostęp do krotek pewnej tabeli według jednej lub kilku kolumn
- Indeks zawiera kopie wybranych wartości kolumn ze związanej tabeli uszeregowane, tak by łatwiej było ją przeszukiwać

```
CREATE [UNIQUE] INDEX nazwa_indeksu  
ON nazwa_tabeli (nazwy_kolumn_klucza)
```

```
CREATE UNIQUE INDEX symbol_nazwa_towaru  
ON towar (symbol_towaru, nazwa_towaru)
```

Indeksy

- Czas trwania prostego wyszukiwania w tabeli zawierającej 161 016 rekordy z indeksem i bez indeksu

```
SQL Server parse and compile time:
```

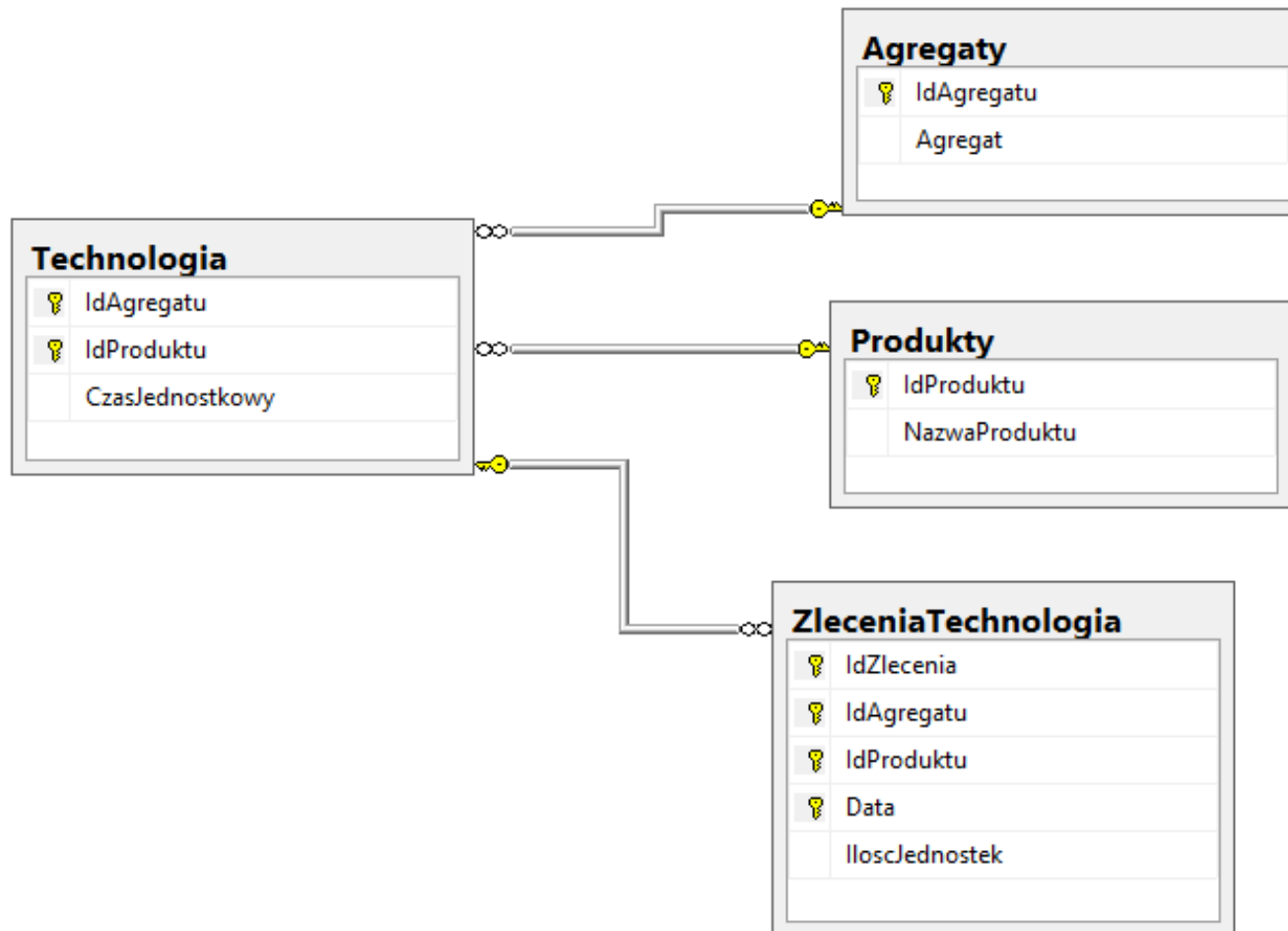
```
  CPU time = 0 ms, elapsed time = 1 ms.
```

```
(1 row(s) affected)
```

```
SQL Server Execution Times:
```

```
  CPU time = 47 ms,  elapsed time = 39 ms.
```

Indeksy – przykład



Indeksy - dane

IdAgregatu	Agregat	IdProduktu	NazwaProduktu
1	Prasa	1	Wałek 01 ...
2	Tokarka	2	Wałek 02 ...

IdAgregatu	IdProduktu	CzasJednostko...
1	1	10,00
2	1	12,00
1	2	15,00
2	2	17,00

IdZlecenia	IdAgregatu	IdProduktu	Data	IloscJednostek
121	1	1	2016-03-21	5
121	1	2	2016-03-21	8
120	2	1	2016-03-21	3

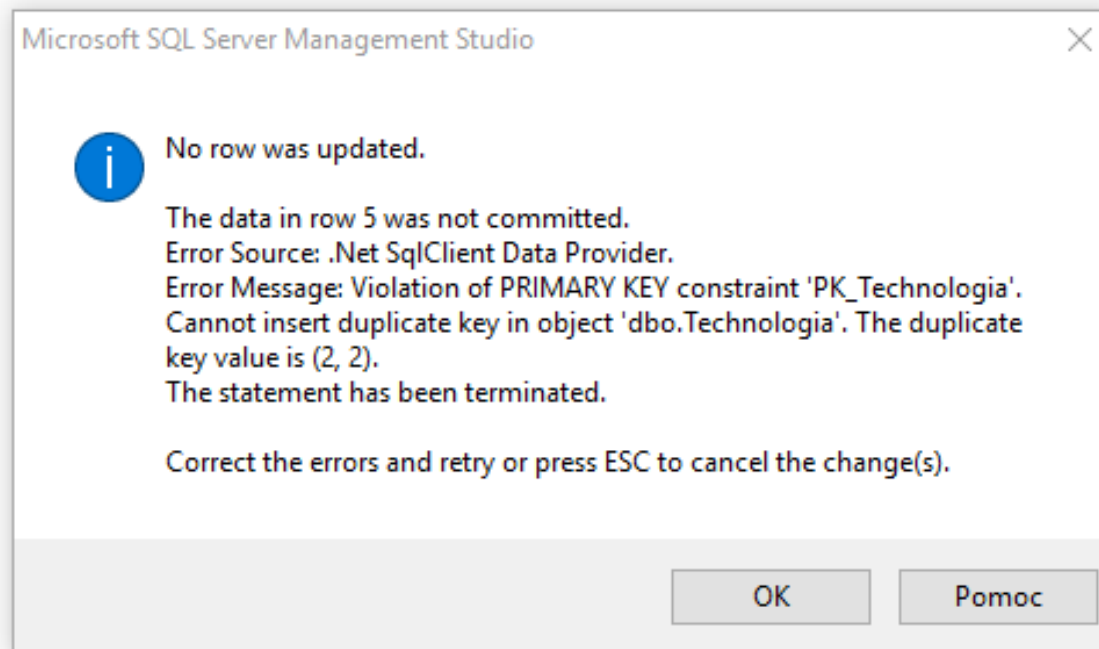
Indeksy - złączenie

```
SELECT TOP (200) ZleceniaTechnologia.IdZlecenia, ZleceniaTechnologia.Data,
Agregaty.Agregat, Produkty.NazwaProduktu, Technologia.CzasJednostkowy,
ZleceniaTechnologia.IloscJednostek,
Technologia.CzasJednostkowy * ZleceniaTechnologia.IloscJednostek AS
Obciazenie
FROM Agregaty INNER JOIN
Technologia ON Agregaty.IdAgregatu = Technologia.IdAgregatu INNER JOIN
Produkty ON Technologia.IdProduktu = Produkty.IdProduktu INNER JOIN
ZleceniaTechnologia ON Technologia.IdAgregatu =
ZleceniaTechnologia.IdAgregatu AND Technologia.IdProduktu =
ZleceniaTechnologia.IdProduktu
```

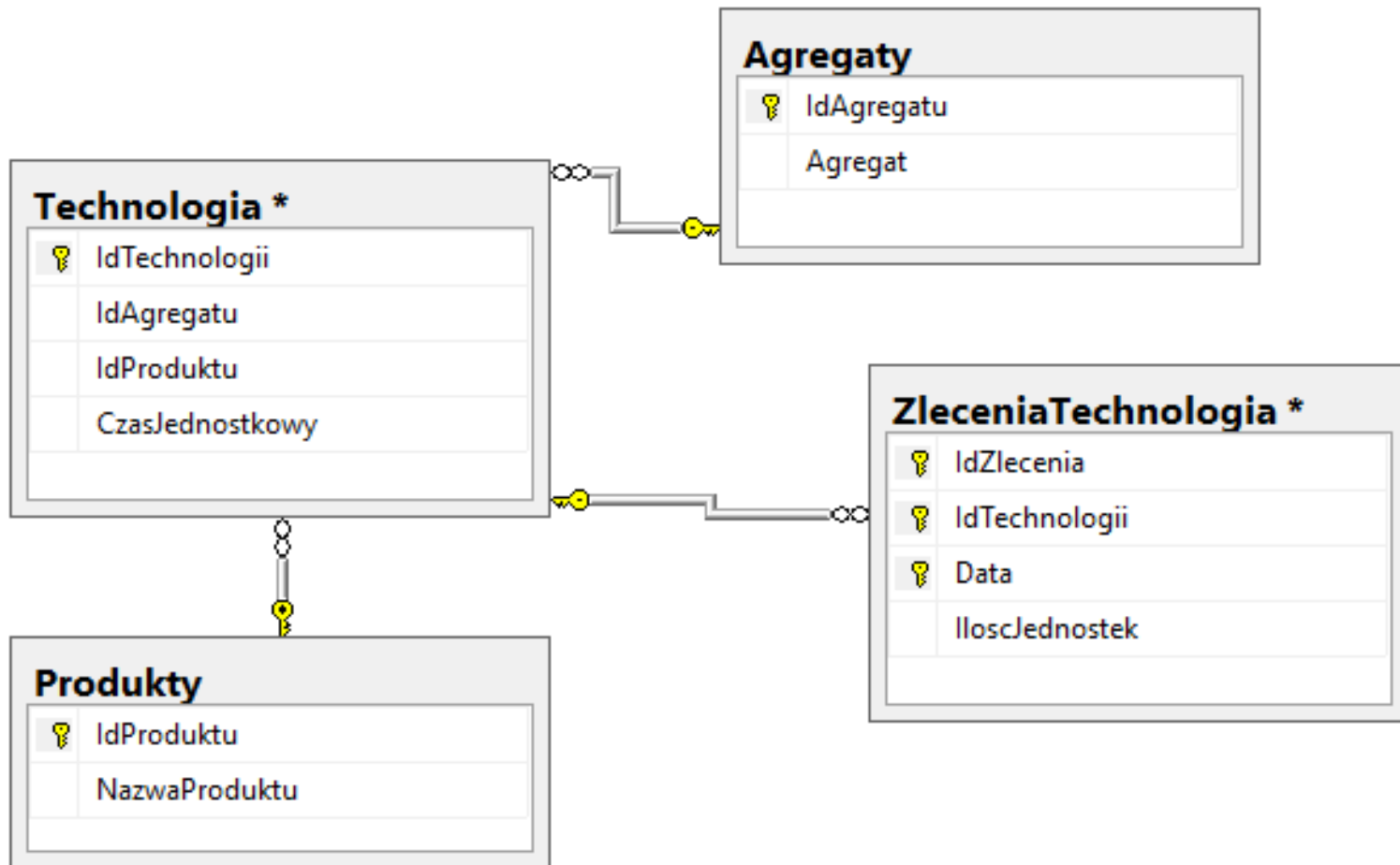
IdZlecenia	Data	Agregat	NazwaProduktu	CzasJednostko...	IloscJednostek	Obciazenie
120	2016-03-21	Tokarka	... Wałek 01	... 12,00	3	36,0000
121	2016-03-21	Prasa	... Wałek 01	... 10,00	5	50,0000
121	2016-03-21	Prasa	... Wałek 02	... 15,00	8	120,0000

Klucz złożony - zabezpieczenie

	IdAgregatu	IdProduktu	CzasJednostko...
	1	1	10,00
	1	2	15,00
	2	1	12,00
	2	2	17,00
✎	2	2	15
*	NULL	NULL	NULL

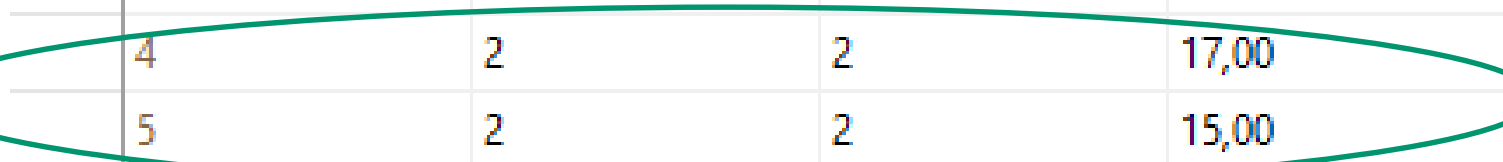


Uproszczenie struktury – klucz sztuczny



Klucz sztuczny – zagrożenie

	IdTechnologii	IdAgregatu	IdProduktu	CzasJednostko...
	1	1	1	10,00
	2	1	2	15,00
	3	2	1	12,00
	4	2	2	17,00
	5	2	2	15,00

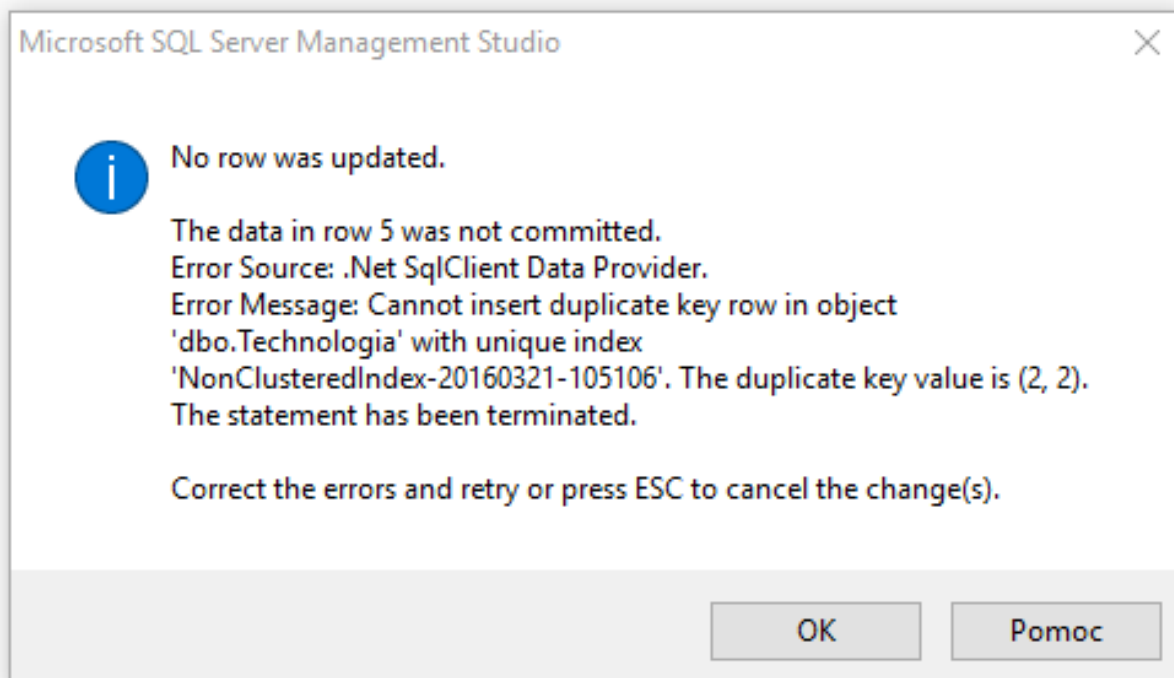


Tworzenie unikalnego indeksu

```
CREATE UNIQUE NONCLUSTERED INDEX  
[NonClusteredIndex-20160321-105106] ON  
[dbo].[Technologia]  
(  
[IdAgregatu] ASC,  
[IdProduktu] ASC  
)
```

Indeks - zabezpieczenie

IdTechnologii	IdAgregatu	IdProduktu	CzasJednostko...
1	1	1	10,00
2	1	2	15,00
3	2	1	12,00
4	2	2	17,00
NULL	2	2	15
NULL	NULL	NULL	NULL



Zmiany schematów relacji

- Instrukcja składa się z dwóch części ALTER TABLE i definicji działania ADD (dodaj) i DROP (usuń)

```
ALTER TABLE towar DROP jednostka_miary
```

```
ALTER TABLE towar ADD opakowanie  
VARCHAR(20)
```

```
DROP opakowania
```

```
DROP INDEX symbol_nazwa_towaru
```

Proste zapytania

- Odpowiednikiem operacji selekcji jest instrukcja **SELECT** wraz z słowami kluczowymi **FROM** i **WHERE**

```
SELECT kolumna1, kolumna2, ...
```

```
FROM tabela1, tabela2, ...
```

```
WHERE kryteria wyboru
```


Przykładowa tabela – dowody wydania Rw

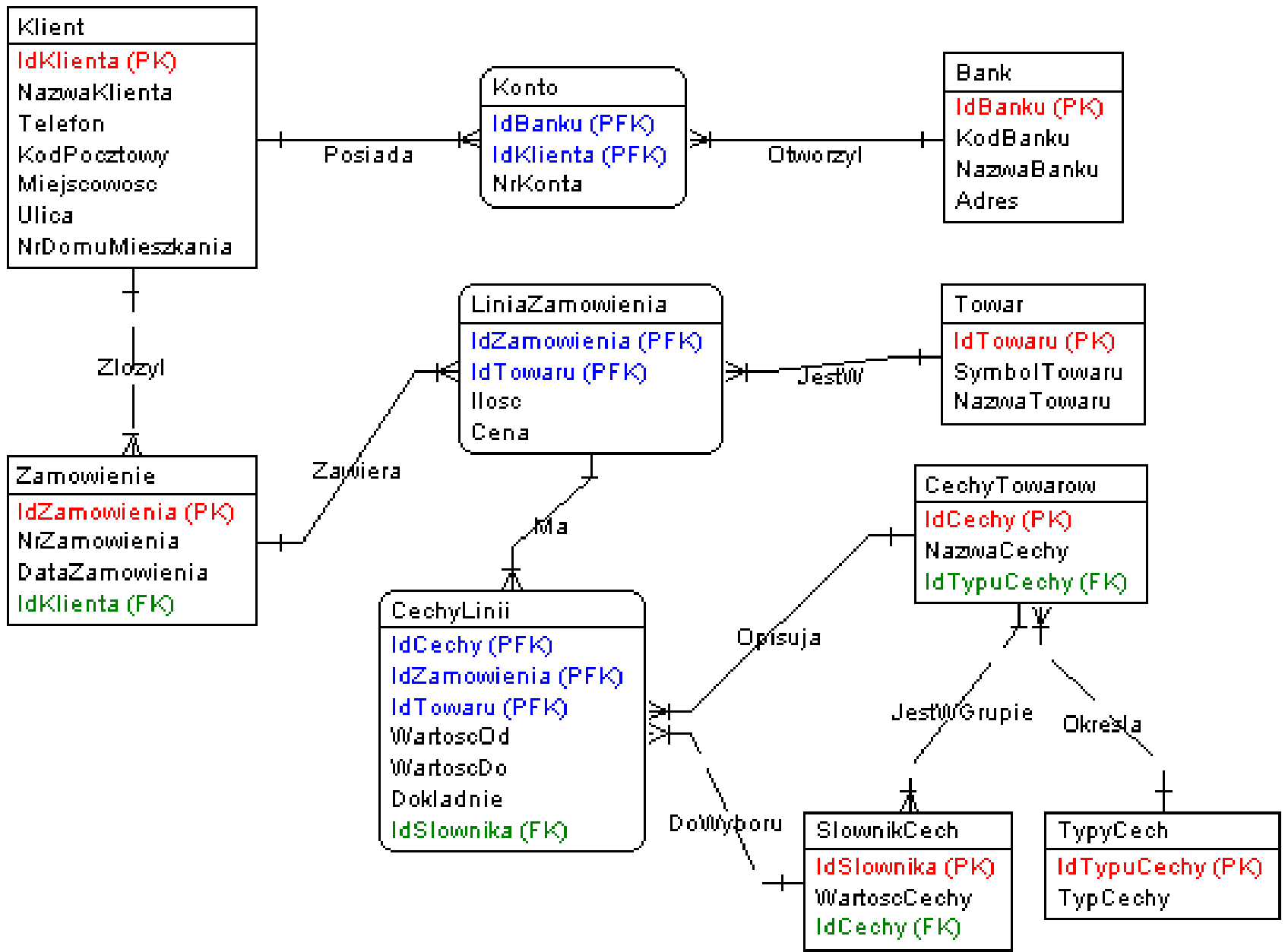
Dowody Wydania

SymbolTowaru	NazwaTowaru	Magazyn	Od-biorca	Data	Ilosc	J.m.
S0001	rura fi 0,63 gr 0,2	MWG01	HS	16.04.04	12	mb
S0025	rura fi 1,26 gr 0,3	MWG02	PP	12.04.04	80	mb
S0001	rura fi 0,63 gr 0,2	MWG01	PP	12.04.04	15	mb
S1025	złączka	MOZ01	ZP	03.04.04	100	szt.
S0025	rura fi 1,26 gr 0,3	MOZ01	ZP	03.04.04	20	mb
S0152	rura kw. 2 gr 0,2	MOP02	TT	01.04.04	1	mb

Przykład zapytania

```
SELECT *  
FROM DowodyWydania  
WHERE Magazyn = 'MWG01'
```

SymbolTowaru	NazwaTowaru	Magazyn	Od-biorca	Data	Ilosc	J.m.
S0001	rura fi 0,63 gr 0,2	MWG01	HS	16.04.04	12	mb
S0001	rura fi 0,63 gr 0,2	MWG01	PP	12.04.04	15	mb



Przykładowa tabela – towary

IdTowaru	SymbolTowaru	NazwaTowaru
1	RZ001	Rura zgrz. fi 6,3 gr 0,2
2	RZ002	Rura zgrz. fi 12,6 gr 0,2
3	RZ003	Rura zgrz. fi 6,3 gr 0,3
4	RZ004	Rura zgrz. fi 12,6 gr 0,3
5	RZ011	Rura zgrz. kw 4 gr 0,2
6	RZ012	Rura zgrz. kw 5 gr 0,3
7	ZL001	Złączka 1'
8	ZL002	Złączka 2'

Przykład zapytania

```
SELECT *  
FROM `towar`  
WHERE SymbolTowaru LIKE 'R%'
```

IdTowaru	SymbolTowaru	NazwaTowaru
1	RZ001	Rura zgrz. fi 6,3 gr 0,2
2	RZ002	Rura zgrz. fi 12,6 gr 0,2
3	RZ003	Rura zgrz. fi 6,3 gr 0,3
4	RZ004	Rura zgrz. fi 12,6 gr 0,3
5	RZ011	Rura zgrz. kw 4 gr 0,2
6	RZ012	Rura zgrz. kw 5 gr 0,3

Rzutowanie w SQL

```
SELECT SymbolTowaru, NazwaTowaru  
FROM `towar`  
WHERE SymbolTowaru LIKE 'R%'
```

SymbolTowaru	NazwaTowaru
RZ001	Rura zgrz. fi 6,3 gr 0,2
RZ002	Rura zgrz. fi 12,6 gr 0,2
RZ003	Rura zgrz. fi 6,3 gr 0,3
RZ004	Rura zgrz. fi 12,6 gr 0,3
RZ011	Rura zgrz. kw 4 gr 0,2
RZ012	Rura zgrz. kw 5 gr 0,3

Warunki selekcji i porządkowanie

```
SELECT NrZamowienia, DataZamowienia
FROM Zamowienie
WHERE NrZamowienia LIKE '____2004' AND
DataZamowienia > '2004-04-04'
ORDER BY DataZamowienia DESC
```

NrZamowienia	DataZamowienia
005/2004	2004-04-07 00:00:00
003/2004	2004-04-06 00:00:00
004/2004	2004-04-06 00:00:00
002/2004	2004-04-05 00:00:00

Operatory logiczne i specjalne

AND - koniunkcja

OR - alternatywa

NOT - zaprzeczenie

BETWEEN *a* AND *b* – zawarty pomiędzy *a* i *b*

LIKE – podobny do

IN – zawarty w zbiorze

IS NULL – ma wartość NULL

Operatory logiczne i specjalne

```
SELECT NrZamowienia, DataZamowienia  
FROM Zamowienie  
WHERE DataZamowienia BETWEEN '2004-04-05' AND  
'2004-04-06'  
AND NrZamowienia IN ('003/2004', '004/2004')
```

NrZamowienia	DataZamowienia
003/2004	2004-04-06 00:00:00
004/2004	2004-04-06 00:00:00

Złączenia – algebra relacji

Iloczyn kartezjański

- (inaczej produkt) relacji R i S to relacja wszystkich uporządkowanych par krotek, z których pierwszy element pary należy do relacji R a drugi do S
- Schemat relacji $R \times S$ jest sumą schematów relacji R i S , w której powtarzające się atrybuty (kolumny) traktowane są jako odrębne elementy schematu, np. $R.A$ i $S.A$

Iloczyn kartezjański

R

Student	Język
Adam Kot	angielski
Adam Kot	niemiecki

S

Student	Przedmiot
Adam Kot	matematyka
Adam Kot	fizyka

R × *S*

R.Student	Język	S.Student	Przedmiot
Adam Kot	angielski	Adam Kot	matematyka
Adam Kot	angielski	Adam Kot	fizyka
Adam Kot	niemiecki	Adam Kot	matematyka
Adam Kot	niemiecki	Adam Kot	fizyka

Złączenie naturalne

- polega na połączeniu w pary tych krotek z relacji R i S , które mają identyczne wartości dla wszystkich wspólnych atrybutów i jest oznaczane $R \bowtie S$
- w rezultacie powstaje relacja, której schemat zawiera atrybuty relacji R i relacji S , przy czym wspólna część uwzględniana jest tylko raz

Złączenie naturalne

Student	Przedmiot	Semestr	Ocena
Adam Kot	Matematyka	I	3,0
Adam Kot	Fizyka	II	4,0
Jan Pies	Matematyka	I	2,0

Przedmiot	Semestr	Prowadzący
Matematyka	I	Prof. Wilk
Fizyka	II	Prof. Zajac
Matematyka	II	Prof. Kos

Student	Przedmiot	Semestr	Ocena	Prowadzący
Adam Kot	Matematyka	I	3,0	Prof. Wilk
Adam Kot	Fizyka	II	4,0	Prof. Zajac
Jan Pies	Matematyka	I	2,0	Prof. Wilk

Złączenie teta

- polega na złączeniu dwóch relacji R i S w iloczyn kartezyjański i wyborze z niego tych krotek, które spełniają wyrażenie warunkowe na parze lub zbiorze par atrybutów z R i S i jest oznaczane symbolem $R \bowtie_{\theta} R$ lub $R \bowtie_C S$, gdzie θ lub C to wyrażenia logiczne

Złączenie teta

R

Towar	Data_Od	Data_do	Cena
mąka	1.01.2004	31.01.2004	2,00
mąka	1.02.2004	31.03.2004	2,10
mąka	1.04.2004		2,05

S

Towar	Data	Ilość
mąka	15.03.2004	10

$R \bowtie_C S$

R.Towar	Data_Od	Data_do	Cena	S.Towar	Data	Ilość
mąka	1.02.2004	31.03.2004	2,10	mąka	15.03.2004	10

$C = (R.Towar = S.Towar \text{ AND } Data \geq Data_Od$
 $\text{ AND } Data \leq Data_Do)$

Równozłączenie

- to szczególny przypadek złączenia teta, w którym warunek ma charakter równości wybranych atrybutów obu relacji
- powtarzające się kolumny opisujące atrybuty z warunku złączenia są pomijane

Równozłączenie

R

Towar	Klient
stal	Exbud
cegła	PBS
złom	

S

Kontrahent	Miasto
Exbud	Kielce
PBS	Kraków
PHS	Tarnów

$R \bowtie_{R.Klient=S.Kontrahent} S$

Towar	Klient	Miasto
stal	Exbud	Kielce
cegła	PBS	Kraków

Typy złączeń

- złączenie wewnętrzne (inner join) – w relacji wynikowej występują wyłącznie te krotki, które spełniają warunek złączenia
- złączenie lewostronne zewnętrzne (left outer join) – zawiera wszystkie krotki **R** uzupełnione krotkami **S** spełniającymi warunek
- złączenie prawostronne zewnętrzne (right outer join) - zawiera wszystkie krotki **S** uzupełnione krotkami **R** spełniającymi warunek

Typy złączeń - ciąg dalszy

- złączenie zewnętrzne pełne (full outer join) – zawiera wszystkie krotki ***R*** oraz ***S*** uzupełnione wartościami typu ***NULL*** gdy do danej krotki nie pasuje żadna krotka z drugiej relacji
- złączenie zewnętrzne typu UNION - zawiera wszystkie krotki ***R*** uzupełnione krotkami ***S***, przy czym krotki wspólne prezentowane są tylko raz

Złączenie lewostronne zewnętrzne

R

Towar	Klient
stal	Exbud
cegła	PBS
złom	

S

Kontrahent	Miasto
Exbud	Kielce
PBS	Kraków
PHS	Tarnów

$R \bowtie_{R.Klient=S.Kontrahent} S$

Towar	Klient	Miasto
stal	Exbud	Kielce
cegła	PBS	Kraków
złom		

Złączenie prawostronne zewnętrzne

R

Towar	Klient
stal	Exbud
cegła	PBS
złom	

S

Kontrahent	Miasto
Exbud	Kielce
PBS	Kraków
PHS	Tarnów

$R \bowtie_{R.Klient=S.Kontrahent} S$

Towar	Klient	Miasto
stal	Exbud	Kielce
cegła	PBS	Kraków
	PHS	Tarnów

Złączenie zewnętrzne pełne

R

Towar	Klient
stal	Exbud
cegła	PBS
złom	

S

Kontrahent	Miasto
Exbud	Kielce
PBS	Kraków
PHS	Tarnów

$R \bowtie_{R.Klient=S.Kontrahent} S$

Towar	Klient	Miasto
stal	Exbud	Kielce
cegła	PBS	Kraków
złom		
	PHS	Tarnów

Złączenie typu UNION

	StudentId	Nazwisko	Imię	PESEL
	2	Kowalski	... Jan	... 87010106618
	3	Abacka	... Anna	... 88101004517
	4	Osa	... Ewa	... 86020405345
	6	Macioł	... Andrzej	... 56031606618
▶*	<i>NULL</i>	<i>NULL</i>	<i>NULL</i>	<i>NULL</i>

	ProwadzącyId	Nazwisko	Imię	Tytuł
	1	Macioł	... Andrzej	... dr inż.
	2	Kot	... Jan	... prof.
	3	Słoń	... Antoni	... mgr
▶*	<i>NULL</i>	<i>NULL</i>	<i>NULL</i>	<i>NULL</i>

	Nazwisko	Imię
▶	Abacka	... Anna
	Kot	... Jan
	Kowalski	... Jan
	Macioł	... Andrzej
	Osa	... Ewa
	Słoń	... Antoni

Złączenia EXCEPT i INTERSECT

- Złączenie EXCEPT prezentuje te krotki z relacji R (z lewej strony złączenia), które nie mają odpowiedników w relacji S
- Złączenie INTERSECT prezentuje te krotki z relacji R (z lewej strony złączenia), które mają odpowiedniki w relacji S

Złączenie typu EXCEPT

R

	IdKlienta	Nazwa	Adres
	1	Firma AS	Kraków
	2	Spółdzielnia SMOK	Zabierzów
	3	Spółka Kruk	Katowice
▶*	<i>NULL</i>	<i>NULL</i>	<i>NULL</i>

S

	IdZamowienia	Data	IdKlienta
	1	2007-10-10 00:...	1
	2	2007-10-12 00:...	2
▶*	<i>NULL</i>	<i>NULL</i>	<i>NULL</i>

R EXCEPT S

	IdKlienta
▶	3

Złączenie typu INTERSECT

R

	IdKlienta	Nazwa	Adres
	1	Firma AS	Kraków
	2	Spółdzielnia SMOK	Zabierzów
	3	Spółka Kruk	Katowice
▶*	NULL	NULL	NULL

S

	IdZamowienia	Data	IdKlienta
	1	2007-10-10 00:...	1
	2	2007-10-12 00:...	2
▶*	NULL	NULL	NULL

R INTERSECT S

	IdKlienta
▶	1
	2

Złączenia - SQL

Przykładowe dane (fragment)

IdKlienta	NazwaKlienta	Telefon	IdPoczto
1	FH Klin SA	48 12 1273210	30-121
2	Firma Krok Sp zoo	48 12 6374532	30-321
3	STALHANDEL	48 32 7865748	34-876
4	Rower Polska SA	48 12 2853364	32-082

IdBanku	IdKlienta	NrKonta
1	1	12345678901234567892022222
2	1	43527897963543645632726336
3	2	46748329374637843254632546
1	2	78789798979879879877878978
1	3	98087079643906432786443324
2	3	67876864376438209876473674
3	4	67686868768348364836483764

Iloczyn kartezjański

```
SELECT NazwaKlienta, NrKonta  
FROM Klient, Konto  
ORDER BY NazwaKlienta
```

NazwaKlienta	NrKonta
FH Klin SA	78789798979879879877878978
FH Klin SA	43527897963543645632726336
FH Klin SA	67686868768348364836483764
FH Klin SA	46748329374637843254632546
FH Klin SA	67876864376438209876473674
FH Klin SA	12345678901234567892022222
FH Klin SA	98087079643906432786443324
Firma Krok Sp zoo	12345678901234567892022222
Firma Krok Sp zoo	43527897963543645632726336
Firma Krok Sp zoo	46748329374637843254632546
Firma Krok Sp zoo	78789798979879879877878978
Firma Krok Sp zoo	67876864376438209876473674
Firma Krok Sp zoo	67686868768348364836483764
Firma Krok Sp zoo	98087079643906432786443324

	PrzedmiotId	Przedmiot
▶	1	matematyka
	2	fizyka
	3	programowanie komputerów
	4	bazy danych
*	<i>NULL</i>	<i>NULL</i>

	StudentId	Nazwisko	Imie	PESEL	Kod
	2	Kowalski ...	Jan ...	87010106618	32-098
	3	Abacka ...	Anna ...	88101004517	32-088
	4	Osa ...	Ewa ...	86020405345	<i>NULL</i>
	6	Macioł ...	Andrzej ...	56031606618	<i>NULL</i>
	7	a ...	b ...	12345678901	<i>NULL</i>
▶*	<i>NULL</i>	<i>NULL</i>	<i>NULL</i>	<i>NULL</i>	<i>NULL</i>

SELECT *
FROM Przedmioty CROSS JOIN
Studenci

	PrzedmiotId	Przedmiot	StudentId	Nazwisko	Imie
▶	1	matematyka	2	Kowalski	Jan
	1	matematyka	3	Abacka	Anna
	1	matematyka	4	Osa	Ewa
	1	matematyka	6	Macioł	Andrzej
	1	matematyka	7	a	b
	2	fizyka	2	Kowalski	Jan
	2	fizyka	3	Abacka	Anna
	2	fizyka	4	Osa	Ewa
	2	fizyka	6	Macioł	Andrzej
	2	fizyka	7	a	h

Złączenie naturalne

```
SELECT NazwaKlienta, NrKonta  
FROM Klient ,Konto  
WHERE Klient.IdKlienta = Konto.IdKlienta  
ORDER BY NazwaKlienta
```

```
SELECT NazwaKlienta, NrKonta  
FROM Klient JOIN Konto USING (IdKlienta)  
ORDER BY NazwaKlienta
```

NazwaKlienta	NrKonta
FH Klin SA	12345678901234567892022222
FH Klin SA	43527897963543645632726336
Firma Krok Sp zoo	46748329374637843254632546
Firma Krok Sp zoo	78789798979879879877878978
STALHANDEL	98087079643906432786443324
STALHANDEL	67876864376438209876473674
Rower Polska SA	67686868768348364836483764

Złączenie naturalne – trzy tabele

```
SELECT NazwaKlienta, NrKonta, NazwaBanku  
FROM Klient JOIN Konto USING (IdKlienta)  
JOIN Bank USING (IdBanku)  
ORDER BY NazwaKlienta, NrKonta
```

```
SELECT NazwaKlienta, NrKonta, NazwaBanku  
FROM Klient, Konto, Bank  
WHERE Klient.IdKlienta = Konto.IdKlienta AND  
Konto.IdBanku = Bank.IdBanku  
ORDER BY NazwaKlienta, NrKonta
```

Złączenie naturalne – trzy tabele

NazwaKlienta	NrKonta	NazwaBanku
FH Klin SA	12345678901234567892022222	Bank BPH
FH Klin SA	43527897963543645632726336	Bank Polski
Firma Krok Sp zoo	46748329374637843254632546	Bank Niemiecki
Firma Krok Sp zoo	78789798979879879877878978	Bank BPH
Rower Polska SA	67686868768348364836483764	Bank Niemiecki
STALHANDEL	67876864376438209876473674	Bank Polski
STALHANDEL	98087079643906432786443324	Bank BPH

Złączenia zewnętrzne – złączenie lewostronne

```
SELECT DISTINCT NazwaKlienta, DataZamowienia  
FROM Klient LEFT JOIN Zamowienie USING (IdKlienta)
```

NazwaKlienta	DataZamowienia
FH Klin SA	2004-04-04 00:00:00
FH Klin SA	2004-04-06 00:00:00
Firma Krok Sp zoo	2004-04-05 00:00:00
STALHANDEL	2004-04-06 00:00:00
Rower Polska SA	2004-04-07 00:00:00
PHPU OSA	[NULL]

Złączenia zewnętrzne – złączenie lewostronne

```
SELECT DISTINCT NazwaTowaru, DataZamowienia, Ilosc  
FROM Towar LEFT JOIN LiniaZamowienia USING  
(IdTowaru) LEFT JOIN Zamowienie USING (IdZamowienia)  
ORDER BY NazwaTowaru
```

NazwaTowaru	DataZamowienia	Ilosc
Rura zgrz. fi 12,6 gr 0,2	2004-04-04 00:00:00	12
Rura zgrz. fi 12,6 gr 0,2	2004-04-07 00:00:00	50
Rura zgrz. fi 12,6 gr 0,3	2004-04-06 00:00:00	12
Rura zgrz. fi 6,3 gr 0,2	2004-04-04 00:00:00	20
Rura zgrz. fi 6,3 gr 0,2	2004-04-06 00:00:00	50
Rura zgrz. fi 6,3 gr 0,2	2004-04-05 00:00:00	100
Rura zgrz. fi 6,3 gr 0,3	2004-04-04 00:00:00	25
Rura zgrz. fi 6,3 gr 0,3	2004-04-06 00:00:00	50
Rura zgrz. kw 4 gr 0,2	2004-04-07 00:00:00	30
Rura zgrz. kw 4 gr 0,2	2004-04-05 00:00:00	6
Rura zgrz. kw 5 gr 0,3	[NULL]	[NULL]
Zlaczka 1'	2004-04-06 00:00:00	100
Zlaczka 2'	2004-04-06 00:00:00	50
Zlaczka 3/4'	[NULL]	[NULL]

Zapytania zawierające unię

```
SELECT NazwaKlienta, NazwaTowaru, Ilosc
FROM Towar JOIN LiniaZamowienia USING (IdTowaru) JOIN
Zamowienie USING (IdZamowienia) JOIN Klient Using
(IdKlienta)
WHERE NazwaTowaru = 'Rura zgrz. fi 6,3 gr 0,3'
UNION
SELECT NazwaKlienta, NazwaTowaru, Ilosc
FROM Towar JOIN LiniaZamowienia USING (IdTowaru) JOIN
Zamowienie USING (IdZamowienia) JOIN Klient Using
(IdKlienta)
WHERE NazwaTowaru = 'Rura zgrz. fi 12,6 gr 0,2'
```

Zapytania zawierające unię

```
SELECT NazwaKlienta, NazwaTowaru, Ilosc
FROM Towar JOIN LiniaZamowienia USING (IdTowaru) JOIN
Zamowienie USING (IdZamowienia) JOIN Klient Using
(IdKlienta)
WHERE NazwaTowaru = 'Rura zgrz. fi 6,3 gr 0,3' OR
NazwaTowaru = 'Rura zgrz. fi 12,6 gr 0,2'
```

NazwaKlienta	NazwaTowaru	Ilosc
FH Klin SA	Rura zgrz. fi 6,3 gr 0,3	25
STALHANDEL	Rura zgrz. fi 6,3 gr 0,3	50
FH Klin SA	Rura zgrz. fi 12,6 gr 0,2	12
Rower Polska SA	Rura zgrz. fi 12,6 gr 0,2	50

Zapytania zawierające unię

```
SELECT NazwaKlienta
FROM Towar JOIN LiniaZamowienia USING (IdTowaru) JOIN
Zamowienie USING (IdZamowienia) JOIN Klient Using
(IdKlienta)
WHERE NazwaTowaru = 'Rura zgrz. fi 6,3 gr 0,3'
UNION
SELECT NazwaKlienta
FROM Towar JOIN LiniaZamowienia USING (IdTowaru) JOIN
Zamowienie USING (IdZamowienia) JOIN Klient Using
(IdKlienta)
WHERE NazwaTowaru = 'Rura zgrz. fi 12,6 gr 0,2'
```

Zapytania zawierające unię

```
+-----+
| NazwaKlienta |
+-----+
| FH Klin SA   |
| STALHANDEL  |
| Rower Polska SA |
+-----+
```

Zapytania zawierające unię

```
SELECT NazwaKlienta
FROM Towar JOIN LiniaZamowienia USING (IdTowaru) JOIN
Zamowienie USING (IdZamowienia) JOIN Klient Using
(IdKlienta)
WHERE NazwaTowaru = 'Rura zgrz. fi 6,3 gr 0,3' OR
NazwaTowaru = 'Rura zgrz. fi 12,6 gr 0,2'
```

```
+-----+
| NazwaKlienta |
+-----+
| FH Klin SA   |
| FH Klin SA   |
| STALHANDEL   |
| Rower Polska SA |
+-----+
```

Zapytania zawierające unię

```
SELECT DISTINCT NazwaKlienta
FROM Towar JOIN LiniaZamowienia USING (IdTowaru) JOIN
Zamowienie USING (IdZamowienia) JOIN Klient Using
(IdKlienta)
WHERE NazwaTowaru = 'Rura zgrz. fi 6,3 gr 0,3' OR
NazwaTowaru = 'Rura zgrz. fi 12,6 gr 0,2'
```

```
+-----+
| NazwaKlienta |
+-----+
| FH Klin SA   |
| STALHANDEL   |
| Rower Polska SA |
+-----+
```

Operator EXIST i zapytania skorelowane

```
SELECT SymbolTowaru, NazwaTowaru
FROM Towar Tablica1
WHERE EXISTS
      (SELECT *
        FROM LiniaZamowienia
        WHERE Tablica1.IdTowaru =
              LiniaZamowienia.IdTowaru)
```

```
SELECT DISTINCT SymbolTowaru, NazwaTowaru
FROM Towar JOIN LiniaZamowienia USING (IdTowaru)
```

Operator EXIST i zapytania skorelowane

SymbolTowaru	NazwaTowaru
RZ001	Rura zgrz. fi 6,3 gr 0,2
RZ002	Rura zgrz. fi 12,6 gr 0,2
RZ003	Rura zgrz. fi 6,3 gr 0,3
RZ004	Rura zgrz. fi 12,6 gr 0,3
RZ011	Rura zgrz. kw 4 gr 0,2
ZL001	Zlaczka 1'
ZL002	Zlaczka 2'

Obliczenia i grupowanie

- Operatory arytmetyczne:
- + suma,
- - różnica
- * mnożenie
- / dzielenie
- +/- operatory unarne (zachowaj/ zmień znak)
- Inne operacje realizowane są przy użyciu funkcji charakterystycznych dla dialektu, np. `power(X,Y)`, `log(X)`, `log10(X)` w MySQL

Operacje na łańcuchach

- konkatencja łańcuchów

SQL-92

```
SELECT KodPocztowy || ' ' || Miejscowosc)
FROM `klient`
```

Microsoft SQL Server

```
SELECT SymbolTowaru + ' ' + NazwaTowaru AS NowaNazwa
FROM Towar
```

MySQL

```
SELECT concat(KodPocztowy, ' ', Miejscowosc)
FROM `klient`
```


Wybrane funkcje tekstowe

- UPPER i LOWER konwertują łańcuchy tekstowe na duże lub małe litery
- TRIM (słowo)
- TRIM (BOTH znak FROM słowo)
- TRIM (LEADING znak FROM słowo)
- TRIM (TRAILING znak FROM słowo)
- SUBSTRING (słowo FROM poz_startowa FOR liczba_znaków)

KodPocztowy	Miejscowosc
30-121	Kraków
30-321	Kraków
34-876	Sosnowiec
32-082	Zabierzów
30-432	Kraków

```

SELECT SUBSTRING(TRIM(LEADING '3' FROM
CONCAT(KodPocztowy, ' ', Miejscowosc)) FROM 1 FOR 12) AS
Miasto
FROM klient
WHERE UPPER(Miejscowosc) = 'KRAKÓW'

```

Miasto
0-121 Kraków
0-321 Kraków
0-432 Kraków

Działania dotyczące czasu

```
SELECT NazwaKlienta, DATE_FORMAT(DataZamowienia, '%Y
%m %d') AS Data1, CURRENT_DATE AS Data2,
TO_DAYS(CURRENT_DATE) - TO_DAYS( DataZamowienia)
Dni
FROM Zamowienie JOIN Klient USING (IdKlienta)
```

NazwaKlienta	Data1	Data2	Dni
FH Klin SA	2004 04 04	2004-05-03	29
FH Klin SA	2004 04 06	2004-05-03	27
Firma Krok Sp zoo	2004 04 05	2004-05-03	28
STALHANDEL	2004 04 06	2004-05-03	27
Rower Polska SA	2004 04 07	2004-05-03	26

Działania dotyczące czasu

```
SELECT NazwaKlienta, DATE_FORMAT(DataZamowienia, '%Y
%m %d') AS DataFaktury,
DATE_FORMAT(DATE_ADD(DataZamowienia, INTERVAL 14
DAY), '%Y %m %d') AS TerminPlatnosci
FROM Zamowienie JOIN Klient USING (IdKlienta)
```

NazwaKlienta	DataFaktury	TerminPlatnosci
FH Klin SA	2004 04 04	2004 04 18
FH Klin SA	2004 04 06	2004 04 20
Firma Krok Sp zoo	2004 04 05	2004 04 19
STALHANDEL	2004 04 06	2004 04 20
Rower Polska SA	2004 04 07	2004 04 21

Operacje na zbiorach

- COUNT zwraca liczbę wierszy
- SUM zwraca sumę wartości danej kolumny we wszystkich wierszach ze zbioru
- AVG zwraca średnią wartość danej kolumny we wszystkich wierszach ze zbioru
- MIN zwraca najmniejszą wartość danej kolumny we wszystkich wierszach ze zbioru
- MAX zwraca największą wartość danej kolumny we wszystkich wierszach ze zbioru

Zbiór wejściowy

NazwaTowaru	Data	Ilosc	Cena
Rura zgrz. fi 12,6 gr 0,2	2004 04 04	12	1.75
Rura zgrz. fi 12,6 gr 0,2	2004 04 07	50	1.75
Rura zgrz. fi 12,6 gr 0,3	2004 04 06	12	2.05
Rura zgrz. fi 6,3 gr 0,2	2004 04 05	100	1.40
Rura zgrz. fi 6,3 gr 0,2	2004 04 04	20	1.50
Rura zgrz. fi 6,3 gr 0,2	2004 04 06	50	1.50
Rura zgrz. fi 6,3 gr 0,3	2004 04 04	25	2.10
Rura zgrz. fi 6,3 gr 0,3	2004 04 06	50	2.10
Rura zgrz. kw 4 gr 0,2	2004 04 05	6	2.20
Rura zgrz. kw 4 gr 0,2	2004 04 07	30	2.20
Rura zgrz. kw 5 gr 0,3	[NULL]	[NULL]	[NULL]
Złączka 1'	2004 04 06	100	0.90
Złączka 2'	2004 04 06	50	1.10
Złączka 3/4'	[NULL]	[NULL]	[NULL]

Funkcja COUNT

```
SELECT COUNT(*) AS Liczba  
FROM Towar LEFT JOIN LiniaZamowienia USING (IdTowaru  
LEFT JOIN Zamowienie USING (IdZamowienia)  
ORDER BY NazwaTowaru
```

14

```
SELECT COUNT(Ilosc) AS Liczba  
FROM Towar LEFT JOIN LiniaZamowienia USING (IdTowaru  
LEFT JOIN Zamowienie USING (IdZamowienia)
```

12

```
SELECT COUNT(DISTINCT NazwaTowaru) AS Liczba  
FROM Towar LEFT JOIN LiniaZamowienia USING (IdTowaru  
LEFT JOIN Zamowienie USING (IdZamowienia)  
ORDER BY NazwaTowaru
```

Pozostałe funkcje

```
SELECT SUM(Ilosc*LiniaZamowienia.Cena)
FROM Towar LEFT JOIN LiniaZamowienia USING (IdTowaru)
LEFT JOIN Zamowienie USING (IdZamowienia)
```

759,8

```
SELECT AVG(LiniaZamowienia.Cena)
FROM Towar LEFT JOIN LiniaZamowienia USING (IdTowaru)
WHERE Towar.IdTowaru=1
```

1,46667

```
SELECT MAX( DATE_FORMAT(DataZamowienia, '%Y %m
%d') ) AS Data
FROM Towar LEFT JOIN LiniaZamowienia USING (IdTowaru)
LEFT JOIN Zamowienie USING (IdZamowienia)
ORDER BY NazwaTowaru
```

2004 04 07

Funkcje w predykatkach – Microsoft SQL Server

```
SELECT  Towar.NazwaTowaru, LiniaZamowienia.Cena
FROM    LiniaZamowienia INNER JOIN
        Towar ON LiniaZamowienia.IdTowaru =
        Towar.IdTowaru
WHERE   (LiniaZamowienia.Cena >
        (SELECT  AVG(CAST(Cena AS decimal(10, 2)))
         FROM    LiniaZamowienia))
ORDER BY Towar.NazwaTowaru
```

Funkcja AVG w predykcje - dane

NazwaTowaru	Cena
Rura zgrz. fi 12,6 gr 0,2	1.75
Rura zgrz. fi 12,6 gr 0,2	1.75
Rura zgrz. fi 12,6 gr 0,3	2.05
Rura zgrz. fi 6,3 gr 0,2	1.50
Rura zgrz. fi 6,3 gr 0,2	1.40
Rura zgrz. fi 6,3 gr 0,2	1.50
Rura zgrz. fi 6,3 gr 0,3	2.10
Rura zgrz. fi 6,3 gr 0,3	2.10
Rura zgrz. kw 4 gr 0,2	2.20
Rura zgrz. kw 4 gr 0,2	2.20
Zlaczka 1'	0.90
Zlaczka 2'	1.10

Funkcja AVG w predykcje - wynik

NazwaTowaru	Cena
Rura zgrz. fi 12,6 gr 0,2	1.75
Rura zgrz. fi 12,6 gr 0,2	1.75
Rura zgrz. fi 12,6 gr 0,3	2.05
Rura zgrz. fi 6,3 gr 0,3	2.10
Rura zgrz. fi 6,3 gr 0,3	2.10
Rura zgrz. kw 4 gr 0,2	2.20
Rura zgrz. kw 4 gr 0,2	2.20

Zapytania grupujące - błędy

```
SELECT    DATE_FORMAT(DataZamowienia, '%Y %m %d')
AS Data,  Towar.NazwaTowaru, SUM(Ilosc) AS Ilosc ,
SUM(Ilosc*Cena) AS Wartosc
FROM Zamowienie JOIN LiniaZamowienia USING
(IdZamowienia) JOIN Towar USING ( IdTowaru)
GROUP BY DataZamowienia
ORDER BY DataZamowienia
```

Data	NazwaTowaru	Ilosc	Wartosc
2004 04 04	Rura zgrz. fi 6,3 gr 0,2	57	103.5
2004 04 05	Rura zgrz. fi 6,3 gr 0,2	106	153.2
2004 04 06	Rura zgrz. fi 6,3 gr 0,2	262	349.6
2004 04 07	Rura zgrz. fi 12,6 gr 0,2	80	153.5

Zapytania grupujące - błędy

```
SELECT  NazwaKLienta, DATE_FORMAT(DataZamowienia,  
'%Y %m %d') AS Data, SUM(Ilosc*Cena) AS Wartosc  
FROM    Klient JOIN Zamowienie USING (IdKlienta) JOIN  
LiniaZamowienia USING (IdZamowienia)  
GROUP BY NazwaKlienta  
ORDER BY NazwaKLienta, DataZamowienia
```

NazwaKLienta	Data	Wartosc
FH Klin SA	2004 04 04	203.1
Firma Krok Sp zoo	2004 04 05	153.2
Rower Polska SA	2004 04 07	153.5
STALHANDEL	2004 04 06	250

Zapytania grupujące

```
SELECT  NazwaKLienta, DATE_FORMAT(DataZamowienia,  
'%Y %m %d') AS Data, SUM(Ilosc*Cena) AS Wartosc  
FROM    Klient JOIN Zamowienie USING (IdKlienta) JOIN  
LiniaZamowienia USING (IdZamowienia)  
GROUP BY NazwaKlienta, DataZamowienia  
ORDER BY NazwaKLienta, DataZamowienia
```

NazwaKLienta	Data	Wartosc
FH Klin SA	2004 04 04	103.5
FH Klin SA	2004 04 06	99.6
Firma Krok Sp zoo	2004 04 05	153.2
Rower Polska SA	2004 04 07	153.5
STALHANDEL	2004 04 06	250

Zapytania grupujące - ograniczenia

```
SELECT  NazwaKLienta, DATE_FORMAT(DataZamowienia,
'%Y %m %d') AS Data, SUM(Ilosc*Cena) AS Wartosc
FROM    Klient JOIN Zamowienie USING (IdKlienta) JOIN
LiniaZamowienia USING (IdZamowienia)
WHERE   DataZamowienia > '2004-04-04'
GROUP BY NazwaKlienta, DataZamowienia
ORDER BY NazwaKLienta, DataZamowienia
```

NazwaKLienta	Data	Wartosc
FH Klin SA	2004 04 06	99.6
Firma Krok Sp zoo	2004 04 05	153.2
Rower Polska SA	2004 04 07	153.5
STALHANDEL	2004 04 06	250

Zapytania grupujące - ograniczenia

```
SELECT  NazwaKlienta, DATE_FORMAT(DataZamowienia,  
'%Y %m %d') AS Data, SUM(Ilosc*Cena) AS Wartosc  
FROM    Klient JOIN Zamowienie USING (IdKlienta) JOIN  
LiniaZamowienia USING (IdZamowienia)  
GROUP BY NazwaKlienta, DataZamowienia  
HAVING Data > '2004 04 04'  
ORDER BY NazwaKlienta, DataZamowienia
```

NazwaKlienta	Data	Wartosc
FH Klin SA	2004 04 06	99.6
Firma Krok Sp zoo	2004 04 05	153.2
Rower Polska SA	2004 04 07	153.5
STALHANDEL	2004 04 06	250

Zapytania grupujące - ograniczenia

- WHERE działa przed sformowaniem grup a HAVING po – predykat tej opcji musi więc odnosić się do kryteriów wykorzystanych przy tworzeniu grup
- Możliwe jest jednoczesne wykorzystania WHERE i HAVING

```
SELECT    COUNT(Srednica) AS Liczba, Norma
FROM      kalibrowane$
WHERE     (Srednica > 40)
GROUP BY  Norma, GruboscScianki
HAVING    (GruboscScianki = 2)
```

Modyfikowanie danych

Wstawianie wierszy

INSERT INTO Klient

(NazwaKlienta, Telefon, KodPocztowy, Miejscowosc, Ulica,
NrDomuMieszkania, Email)

VALUES('Nowy klient', '48 12 1234567', '30-333', 'Bolechowice',
'Jurajska', '20','ala@tlen.pl')

IdKlienta	NazwaKlienta	Telefon
3	STALHANDEL	48 32 7865748
2	Firma Krok Sp zoo	48 12 6374532
5	PHPU OSA	48 12 6372312
4	Rower Polska SA	48 12 2853364
1	FH Klin SA	48 12 1273210
6	Nowy klient	48 12 1234567

Kopiowanie wierszy

```
INSERT INTO zamowieniedoedycji
SELECT NazwaKlienta, NrZamowienia, NazwaTowaru, Ilosc,
Cena
FROM Klient JOIN Zamowienie USING (IdKlienta) JOIN
LiniaZamowienia USING (IdZamowienia) JOIN Towar USING
(IdTowaru)
WHERE Zamowienie.DataZamowienia = '2004-04-05'
```

NazwaKlienta	NrZamowienia	NazwaTowaru
Firma Krok Sp zoo	002/2004	Rura zgrz. fi 6,3 gr 0
Firma Krok Sp zoo	002/2004	Rura zgrz. kw 4 gr 0,2

Kopiowanie wierszy

- Uwaga! Kolumny zwracane przez SELECT muszą odpowiadać tym, z których składa się tabela docelowa, co do ilości i typu danych
- Tworzenie trwałych kopii tabel czy zapytań ma sens tylko wtedy gdy skraca przetwarzanie danych lub ułatwia edycję w osadzonym SQL

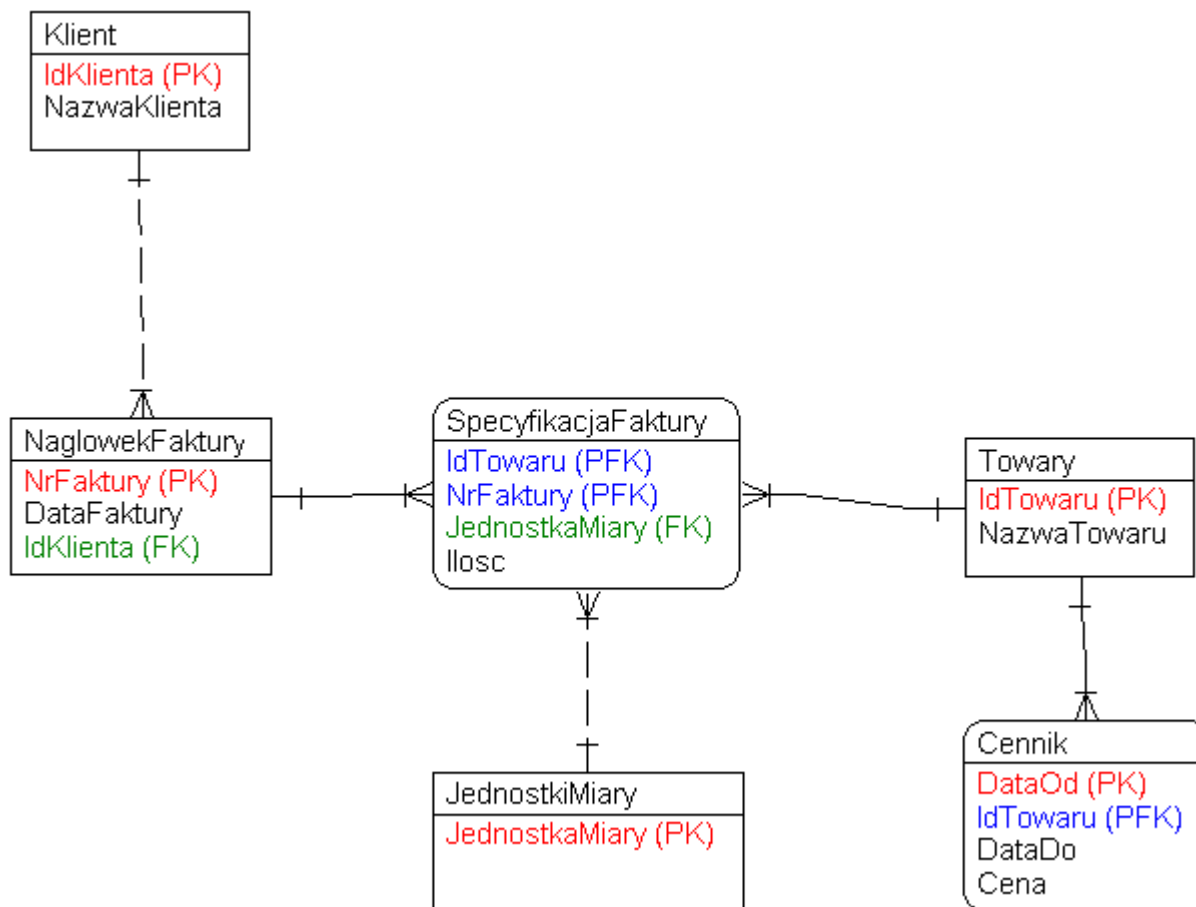
Modyfikowanie danych - Microsoft SQL Server

```
UPDATE  LiniaZamowienia
SET      Cena = Cena * 1.1
WHERE    (IdZamowienia IN
          (SELECT  IdZamowienia
           FROM    Zamowienie
           WHERE   DataZamowienia = '2004-04-04' AND
                  IdKlienta =
                  (SELECT  IdKlienta
                   FROM    Klient
                   WHERE   NazwaKlienta = 'FH Klin SA'))))
```

Usuwanie wierszy - Microsoft SQL Server

```
DELETE FROM LiniaZamowienia
WHERE (IdZamowienia IN
      (SELECT IdZamowienia
      FROM Zamowienie
      WHERE DataZamowienia = '2004-04-04' AND
            IdKlienta =
            (SELECT IdKlienta
            FROM Klient
            WHERE NazwaKlienta = 'FH Klin SA'))))
```

Przykładowa baza danych



Tworzenie i użycie tabeli

```
create database Dydaktyka
```

```
Use Dydaktyka
```

Tworzenie tabel i związków

Create table Klient

```
(  
    IdKlienta Bigint Identity NOT NULL,  
    NazwaKlienta Char(100) NULL,  
Primary Key (IdKlienta)
```

```
)  
Create table NaglowekFaktury
```

```
(  
    NrFaktury Char(20) NOT NULL,  
    DataFaktury Datetime NULL,  
    IdKlienta Bigint NOT NULL,  
Primary Key (NrFaktury),  
foreign key(IdKlienta) references Klient (IdKlienta)  
on update no action on delete no action
```

Tworzenie tabel i związków

Create table Cennik

(

DataOd Datetime NOT NULL,

IdTowaru Bigint NOT NULL,

DataDo Datetime NOT NULL,

Cena Money NULL,

Primary Key (DataOd,IdTowaru),

foreign key(IdTowaru) references Towary (IdTowaru)

on update no action on delete no action

)

Wprowadzanie danych

```
insert into Cennik (DataOd, DataDo, IdTowaru, Cena)  
values ('2009-01-01', '2009-03-01',1,2)
```

```
insert into Cennik (DataOd, DataDo, IdTowaru, Cena)  
values ('2009-01-01', "",2,3)
```

```
insert into Cennik (DataOd, DataDo, IdTowaru, Cena)  
values ('2009-03-02', '2009-01-01',1,2.2)
```

Prezentacja wszystkich krotek

```
select * from cennik
```

	DataOd	IdTowaru	DataDo	Cena
1	2009-01-01 00:00:00.000	1	2009-03-01 00:00:00.000	2,00
2	2009-01-01 00:00:00.000	2	NULL	3,00
3	2009-03-02 00:00:00.000	1	NULL	2,20

Wszystkie dane

	IdKlienta	NazwaKlienta
1	1	Firma As
2	2	Firma Kot

	NrFaktury	DataFaktury	IdKlienta
1	123	2009-01-15 00:00:00.00000000	1
2	128	2009-01-16 00:00:00.00000000	2
3	325	2009-03-15 00:00:00.00000000	1

	IdTowaru	NazwaTowaru
1	1	Cukier
2	2	Mąka
3	3	Sól

	JednostkaMiary
1	kg
2	szt

	IdTowaru	NrFaktury	JednostkaMiary	Ilosc
1	1	123	kg	100
2	1	128	kg	120
3	1	325	kg	110
4	2	123	kg	50
5	3	325	kg	200

	DataOd	IdTowaru	DataDo	Cena
1	2009-01-01 00:00:00.000	1	2009-03-01 00:00:00.000	2,00
2	2009-01-01 00:00:00.000	2	NULL	3,00
3	2009-03-02 00:00:00.000	1	NULL	2,20

Złączenie nieprawidłowe

```
SELECT Klient.NazwaKlienta, NaglowekFaktury.NrFaktury, NaglowekFaktury.DataFaktury,  
Towary.NazwaTowaru, SpecyfikacjaFaktury.Ilosc,  
        JednostkiMiary.JednostkaMiary, Cennik.Cena  
FROM SpecyfikacjaFaktury INNER JOIN  
        JednostkiMiary ON SpecyfikacjaFaktury.JednostkaMiary =  
JednostkiMiary.JednostkaMiary INNER JOIN  
        NaglowekFaktury ON SpecyfikacjaFaktury.NrFaktury =  
NaglowekFaktury.NrFaktury INNER JOIN  
        Klient ON NaglowekFaktury.IdKlienta = Klient.IdKlienta INNER JOIN  
        Towary ON SpecyfikacjaFaktury.IdTowaru = Towary.IdTowaru INNER JOIN  
        Cennik ON Towary.IdTowaru = Cennik.IdTowaru  
ORDER BY NaglowekFaktury.NrFaktury
```

Efekt

	NazwaKlienta	NrFaktury	DataFaktury	NazwaTowaru	Ilosc	JednostkaMiary	Cena
1	Firma As	123	2009-01-15 00:00:00.00000000	Cukier	100	kg	2,00
2	Firma Kot	128	2009-01-16 00:00:00.00000000	Cukier	120	kg	2,00
3	Firma As	325	2009-03-15 00:00:00.00000000	Cukier	110	kg	2,00
4	Firma As	123	2009-01-15 00:00:00.00000000	Mąka	50	kg	3,00
5	Firma As	123	2009-01-15 00:00:00.00000000	Cukier	100	kg	2,20
6	Firma Kot	128	2009-01-16 00:00:00.00000000	Cukier	120	kg	2,20
7	Firma As	325	2009-03-15 00:00:00.00000000	Cukier	110	kg	2,20

Poprawne złączenie

```
SELECT    NazwaKlienta, NaglowekFaktury.NrFaktury, DataFaktury, NazwaTowaru, Ilosc,  
JednostkiMiary.JednostkaMiary, Cena  
FROM      SpecyfikacjaFaktury JOIN  
JednostkiMiary ON SpecyfikacjaFaktury.JednostkaMiary = JednostkiMiary.JednostkaMiary  
JOIN NaglowekFaktury ON SpecyfikacjaFaktury.NrFaktury = NaglowekFaktury.NrFaktury  
JOIN Klient ON NaglowekFaktury.IdKlienta = Klient.IdKlienta  
JOIN Towary ON SpecyfikacjaFaktury.IdTowaru = Towary.IdTowaru  
JOIN Cennik ON Towary.IdTowaru = Cennik.IdTowaru  
AND NaglowekFaktury.DataFaktury > Cennik.DataOd  
AND (NaglowekFaktury.DataFaktury < Cennik.DataDo OR Cennik.DataDo IS NULL)  
ORDER BY NaglowekFaktury.NrFaktury
```

Efekt

	NazwaKlienta	NrFaktury	DataFaktury	NazwaTowaru	Ilosc	JednostkaMiary	Cena
1	Firma As	123	2009-01-15 00:00:00.00000000	Cukier	100	kg	2,00
2	Firma Kot	128	2009-01-16 00:00:00.00000000	Cukier	120	kg	2,00
3	Firma As	123	2009-01-15 00:00:00.00000000	Mąka	50	kg	3,00
4	Firma As	325	2009-03-15 00:00:00.00000000	Cukier	110	kg	2,20

Obliczamy wartość

```
SELECT NazwaKlienta, NaglowekFaktury.NrFaktury, DataFaktury,  
NazwaTowaru, Ilosc, JednostkiMiary.JednostkaMiary, Cena,  
Ilosc * Cena AS Wartosc
```

Efekt

	NazwaKlienta	NrFaktury	DataFaktury	NazwaTowaru	Ilosc	JednostkaMiary	Cena	Wartosc
1	Firma As	123	2009-01-15 00:00:00.00000000	Cukier	100	kg	2,00	200
2	Firma As	123	2009-01-15 00:00:00.00000000	Mąka	50	kg	3,00	150
3	Firma Kot	128	2009-01-16 00:00:00.00000000	Cukier	120	kg	2,00	240
4	Firma As	325	2009-03-15 00:00:00.00000000	Cukier	110	kg	2,20	242