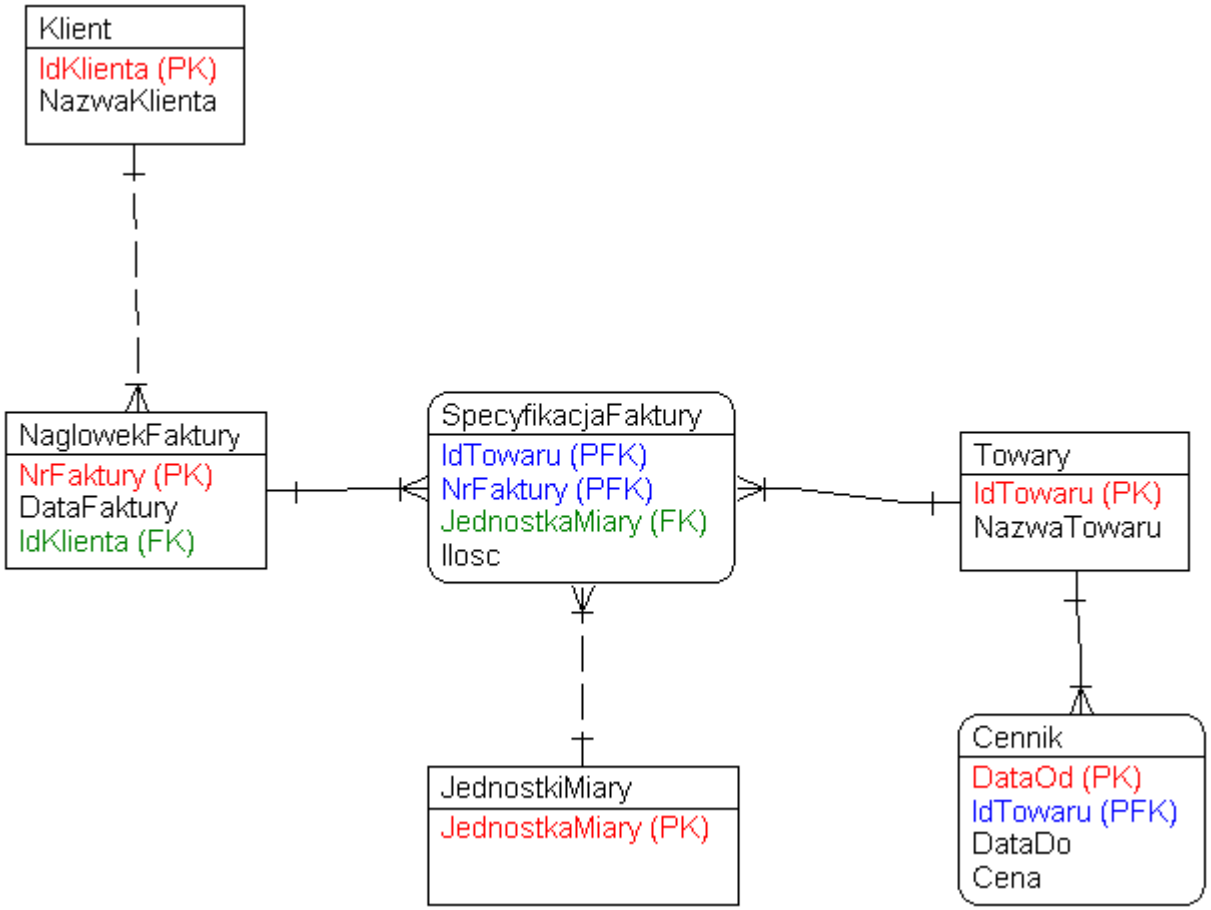
Język SQL

Część II

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 00:00:00.0000000	1
2	128	2009-01-16 00:00:00.0000000	2
3	325	2009-03-15 00:00:00.0000000	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 00:00:00.0000000	Cukier	100	kg	2,00
2	Firma Kot	128	2009-01-16 00:00:00.0000000	Cukier	120	kg	2,00
3	Firma As	325	2009-03-15 00:00:00.0000000	Cukier	110	kg	2,00
4	Firma As	123	2009-01-15 00:00:00.0000000	Mąka	50	kg	3,00
5	Firma As	123	2009-01-15 00:00:00.0000000	Cukier	100	kg	2,20
6	Firma Kot	128	2009-01-16 00:00:00.0000000	Cukier	120	kg	2,20
7	Firma As	325	2009-03-15 00:00:00.0000000	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 00:00:00.0000000	Cukier	100	kg	2,00
2	Firma Kot	128	2009-01-16 00:00:00.0000000	Cukier	120	kg	2,00
3	Firma As	123	2009-01-15 00:00:00.0000000	Mąka	50	kg	3,00
4	Firma As	325	2009-03-15 00:00:00.0000000	Cukier	110	kg	2,20

Obliczamy wartość

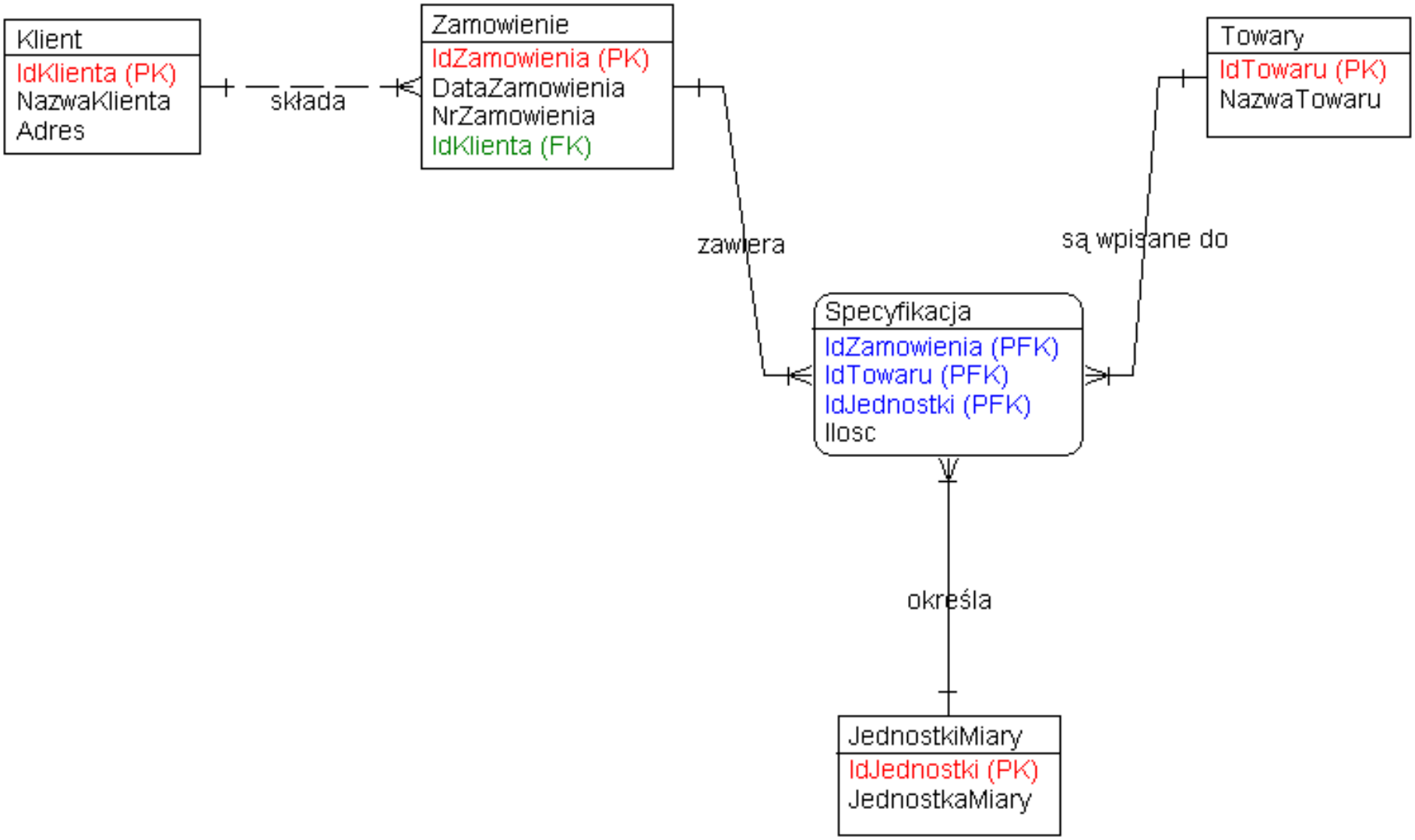
```
SELECT    NazwaKlienta, NaglowekFaktury.NrFaktury, DataFaktury,
NazwaTowaru,  Ilosc, JednostkiMiary.JednostkaMiary, Cena,
Ilosc * Cena AS Wartosc
```

Efekt

	NazwaKlienta	NrFaktury	DataFaktury	NazwaTowaru	Ilosc	JednostkaMiary	Cena	Wartosc
1	Firma As	123	2009-01-15 00:00:00.0000000	Cukier	100	kg	2,00	200
2	Firma As	123	2009-01-15 00:00:00.0000000	Mąka	50	kg	3,00	150
3	Firma Kot	128	2009-01-16 00:00:00.0000000	Cukier	120	kg	2,00	240
4	Firma As	325	2009-03-15 00:00:00.0000000	Cukier	110	kg	2,20	242

Integralność danych w SQL

- Wprowadzenie kluczy podstawowych i obcych zapewnia automatyczną kontrolę poprawności struktury danych i operacji przetwarzania danych
- Klucz podstawowy zapewnia unikalność i możliwość identyfikacji każdego zapisu
- Klucze obce zapewniają integrację referencyjną głoszącą, że każda niepusta wartość klucza obcego musi odpowiadać jednej z istniejących wartości klucza podstawowego


Klient
IdKlienta (PK)
NazwaKlienta
Adres
składa
Zamowienie
IdZamowienia (PK)
DataZamowienia
NrZamowienia
IdKlienta (FK)
zawiera
Towary
IdTowaru (PK)
NazwaTowaru
są wpisane do
Specyfikacja
IdZamowienia (PFK)
IdTowaru (PFK)
IdJednostki (PFK)
Ilosc
określa
JednostkiMiary
IdJednostki (PK)
JednostkaMiary

Klucz podstawowy

- W tablicach rodzicach (parent) jest niezbędny jako łącznik z tablicami dziećmi (child)
- Najlepiej rolę tą spełnia klucz, który jest kolumną tożsamości (identity)

Klucz podstawowy w definicji i zmianie tabeli

```
CREATE TABLE Towary (
        IdTowaru Int NOT NULL AUTO_INCREMENT,
        NazwaTowaru Char(50),
 PRIMARY KEY (IdTowaru)) ENGINE = MyISAM
ROW_FORMAT = Default;

ALTER TABLE Towary DROP PRIMARY KEY;

ALTER TABLE Towary ADD PRIMARY KEY (IdTowaru);
```

Klucze obce

- W związkach nieidentyfikujących pozwalają na tworzenie złączeń (nie jest do tego konieczne wymuszanie integralności)
- W związkach identyfikujących pozwalają na kontrolę unikalności krotek w relacjach dzieciach

```
CREATE TABLE Klient (
        IdKlienta Int NOT NULL AUTO_INCREMENT,
        NazwaKlienta Char(50),
        Adres Char(50),
 PRIMARY KEY (IdKlienta)) ENGINE = MyISAM
ROW_FORMAT = Default;

CREATE TABLE Zamowienie (
        IdZamowienia Int NOT NULL AUTO_INCREMENT,
        DataZamowienia Date,
        NrZamowienia Char(20),
        IdKlienta Int,
 PRIMARY KEY (IdZamowienia)) ENGINE = MyISAM
ROW_FORMAT = Default;
```

Złączenie naturalne

```
mysql> select nazwaklienta, datazamowienia from klient join zamowienie using
    -> (idklienta) order by nazwaklienta;
+----------------------------------+-----------------+
| nazwaklienta                     | datazamowienia  |
+----------------------------------+-----------------+
| Fabryka Rower~w                  | 2005-03-30      |
| Fabryka Rower~w                  | 2005-03-30      |
| Fabryka Rower~w                  | 2005-03-30      |
| Fabryka Rower~w                  | 2005-03-30      |
| Fabryka Rower~w                  | 2005-03-30      |
| Fabryka w~zk~w dzieciźcych       | 2005-04-02      |
| Hurtownia Materia|~w Budowlanych | 2005-03-30      |
| Klient 2                         | 2005-04-06      |
| Klient 3                         | 2005-04-07      |
| Klient 5                         | 2005-04-02      |
| Klient 5                         | 2005-04-02      |
| Klient 5                         | 2005-04-02      |
| Sklep Og~lny                     | 2005-03-27      |
+----------------------------------+-----------------+
13 rows in set (0.39 sec)
```

Złożony klucz obcy

```
CREATE TABLE JednostkiMiary (
        IdJednostki Int NOT NULL AUTO_INCREMENT,
        JednostkaMiary Char(20),
 PRIMARY KEY (IdJednostki)) ENGINE = MyISAM
ROW_FORMAT = Default;

CREATE TABLE Specyfikacja (
        IdZamowienia Int NOT NULL,
        IdTowaru Int NOT NULL,
        IdJednostki Int NOT NULL,
        Ilosc Float,
 PRIMARY KEY (IdZamowienia,IdTowaru,IdJednostki)) ENGINE
= MyISAM
ROW_FORMAT = Default;
```

```
mysql> select * from specyfikacja;
+--------------+----------+--------------+-------+
| IdZamowienia | IdTowaru | IdJednostki | Ilosc |
+--------------+----------+--------------+-------+
|            1 |        1 |            1 |   100 |
|            1 |        2 |            1 |   150 |
|            2 |        1 |            2 |   200 |
|            2 |        2 |            1 |   120 |
|           59 |        1 |            2 |   200 |
|           59 |        2 |            1 |   100 |
|           60 |        2 |            1 |    50 |
|           60 |        1 |            2 |   100 |
|           66 |        3 |            1 |   500 |
|           61 |        2 |            1 |   543 |
|           62 |        2 |            1 |   100 |
|           62 |        1 |            1 |   200 |
|           63 |        2 |            1 |   105 |
|           70 |        1 |            1 |   250 |
|           70 |        3 |            1 |   200 |
|           69 |        2 |            2 |   120 |
|           69 |        6 |            3 |   300 |
|           69 |        1 |            1 |   250 |
|           69 |        3 |            1 |   200 |
|           69 |        2 |            1 |   150 |
|           70 |        6 |            3 |   320 |
|           71 |        3 |            1 |   150 |
|           70 |        2 |            1 |   150 |
|           71 |        2 |            1 |   300 |
|           72 |        2 |            1 |   100 |
|           72 |        3 |            1 |   300 |
+--------------+----------+--------------+-------+
26 rows in set (0.41 sec)

mysql> insert into specyfikacja (idzamowienia, idtowaru, idjednostki, ilosc)
    -> values('1', '1', '1', '200');
ERROR 1062 (23000): Powt~rzone wyst?pienie '1-1-1' dla klucza 1
mysql>
```

próba wprowadzenia identycznego klucza

Wymuszanie integralności

- REFERENCES – podaje źródło klucza obcego, tj. tabelę i klucz podstawowy
- ON DELETE, ON UPDATE – określenie czynności, które należy podjąć jeśli wartość klucza podstawowego zostanie usunięta lub ulegnie zmianie:
  - SET NULL zastąp wartość klucza obcego przez NULL,
  - SET DEFAULT zastąp wartość klucza obcego przez wartość domyślną,
  - CASCADE skasuj lub zmodyfikuj wszystkie wiersze zawierające zmienianą wartość klucza obcego
  - NO ACTION (tylko modyfikacja) nie zmieniaj wartości klucza
  - RESTRICT nie dopuść do zmiany

Zabronione usuwanie w MySQL

ALTER TABLE `zamowienie` TYPE = INNODB;

ALTER TABLE `klient` TYPE = INNODB;

ALTER TABLE Zamowienie ADD FOREIGN KEY (IdKlienta)
REFERENCES Klient (IdKlienta) ON DELETE RESTRICT ON UPDATE RESTRICT;

DELETE FROM `klient` WHERE `IdKlienta` =1 LIMIT 1

#1217 - Cannot delete a parent row: a foreign key constraint fails

IdKlienta	NazwaKlienta	Adres
1	Klient 1	*NULL*
2	Sklep Ogólny	Zabierzów ul. Spokojna 2
3	Firma Kruk	Modlniczka 127
4	Fabryka Rowerów	Kraków ul. Bracka 1
5	Fabryka Mebli	Zabierzów, ul. Krakowska 12
6	Hurtownia Materia³ów Budowlanych	Kraków ul.Wielicka 20
7	Fabryka wózków dzieciêcych	Krzeszowice ul, Krakowska 5
10	Klient 5	*NULL*
11	Klient 2	*NULL*
12	Klient 3	*NULL*

IdZamowienia	DataZamowienia	NrZamowienia	IdKlienta
1	2005-03-27	2005/127	1
2	2005-03-27	2005/128	2
59	2005-03-30	2005/201	4
60	2005-03-30	2005/202	4
61	2005-03-30	2005/203	4
62	2005-03-30	2005/204	4
63	2005-03-30	2005/20	4

Kaskadowa aktualizacja w MySQL

ALTER TABLE Zamowienie ADD FOREIGN KEY (IdKlienta)
REFERENCES Klient(IdKlienta)
ON DELETE CASCADE ON UPDATE CASCADE

DELETE FROM `klient` WHERE `IdKlienta` =1

IdKlienta	NazwaKlienta	Adres
2	Sklep Ogólny	Zabierzów ul. Spokojna 2
3	Firma Kruk	Modlniczka 127
4	Fabryka Rowerów	Kraków ul. Bracka 1
5	Fabryka Mebli	Zabierzów, ul. Krakowska 12
6	Hurtownia Materia³ów Budowlanych	Kraków ul.Wielicka 20
7	Fabryka wózków dzieciêcych	Krzeszowice ul, Krakowska 5
10	Klient 5	*NULL*
11	Klient 2	*NULL*
12	Klient 3	*NULL*

IdZamowienia	DataZamowienia	NrZamowienia	IdKlienta
2	2005-03-27	2005/128	2
59	2005-03-30	2005/201	4
60	2005-03-30	2005/202	4
61	2005-03-30	2005/203	4
62	2005-03-30	2005/204	4
63	2005-03-30	2005/an	4
66	2005-03-30	2005/303	6
67	2005-04-02	2005/876	10

Porządkowanie kluczy w MySQL

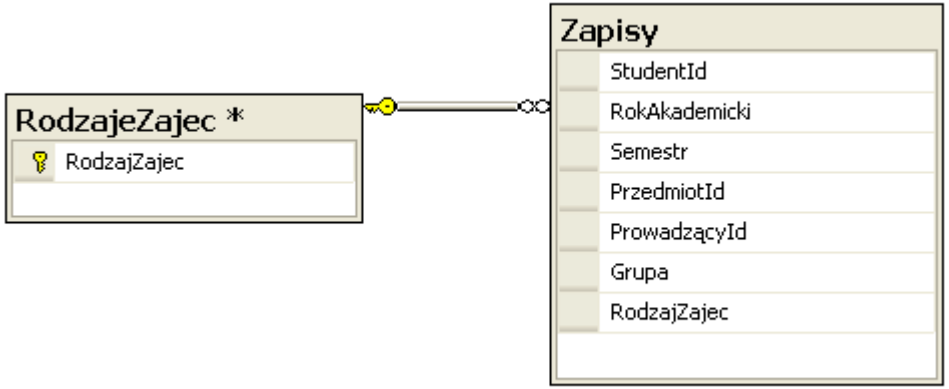
- Zmiana wartości klucza w tabeli rodzicielskiej powoduje odpowiednie zmiany w tabelach dzieciach

UPDATE `klient` SET `IdKlienta` = '8‘ WHERE `IdKlienta` =10;

IdKlienta	NazwaKlienta	Adres
2	Sklep Ogólny	Zabierzów ul. Spokojna 2
3	Firma Kruk	Modlniczka 127
4	Fabryka Rowerów	Kraków ul. Bracka 1
5	Fabryka Mebli	Zabierzów, ul. Krakowska 12
6	Hurtownia Materia³ów Budowlanych	Kraków ul.Wielicka 20
7	Fabryka wózków dzieciêcych	Krzeszowice ul, Krakowska 5
8	Klient 5	*NULL*

IdZamowienia	DataZamowienia	NrZamowienia	IdKlienta
2	2005-03-27	2005/128	2
59	2005-03-30	2005/201	4
60	2005-03-30	2005/202	4
61	2005-03-30	2005/203	4
62	2005-03-30	2005/204	4
63	2005-03-30	2005/an	4
66	2005-03-30	2005/303	6
67	2005-04-02	2005/876	8
68	2005-04-02	2005/876	8
69	2005-04-02	2005/876	8

Kaskadowa aktualizacja w MS SQL




```
Alter table [Zapisy] add  foreign
   key([RodzajZajec]) references
   [RodzajeZajec] ([RodzajZajec])  on
   update cascade on delete cascade
```

STACJONARNY\S... - dbo.Z

	RodzajZajec
	ćwiczenia
	wykład
▶*	NULL

	StudentId	RokAkademicki	Semestr	PrzedmiotId	ProwadzącyId	Grupa	RodzajZajec
	2	2007/2008	zimowy	1	3	1	ćwiczenia
	3	2007/2008	zimowy	1	3	1	ćwiczenia
	2	2007/2008	zimowy	1	2	0	wykład
	3	2007/2008	zimowy	1	2	0	wykład
▶*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

STACJONARNY\S... - dbo

	RodzajZajec
	ćwiczenia
	Wykład
▶*	NULL

STACJONARNY\S... - dbo.Zapisy | STACJONARNY\S....RodzajeZajec | Summary

	StudentId	RokAkademicki	Semestr	PrzedmiotId	ProwadzącyId	Grupa	RodzajZajec
	2	2007/2008	zimowy	1	3	1	ćwiczenia
	3	2007/2008	zimowy	1	3	1	ćwiczenia
	2	2007/2008	zimowy	1	2	0	Wykład
	3	2007/2008	zimowy	1	2	0	Wykład
▶*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Struktura baz danych

Dodatkowe elementy

Perspektywy (ang. Views) – brak w MySQL 4.0

- Zapytanie posiadające własną nazwę i przechowywane w słowniku danych
- Perspektywy tworzymy po to by:
  - zapisać często wykonywane złożone zapytania
  - stworzyć logiczne modele dla różnych użytkowników
  - zwiększyć bezpieczeństwo danych

Tworzenie perspektyw

```
CREATE VIEW PodgladZamowienia
SELECT     Klient.NazwaKlienta, Zamowienie.NrZamowienia,
Zamowienie.datazamowienia, Towar.NazwaTowaru,
LiniaZamowienia.Ilosc, LiniaZamowienia.Cena
FROM       Zamowienie INNER JOIN
Klient ON Zamowienie.IdKlienta = Klient.IdKlienta INNER
JOIN
LiniaZamowienia ON Zamowienie.IdZamowienia =
LiniaZamowienia.IdZamowienia INNER JOIN
Towar ON LiniaZamowienia.IdTowaru = Towar.IdTowaru
```

Wykorzystanie perspektyw

```
SELECT    NazwaKlienta, NazwaTowaru, Ilosc, Cena
FROM      PodgladZamowienia
WHERE     (NazwaKlienta = 'FH Klin SA')
```

Nazwaklienta	NazwaTowaru	Ilosc	cena
FH Klin SA	Rura zgrz. fi 6,3 gr 0,2	20	1.50
FH Klin SA	Rura zgrz. fi 12,6 gr 0,2	12	1.75
FH Klin SA	Rura zgrz. fi 6,3 gr 0,3	25	2.10

Modyfikowalność perspektyw

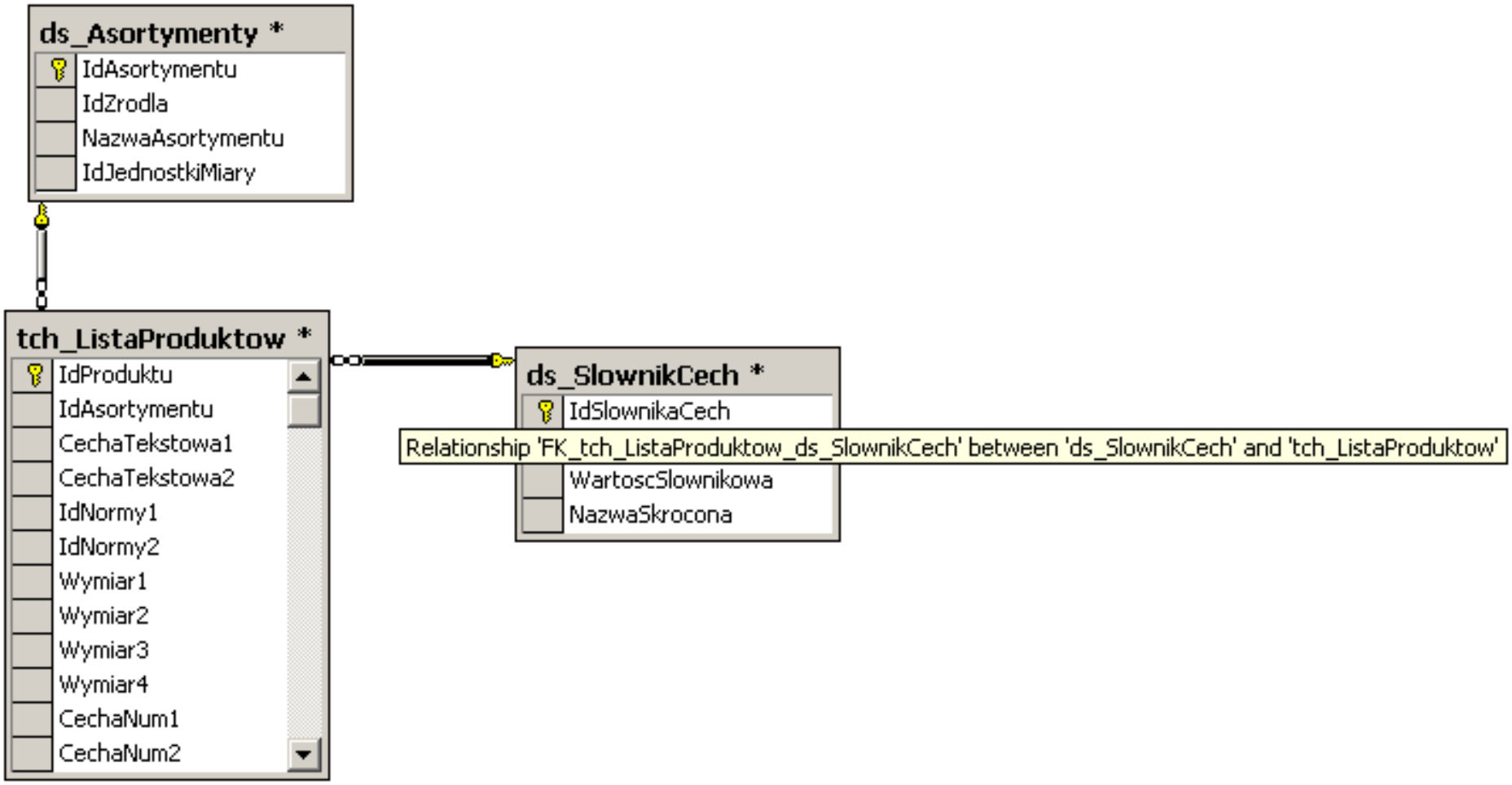
- Zgodnie ze standardem SQL-92 można modyfikować wyłącznie takie perspektywy, które:
  - nie zawierają złączeń,
  - są pojedyncze (np. unie nie są dopuszczalne),
  - nie mogą być oparte na zapytaniu grupującym lub zawierającym słowo DISTINCT,
  - nie można modyfikować kolumn wyliczonych

Tabele tymczasowe

- istnieją wyłącznie w trakcie trwania sesji
- obsługuje się je identycznie jak tabele stałe
- są znacznie szybciej obsługiwane niż zapytania czy perspektywy ale nie są automatycznie modyfikowane
- w przeciwieństwie do widoków są w pełni modyfikowalne

Edycja złożenia

- Edycja złożenia i perspektywy zawierającej elementy pochodzące z więcej iż jednej tabeli nie jest możliwa wprost
- Można natomiast stworzyć tabelę tymczasową zawierającą kody i treść atrybutów z połączonych tabel i ją edytować


ds_Asortymenty *
IdAsortymentu
IdZrodla
NazwaAsortymentu
IdJednostkiMiary
tch_ListaProduktow *
IdProduktu
IdAsortymentu
CechaTekstowa1
CechaTekstowa2
IdNormy1
IdNormy2
Wymiar1
Wymiar2
Wymiar3
Wymiar4
CechaNum1
CechaNum2
ds_SlownikCech *
IdSlownikaCech
WartoscSlownikowa
NazwaSkrocona
Relationship 'FK_tch_ListaProduktow_ds_SlownikCech' between 'ds_SlownikCech' and 'tch_ListaProduktow'

Tworzenie tabeli tymczasowej w MS SQL

```
CREATE TABLE #tmp (
NazwaAsortymentu char(255),
WartoscSlownikowa char(255),
IdSlownikaCech bigint;
Wymiar1 float,
Wymiar2 float,
wymiar3 float);
```

Przenoszenie złożenia to tabeli

```
INSERT INTO #tmp
SELECT     dbo.ds_Asortymenty.NazwaAsortymentu,
dbo.ds_SlownikCech.WartoscSlownikowa,
dbo.tch_ListaProduktow.Wymiar1,
                    dbo.tch_ListaProduktow.Wymiar2,
dbo.tch_ListaProduktow.Wymiar3,
dbo.ds_SlownikCech.IdSlownikaCech,
                              dbo.tch_ListaProduktow.IdProduktu
FROM         dbo.tch_ListaProduktow INNER JOIN
                    dbo.ds_SlownikCech ON
dbo.tch_ListaProduktow.CechaTekstowa1 =
dbo.ds_SlownikCech.IdSlownikaCech INNER JOIN
                    dbo.ds_Asortymenty ON
dbo.tch_ListaProduktow.IdAsortymentu =
dbo.ds_Asortymenty.IdAsortymentu;
```

Edycja tabeli tymczasowej

- W tabeli możemy zmienić dane posługując się nazwami pochodzącymi w systemie z różnych tabel
- Po edycji można odtworzyć tabele źródłowe

```
DECLARE @indeks bigint;

SELECT     @indeks= dbo.ds_SlownikCech.IdSlownikaCech
FROM       dbo.ds_SlownikCech
WHERE
dbo.ds_SlownikCech.WartoscSlownikowa='eliptyczna';

UPDATE #tmp SET IdSlownikaCech=@indeks,
WartoscSlownikowa='eliptyczna' WHERE
NazwaAsortymentu='Rura profilowa' AND
WartoscSlownikowa='kroplowa';

UPDATE dbo.tch_ListaProduktow SET
CechaTekstowa1=t.IdSlownikaCech
FROM #tmp t WHERE
dbo.tch_ListaProduktow.IdProduktu=t.IdProduktu;
```

NazwaAsortymentu	WartoscSlownikowa	Wymiar1	Wymiar2	Wymiar3	IdSlo...	IdProdu
Rura profilowa	eliptyczna	34.0	17.5	1.0	4	6978
Rura profilowa	eliptyczna	34.0	17.5	1.2	4	6979
Rura profilowa	eliptyczna	31.0	17.5	1.0	4	6980
Rura profilowa	eliptyczna	37.0	21.0	2.5	4	6981
Rura profilowa	eliptyczna	44.0	32.0	1.2	4	6982
Rura profilowa	eliptyczna	37.0	21.0	2.5	4	2939
Rura profilowa	eliptyczna	34.0	17.5	1.0	4	2940
Rura profilowa	eliptyczna	34.0	17.5	1.2	4	2941
Rura profilowa	kroplowa	81.0	37.0	1.2	5	2944
Rura b/s okreslonego zastosowan...	kroplowa	56.0	30.0	1.0	5	2961
Rura b/s okreslonego zastosowan...	kroplowa	81.0	34.0	2.5	5	2962
Rura b/s okreslonego zastosowan...	kroplowa	105.0	50.0	2.5	5	2963

NazwaAsortymentu	WartoscSlown...	Wymiar1	Wymiar2	Wymiar3	IdSlown...	IdProdu
Rura profilowa	eliptyczna	34.0	17.5	1.0	4	6978
Rura profilowa	eliptyczna	34.0	17.5	1.2	4	6979
Rura profilowa	eliptyczna	31.0	17.5	1.0	4	6980
Rura profilowa	eliptyczna	37.0	21.0	2.5	4	6981
Rura profilowa	eliptyczna	44.0	32.0	1.2	4	6982
Rura profilowa	eliptyczna	37.0	21.0	2.5	4	2939
Rura profilowa	eliptyczna	34.0	17.5	1.0	4	2940
Rura profilowa	eliptyczna	34.0	17.5	1.2	4	2941
Rura profilowa	eliptyczna	81.0	37.0	1.2	4	2944
Rura b/s okreslonego zastosowa...	kroplowa	56.0	30.0	1.0	5	2961
Rura b/s okreslonego zastosowa...	kroplowa	81.0	34.0	2.5	5	2962
Rura b/s okreslonego zastosowa...	kroplowa	105.0	50.0	2.5	5	2963

Przyznawanie praw dostępu

- Serwer baz danych może obsługiwać wielu użytkowników identyfikowanych przez nazwę i hasło
- Nie każdy z użytkowników musi mieć równe prawa
- W momencie utworzenia nowego elementu bazy danych aktualny użytkownik staje się jego właścicielem
- Właściciel może nadawać i odbierać prawa innym użytkownikom

Przyznawanie praw dostępu składnia SQL99

```
GRANT  {ALL [PRIVILEGES] }
| SELECT
| INSERT [ nazwa_kolumny [,...n]) ]
| DELETE
| UPDATE [ nazwa_kolumny [,...n]) ]
| REFERENCES [ nazwa_kolumny [,...n]) ]
| USAGE } [,...n]
ON { [TABLE] nazwa_tabeli
| DOMAIN nazwa_domeny
| COLLATION nazwa_zestawienia
CHARACTER SET nazwa_zestawu_znaków
| TRANSLATION nazwa_translacji }
TO {nazwa_podmiotu | PUBLIC}
[WITH GRANT OPTION]
```

Przykład MySQL

- Aktualny użytkownik, posiadający wszelkie uprawnienia przekazuje część uprawnień użytkownikowi ***andrzej*** na serwerze ***localhost***
- Następnie odbiera mu wszelkie uprawnienia i przekazuje inne

haslo

GRANT CREATE , DROP , INDEX , ALTER , CREATE TEMPORARY TABLES ON * . * TO 'andrzej'@'localhost' IDENTIFIED BY '***' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0** ;

GRANT ALL PRIVILEGES ON `sprzedaz` . * TO 'andrzej'@'localhost' WITH GRANT OPTION ;

REVOKE ALL PRIVILEGES ON * . * FROM 'andrzej'@'localhost';

GRANT SELECT ,
INSERT ,
UPDATE ,
DELETE ,
RELOAD ,
SHUTDOWN ,
PROCESS ,
FILE ,
REFERENCES ,
SHOW DATABASES ,
SUPER ,
LOCK TABLES ,
EXECUTE ,
REPLICATION SLAVE ,
REPLICATION CLIENT ON * . * TO 'andrzej'@ 'localhost' WITH GRANT OPTION MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 ;

Nowy użytkownik

- Logowanie nowego użytkownika

- Użytkownik ma prawa do edycji danych, nie może jednak zmieniać struktur bazy danych

```
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 155 to server version: 4.1.9-max

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use planowanie;
Database changed
mysql> create table test (
    -> id char(10));
ERROR 1044 (42000): Access denied for user 'andrzej'@'localhost' to database 'pl
anowanie'
mysql> select * from towary;
+----------+--------------------+------+
| IdTowaru | NazwaTowaru        | cena |
+----------+--------------------+------+
|        1 | Rura stalowa fi 10 | 0.00 |
|        2 | Rura b/s fi 12     | 1.30 |
|        3 | Rura b/s fi 15     | 1.20 |
|       10 | rura b/s fi 32     | 1.10 |
|        6 | Z|▒czka fi 10      | 2.00 |
+----------+--------------------+------+
5 rows in set (0.41 sec)

mysql> update towary set cena=1.5 where idtowaru=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from towary;
+----------+--------------------+------+
| IdTowaru | NazwaTowaru        | cena |
+----------+--------------------+------+
|        1 | Rura stalowa fi 10 | 1.50 |
|        2 | Rura b/s fi 12     | 1.30 |
|        3 | Rura b/s fi 15     | 1.20 |
|       10 | rura b/s fi 32     | 1.10 |
|        6 | Z|▒czka fi 10      | 2.00 |
+----------+--------------------+------+
5 rows in set (0.00 sec)

mysql>
```

brak uprawnień do tworzenia tabeli

są uprawnienia do aktualizacji tabeli

Transakcje

- W systemie baz danych z wieloma użytkownikami transakcja jest niepodzielną, spójną, izolowaną i trwałą (ACID-atomic, consistent, isolatable, and durable) procedurą realizującą dostęp do danych
- Podczas realizacji transakcji możliwe jest wykorzystanie zmiennych oraz blokowanie dostępu do danych
- Transakcje mogą być wycofane

Transakcje składnia

- START (BEGIN) TRANSACTION;
- ***polecenia***
- COMMIT – zatwierdzenie transakcji
- lub
- ROLLBACK – wycofanie transakcji

Transakcje w MS SQL

- Użycie tabeli w trakcie transakcji blokuje dostęp do tabeli
- Polecenie ROLLBACK kasuje działania wykonane podczas transakcji
- Polecenie COMMIT zatwierdza działania

	IdProduktu	IdAsortymentu	CechaTekstowa1	CechaTekstowa2
▶	2944	25	5	2048

```
BEGIN TRANSACTION;
UPDATE    tch_ListaProduktow
SET        CechaTekstowa1 = 4
WHERE    (IdProduktu = 2944);
ROLLBACK;
```

	IdProduktu	IdAsortymentu	CechaTekstowa1	CechaTekstowa2
▶	2944	25	5	2048

```
BEGIN TRANSACTION;
UPDATE    tch_ListaProduktow
SET        CechaTekstowa1 = 4
WHERE    (IdProduktu = 2944);
COMMIT;
```

	IdProduktu	IdAsortymentu	CechaTekstowa1	CechaTekstowa2
▶	2944	25	4	2048

Transakcje w MySQL

- Potwierdzanie lub wycofywanie transakcji możliwe jest tylko w przypadku użycia mechanizmu obsługi pamięci InnoDB

Procedury składowane

- Specyficzny język umożliwiający wykonywanie wielu poleceń SQL oraz wprowadzający dynamikę

```
CREATE PROCEDURE procedura
@Asortyment char(255),
@Cecha char(255)
 AS
SELECT dbo.ds_Asortymenty.NazwaAsortymentu,
dbo.ds_SlownikCech.WartoscSlownikowa, dbo.tch_ListaProduktow.Wymiar1,
                  dbo.tch_ListaProduktow.Wymiar2,
dbo.tch_ListaProduktow.Wymiar3, dbo.ds_SlownikCech.IdSlownikaCech,
                                    dbo.tch_ListaProduktow.IdProduktu
FROM        dbo.tch_ListaProduktow INNER JOIN
                  dbo.ds_SlownikCech ON dbo.tch_ListaProduktow.CechaTekstowa1 =
dbo.ds_SlownikCech.IdSlownikaCech INNER JOIN
                  dbo.ds_Asortymenty ON dbo.tch_ListaProduktow.IdAsortymentu =
dbo.ds_Asortymenty.IdAsortymentu
WHERE ds_Asortymenty.NazwaAsortymentu=@asortyment  AND
dbo.ds_SlownikCech.WartoscSlownikowa=@Cecha
GO
```

```
procedura 'Rura profilowa', 'eliptyczna'
```

	NazwaAsortymentu	WartoscSlownikowa	Wymiar1	Wymiar2	Wymiar3	IdSlownikaCech	IdProduktu
1	Rura profilowa	eliptyczna	37.0	21.0	2.5	4	2939
2	Rura profilowa	eliptyczna	34.0	17.5	1.0	4	2940
3	Rura profilowa	eliptyczna	34.0	17.5	1.2	4	2941
4	Rura profilowa	eliptyczna	42.0	34.0	1.2	4	6927
5	Rura profilowa	eliptyczna	44.0	32.0	1.2	4	6928
6	Rura profilowa	eliptyczna	56.0	30.0	1.2	4	6929
7	Rura profilowa	eliptyczna	65.0	30.0	1.2	4	6930
8	Rura profilowa	eliptyczna	34.0	17.5	1.0	4	6978
9	Rura profilowa	eliptyczna	34.0	17.5	1.2	4	6979
10	Rura profilowa	eliptyczna	31.0	17.5	1.0	4	6980
11	Rura profilowa	eliptyczna	37.0	21.0	2.5	4	6981
12	Rura profilowa	eliptyczna	44.0	32.0	1.2	4	6982

Wyzwalacze w SQL 99

- wyzwalacz (ang. trigger) jest specjalnym rodzajem składowanej procedury, która jest uruchamiana automatycznie podczas wykonywania instrukcji modyfikacji danych
- wyzwalacz jest powiązany z konkretną instrukcją modyfikacji danych (INSERT, UPDATE, DELETE)
- wyzwalacz jest uruchamiany przed modyfikacją, po lub zamiast modyfikacji

Wyzwalacze SQL 99

CREATE TRIGGER nazwa_wyzwalacza
{ BEFORE | AFTER } { [DELETE] | [NSERT] [UPDATE]
[OF kolumna [,...n]}
ON nazwa_tabeli
[REFERENCING {OLD [ROW] [AS]
nazwa_starej_krotki | NEW [ROW] [AS]
nazwa_nowej_krotki OLD TABLE [AS]
nazwa_starej_tabeli | NEW TABLE [AS]
nazwa_nowej_tabeli}]
{FOR EACH {ROW | STATEMENT}]
[WHEN (warunki)]
blok_kodu

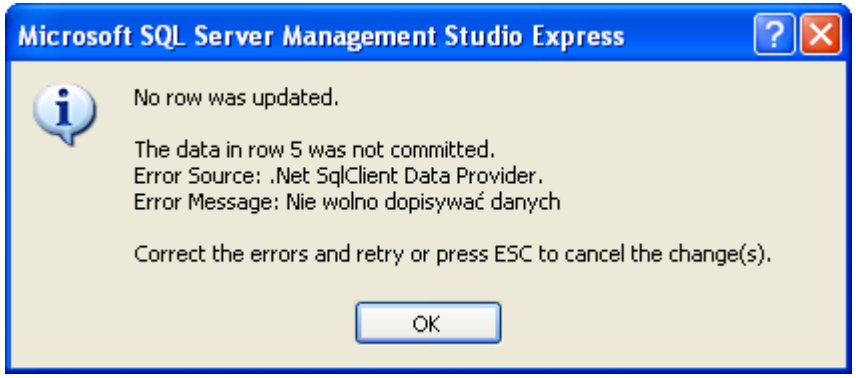
Wyzwalacze SQL 99

```
CREATE TRIGGER dopisanie_rekordu
INSTEAD OF INSERT SymbolTowaru, NazwaTowaru
ON Towar
REFERENCING
     OLD AS StaraKrotka
     NEW AS NowaKrotka
     OLD_TABLE Stare
WHEN NowaKrotka.NazwaTtowaru IN
      Stare;
      UPDATE Towar
      SET NazwaTowaru = StaraKrotka.NazwaTowaru;
```

Wyzwalacz w MS SQL

```
USE test
   DROP TRIGGER uwaga
GO
CREATE TRIGGER uwaga
ON Przedmioty
AFTER INSERT, UPDATE
AS RAISERROR ('Nie wolno dopisywaæ danych', 16, 10)
GO
```

Wyzwalacz w MS SQL



Archiwacja w MS SQL

```
CREATE TRIGGER archiwacja ON ds_asortymenty
FOR DELETE
AS
INSERT INTO ds_asortymenty_arch
SELECT *
FROM deleted
```

Stan wyjściowy

ds-asortymenty

IdAsortymentu	IdZrodla	NazwaAsortymentu	IdJednostkiMiary
1	0	Krąg kupny	5
2	1	Taśma po perforacji	5
3	1	Krąg przycięty na wymiar	5
4	0	Lupa	5
5	1	Lupa przycięta na wymiar	5
6	9	R.wysokostop.z/s ciagniona na zimno	5
7	9	Rura b/s	5
8	9	Rura b/s do pracy w podwyższ.temp.	5
9	9	Rura b/s kotłowa	5
10	9	Rura b/s ogolnego przeznaczenia	5
11	9	Rura b/s okreslonego zastosowania	5
12	9	Rura b/s okretowa	5
13	9	Rura b/s precyzyjna	5
14	9	Rura b/s wysokostop.austen.-ferryt.	5
15	9	Rura b/s wysokostop.ciagn.na zimno	5
16	9	Rura b/s wysokostop.walcow.na zimno	5
17	9	Rura b/s wysokostopowa	5
18	9	Rura b/s ze st.żaroodp.i na zb.cis.	5
19	9	Rura b/s ze stali zaroodpornych	5

Efekt usuwania

DELETE FROM ds_asortymenty WHERE IdAsortymentu<4

ds-asortymenty

	IdAsortymentu	IdZrodla	NazwaAsortymentu	IdJednostkiMiary
▶	4	0	Lupa	5
	5	1	Lupa przycięta na wymiar	5
	6	9	R.wysokostop.z/s ciagniona na zimno	5
	7	9	Rura b/s	5
	8	9	Rura b/s do pracy w podwyższ.temp.	5
	9	9	Rura b/s kotłowa	5
	10	9	Rura b/s ogolnego przeznaczenia	5
	11	9	Rura b/s okreslonego zastosowania	5
	12	9	Rura b/s okretowa	5
	13	9	Rura b/s precyzyjna	5

ds_asortymenty_arch

	IdAsortymentu	IdZrodla	NazwaAsortymentu	IdJednostkiMiary
▶	1	0	Krąg kupny	5
	2	1	Taśma po perforacji	5
	3	1	Krąg przycięty na wymiar	5
*				

Inne narzędzia proceduralne

- języki czwartej generacji 4GL
- graficzne interfejsy użytkownika GUI (Graphical User Interface)
- interfejsy typu QBE (Qery By Example)
- interfejsy języka naturalnego
- osadzony SQL
- interfejsy programów użytkowych API (application programming interface)

Transact - SQL

- Pracochłonne zapytania np. aktualizujące dane można zastąpić procedurami składowanymi

```
use buczek
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE Dopisywanie

        @Nazwa char(255),
        @KodPocztowy char(255),
        @Miejscowosc char(255)
AS
BEGIN

insert into ds_klienci (Nazwa, KodPocztowy, Miejscowosc)
values (@Nazwa, @KodPocztowy, @Miejscowosc)
END
GO
```

dopisywanie 'Firma A', '30-150', 'Kraków'

Id	Nazwa	KodPocztowy	Miejscowosc	Ulica	N
968	"PAHER" SJ PA...	47110	KOLONOWSKIE	DLUGA 2	7
969	ZAKLADY SKLEJ...	85957	BYDGOSZCZ	PRZEMYSLOWA 8	/\
970	ORZECHOWSKI...	62322	ORZECHOWO	MILOSLAWSKA 13	/\
971	ZAKLADY PLYT P...	64700	CZARNKOW	PRZEMYSLOWA 2	/\
972	ZAKLADY PLYT P...	37700	PRZEMYSL	OFIAR KATYNIA...	7
973	ZAKLADY REMO...	42230	KONIECPOL	TARCHALSKIEGO 2	5
974	KONTRAHENT D...	*NULL*	*NULL*	*NULL*	/\
975	"INTER-COMER...	2414	WARSZAWA	KS.CHROSCICKI...	5
976	SLASKIE ZAKLA...	43100	TYCHY	KATOWICKA 182	/\
977	KOSTRZYNSKIE ...	66470	KOSTRZYN ODR...	FABRYCZNA 1	/\
978	ZAKLADY PAPIE...	47300	KRAPKOWICE	OPOLSKA 103	7
979	MYSZKOWSKIE ...	42300	MYSZKOW	M.POZNANSKIEJ 6	/\
980	NIEDOMICKIE Z...	33132	NIEDOMICE K/T...	*NULL*	/\
981	"INTERCELL" S.A.	7401	OSTROLEKA	I AWP 21	7
982	"WOLCO"SP.Z.O.O	20148	LUBLIN	ZWIAZKOWA 23	/\
983	ZAKLADY MECH...	47300	KRAPKOWICE	OPOLSKA 79A	7
984	CENTRALNE BIU...	93176	LODZ	SUWALSKA 25/27	/\
985	ZAKLADY CELUL...	82500	KWIDZYN	LOTNICZA 1	/\
986	FIRMA PRODUK...	42280	CZESTOCHOWA	MALA WARSZA...	5
987	ZAKLADY MASZY...	17200	HAJNOWKA	A.KRAJOWEJ 3	5
988	CHADZYNSKI KA...	41200	SOSNOWIEC	RADOSNA 30	/\
989	WROCLAWSKI Z...	54204	WROCLAW	LEGNICKA 62	8
990	ZAKLADY NAPR...	5800	PRUSZKOW	3-GO MAJA 8	5
991	TYL RYSZARD	41200	SOSNOWIEC	GEN.HALLERA 6...	/\
992	ZAKLADY NAPR...	33310	NOWY SACZ	WYSPIANSKIEGO 3	7
993	ZAKLADY NAPR...	44100	GLIWICE	BL.CZESLAWA 13	6
994	Z-DY NAPRAWC...	45332	OPOLE	REJTANA 7	/\
995	AGENCJA TECH...	40273	KATOWICE	GRANICZNA 57...	6
996	ZAKLADY NAPR...	56400	OLESNICA	MONIUSZKI 20	9
997	ZAKLADY NAPR...	85082	BYDGOSZCZ	ZUGMUNTA AUG...	5
998	ZAKLADY NAPR...	64920	PILA	WARSZTATOWA 8	7
999	ZAKLADY NAPR...	80958	GDANSK	SIENNICKA 25	5
1000	PKP Z-DY NAPR...	73100	STARGARD SZC...	PIERWSZEJ BRY...	8
1001	Firma A ...	30-150 ...	Kraków ...	*NULL*	/\
NULL	*NULL*	*NULL*	*NULL*	*NULL*	/\

Kursor

- Kursor jest tabelą rekordów umożliwiającą dostęp do kolekcji pobranej z tabeli lub złożenia

```
USE buczek
GO
-- Execute the SELECT statement alone to show the
-- full result set that is used by the cursor.
SELECT Nazwa, Miejscowosc FROM ds_klienci
ORDER BY Nazwa
-- Declare the cursor.
DECLARE kursor SCROLL CURSOR FOR
SELECT Nazwa, Miejscowosc  FROM ds_klienci
ORDER BY Nazwa
OPEN kursor
-- Fetch the last row in the cursor.
FETCH LAST FROM kursor
-- Fetch the row immediately prior to the current row in the cursor.
FETCH PRIOR FROM kursor
```

	Nazwa	Miejscowosc
992	ZJEDNOCZONE ZAKLADY GAZOW TECHNICZNYCH 'POLGAZ'	GLIWICE
993	ZKLADY CHEMICZNE "POLICE" S.A.	POLICE
994	ZRZESZENIE PRZEDS.HANDLU ART.TECHNICZNYMI "ELMET"	KATOWICE
995	ZRZESZENIE PRZEM. CIAGN. "URSUS" Z-DY. MECHAN. WLOCLA...	WLOCLAWEK
996	ZRZESZENIE PRZEMYSLU CIAGNIKOWEGO "URSUS"	WARSZAWA URSUS
997	ZULKOWSKI WITOLD	SOSNOWIEC
998	ZYRARDOWSKIE Z-DY PRZEMYSLU SPIRYTUSOWEGO "POLMOS"	ZYRARDOW
999	ZYWIECKA F-KA SPRZETU SZPITALNEGO "FAMED" S.A.	ZYWIEC

	Nazwa	Miejscowosc
1	ZYWIECKA F-KA SPRZETU SZPITALNEGO "FAMED" S.A.	ZYWIEC

	Nazwa	Miejscowosc
1	ZYRARDOWSKIE Z-DY PRZEMYSLU SPIRYTUSOWEGO "POLMOS"	ZYRARDOW

```
-- Fetch the second row in the cursor.
FETCH ABSOLUTE 2 FROM kursor
-- Fetch the row that is three rows after the current row.
FETCH RELATIVE 3 FROM kursor
-- Fetch the row that is two rows prior to the current row.
FETCH RELATIVE -2 FROM kursor
CLOSE kursor
DEALLOCATE kursor
GO
```

	Nazwa	Miejscowosc
1	"AGROMET" FABRYKA MASZYN ROLNICZYCH	BRZEG
2	"AGROMET" FABRYKA MASZYN ROLNICZYCH	LUBLIN
3	"AGROMET" FABRYKA MASZYN ROLNICZYCH	MALBORK
4	"AGROMET" PRZED.BUD-MONTAZOWE PRZEMYSLU MASZYN R...	WROCLAW 2
5	"AGROMET"FABRYKA MASZYN ROLNICZYCH	CZARNA BIALOSTOCKA
6	"AGROMET"ZEHS LUBAN SP. Z O.O.	LUBAN
7	"AGROMET-PIONIER"FABRYKA MASZYN I URZADZEN	STRZELCE OPOLSKIE
8	"AGROMET-UNIA" F-KA MASZYN ROLNICZYCH Z-D NR 1	GIDLE

	Nazwa	Miejscowosc
1	ZYWIECKA F-KA SPRZETU SZPITALNEGO "FAMED" S.A.	ZYWIEC

	Nazwa	Miejscowosc
1	ZYRARDOWSKIE Z-DY PRZEMYSLU SPIRYTUSOWEGO "POLMOS"	ZYRARDOW

	Nazwa	Miejscowosc
1	"AGROMET" FABRYKA MASZYN ROLNICZYCH	LUBLIN

	Nazwa	Miejscowosc
1	"AGROMET"FABRYKA MASZYN ROLNICZYCH	CZARNA BIALOSTOCKA

	Nazwa	Miejscowosc
1	"AGROMET" FABRYKA MASZYN ROLNICZYCH	MALBORK

Wykorzystanie kursora

```
SELECT     DataZamowienia, SUM(DISTINCT Ilosc) AS razem
FROM       Sprzedaz
GROUP BY DataZamowienia
ORDER BY DataZamowienia
```

	DataZamowienia	razem
	2005-09-26 15:52:09	1120
	2005-09-27 00:00:00	1000
	2005-09-28 00:00:00	900

```
GO
DECLARE @Il real, @Poprzedni real, @Dynamika real
DECLARE kursor CURSOR FOR
SELECT     SUM(DISTINCT Ilosc) AS razem
FROM         Sprzedaz
GROUP BY DataZamowienia
ORDER BY DataZamowienia
OPEN kursor
FETCH NEXT FROM kursor
INTO @Poprzedni
WHILE @@FETCH_STATUS = 0
BEGIN
   FETCH NEXT FROM kursor
   INTO @Il
   PRINT @Il
   PRINT @Poprzedni
   PRINT @Il/@Poprzedni
   SET @Poprzedni=@Il
END
```

	DataZamowienia	razem
	2005-09-26 15:52:09	1120
	2005-09-27 00:00:00	1000
	2005-09-28 00:00:00	900

```
1000
1120
0.892857
900
1000
0.9
900
900
1
```

Dopisywanie rekordów i identyfikacja klucza

```
GO
/****** Object:  StoredProcedure [dbo].[InsertZamowienie]    Script
Date: 11/27/2007 18:58:44 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[InsertZamowienie]
  @Klient char(255),
  @Data datetime,
  @Identity bigint
  AS
DECLARE @IdKlienta bigint
SELECT @IdKlienta = IdKlienta FROM Klienci
WHERE Nazwa = @Klient
INSERT INTO Zamowienia (Data, IdKlienta) VALUES(@Data,
@IdKlienta)
SET @Identity = SCOPE_IDENTITY()
Print @Identity
```

Uruchomienie procedury

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE Test
AS
DECLARE @indeks bigint
begin
exec InsertZamowienie 'Firma AS', '2007-11-30', @indeks
PRINT @indeks
end
```

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

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

```
test


(1 row(s) affected)
17
```

	IdZamowienia	Data	IdKlienta
▶	1	2007-10-10 00:00:00	1
	2	2007-10-12 00:00:00	2
	16	2007-11-30 00:00:00	1
	17	2007-11-30 00:00:00	1

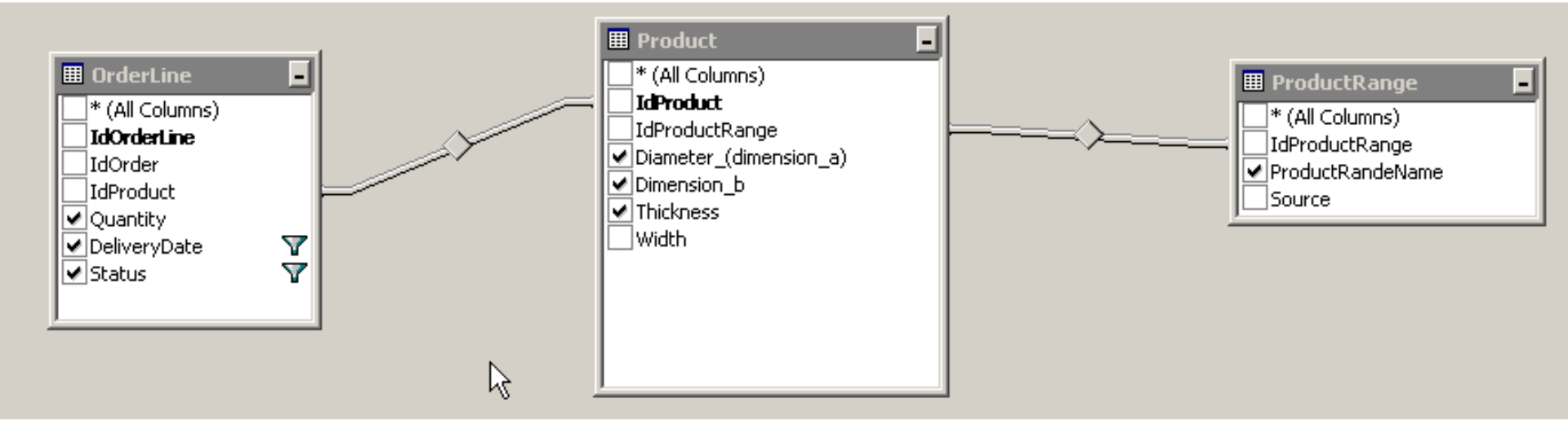
Przykład

- Fabryka rur stalowych produkuje rury z kręgów blachy stalowej w następujących operacjach
  - rozcięcie kręgu (coil cut)
  - perforowanie (perforating) – tylko dla rur perforowanych
  - zgrzewanie
- Zamówienia składają się z wielu linii
- Dla każdej wprowadzonej linii należy opracować kartę technologiczną (Product card)

„Uzbrajanie” linii zamówienia

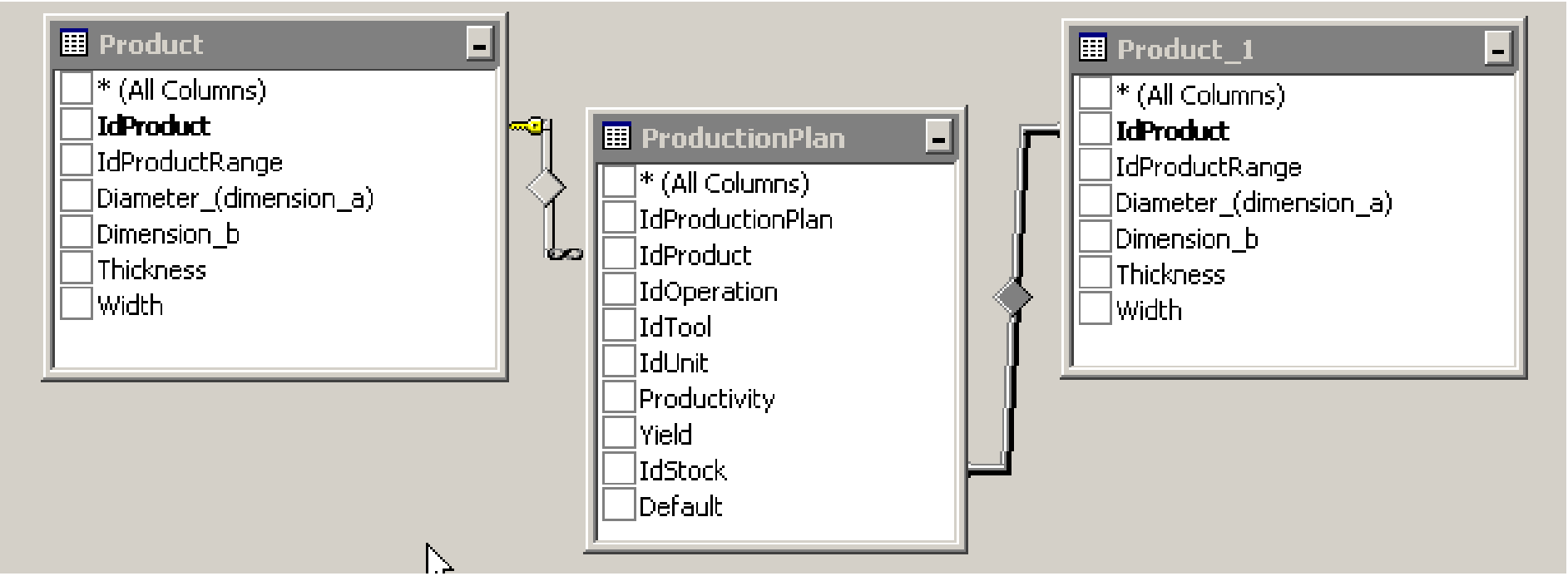
- Każda linia zawiera identyfikator produktu finalnego
- Opis wszystkich faz procesu zapisany jest w tabeli ProductionPlan, przy czym IdStock jest identyfkatorem półproduktu opisanego w tabeli Product
- Ze względu na iterację nie można wygenerować opisu technologii dla linii zamówienia przy pomocy prostego zapytania SQL

Diagram i zawartość linii zamówień



ProductRandeName	Diameter_(dimensic	Dimension_b	Thickness	Quantity	DeliveryDate	Status
A- Perforated welded tube made of aluminium coated steel strip	20	<NULL>	1,5	59	2006-02-03	1
A-Cold calibrated welded steel tube made of CR strip (round section)	57	<NULL>	1,8	120	2006-02-01	1
A-Cold calibrated welded steel tube made of HR strip (square section)	30	30	2,5	78	2006-02-02	1
A-Cold calibrated welded steel tube made of CR strip (round section)	42	<NULL>	3	59	2006-02-03	1
A-Cold calibrated welded steel tube made of CR strip (round section)	56	<NULL>	1,8	110	2006-02-01	1
A-Cold calibrated welded Steel tube made of HR strip (round section)	51	<NULL>	2,5	59	2006-02-03	1
A-Cold calibrated welded steel tube made of CR strip (round section)	56	<NULL>	1,5	58	2006-02-01	1
A-Cold calibrated welded Steel tube made of HR strip (round section)	45	<NULL>	2,5	59	2006-02-03	1
A-Cold calibrated welded steel tube made of HR strip (square section)	20	20	2	140	2006-02-02	1
A-Cold calibrated welded steel tube made of CR strip (round section)	55	<NULL>	1,5	64	2006-02-01	1
A-Cold calibrated welded Steel tube made of HR strip (round section)	43	<NULL>	1,5	59	2006-02-03	1
A-Cold calibrated welded steel tube made of HR strip (square section)	20	20	1,8	44	2006-02-02	1
A- Perforated welded tube made of aluminium coated steel strip	55	<NULL>	1,5	59	2006-02-04	1

Diagram powiązań opisu technologii



Wykorzystanie Transact-SQL

- Język Transact-SQL wprowadza elementy języków proceduralnych takich jak pętla, warunek logiczny, zmienna pamięciowa
- Przykładowa procedura odszukuje w tabeli ProductionPlan wiersz dla operacji, której wynikiem jest wyrób gotowy a następnie w pętli szuka kolejnych półproduktów aż do chwili gdy zapytanie SELECT nie zmienia wartości indeksu półproduktu (@adress1)

```
declare @adress bigint, @adress1 bigint, @stock real, @yield real
SELECT         TOP 1 @adress1=ProductionPlan.IdStock,@adress=OrderLine.IdProduct,
                                    @stock=OrderLine. Quantity, @yield=ProductionPlan.Yield
FROM                 OrderLine INNER JOIN
                  Product ON OrderLine.IdProduct = Product.IdProduct INNER JOIN
                  ProductionPlan ON Product.IdProduct = ProductionPlan.IdProduct
WHERE     orderline.status=1
print 'Ordered product Id:'
Print @adress
print 'First semiproduct Id:'
print @adress1
print 'Yield:'
print @yield
set @stock=@stock/@yield
print 'Stock for first operation:'
print @stock
while @adress1 <> @adress
begin
                  set @adress=@adress1
                  SELECT    @adress1=ProductionPlan.IdStock
                  FROM                 ProductionPlan
                  WHERE     ProductionPlan.IdProduct=@adress
                  SELECT    @yield=ProductionPlan.Yield
                  FROM                 ProductionPlan
                  WHERE     ProductionPlan.IdProduct=@adress
                  if @adress1<> @adress
                  begin
                                    set @stock=@stock/@yield
                                    print 'Next semiproduct Id:'
                                    print @adress1
                                    print 'Yield:'
                                    print @yield
                                    print 'Stock for next operation:'
                                    print @stock
                  end
end
```

Ordered product Id:
43
First semiproduct Id:
999
Yield:
0.95
Stock for first operation:
62.1053
Next semiproduct Id:
937
Yield:
0.9
Stock for next operation:
69.0059
Next semiproduct Id:
1781
Yield:
1
Stock for next operation:
69.0059

Tworzenie karty technologicznej

- Do tabeli ProductCard dopisywane są kolejne rekordy opisujące operacje w ciągu technologicznym

```
declare @adress bigint, @adress1 bigint, @orderid bigint, @stock real, @yield real,
        @operation bigint,@tool bigint,@unit bigint, @productivity real
SELECT     TOP 1 @adress1=ProductionPlan.IdStock,@adress=OrderLine.IdProduct,
                  @stock=OrderLine. Quantity, @yield=ProductionPlan.Yield,
                  @orderid=OrderLine.IdOrderLine, @operation=ProductionPlan.IdOperation,
                  @tool=ProductionPlan.IdTool, @unit=ProductionPlan.IdUnit,
                  @productivity=ProductionPlan.Productivity
FROM            OrderLine INNER JOIN
              Product ON OrderLine.IdProduct = Product.IdProduct INNER JOIN
              ProductionPlan ON Product.IdProduct = ProductionPlan.IdProduct
WHERE    orderline.status=1
INSERT INTO ProductCard (IdOrderLine, IdOperation,IdTool,IdUnit,Productivity,Yield,IdStock)
VALUES                (@orderid, @operation, @tool, @unit, @productivity, @yield, @adress1)
set @stock=@stock/@yield
while @adress1 <> @adress
begin
        set @adress=@adress1
        SELECT                @adress1=ProductionPlan.IdStock, @yield=ProductionPlan.Yield,
                              @operation=ProductionPlan.IdOperation,
                              @tool=ProductionPlan.IdTool, @unit=ProductionPlan.IdUnit,
                              @productivity=ProductionPlan.Productivity
        FROM          ProductionPlan
        WHERE                 ProductionPlan.IdProduct=@adress
        if @adress1<> @adress
        begin
                      INSERT INTO ProductCard (IdOrderLine,
IdOperation,IdTool,IdUnit,Productivity,Yield,IdStock)
                      VALUES                (@orderid, @operation, @tool, @unit, @productivity, @yield,
@adress1)
                      set @stock=@stock/@yield
        end
end
```

	ProductRandeName	Diameter	Thickness	Range1	Thickness	Width	Unit	Operation	Yield
	A-Cold calibrated welded steel tube made of CR strip (round section)	62,2	2	D-Perforated aluminium coated steel strip	1,4	61	Welder W-1,5, W-2	welding	0,95
	A-Cold calibrated welded steel tube made of CR strip (round section)	62,2	2,5	D-Aluminium coated steel strip	1,4	61	Press	perforating	0,9
▶	A-Cold calibrated welded steel tube made of CR strip (round section)	62,2	2,6	E-CR coil	1,4	<NULL	Parting die	coil cut	1
*									