Warszawska
Wyższa Szkoła
Informatyki

Bazy danych – Semestr 2
Zajęcia nr 1

Wprowadzenie

Mgr inż. Jerzy Stankiewicz

PLAN REALIZACJI PROGRAMU NAUCZANIA

W ramach laboratorium studenci wykorzystując język DCL i DDL mają tworzyć i na kolejnych zajęciach rozbudowywać skrypt tworzący bazę danych oraz wybrane mechanizmy wewnętrzne baz danych

Plan zajęć

- Zajęcia 1 - Powtórzenie (przypomnienie / uzupełnienie przerobionego) materiału
- Zajęcia 2 - Uprawnienia - role (serwer-baza danych)
- Zajęcia 3-7 - Instrukcje języka Transact- SQL. Tworzenie skryptu do tworzenia (modyfikacji) bazy danych
  - Zajęcia 3 – wprowadzenie
  - Zajęcia 4 – check (ograniczenie typu danych), reguły, default, widoki, funkcje (daty, agregujące,…), wstawianie - usuwanych danych do/z tabel
  - Zajęcia 5 – wyzwalacze, funkcje skalarne, tabularne
  - Zajęcia 6 – procedury składowane, zmienne, instrukcje: IF, Case, While, kursory – wprowadzenie
  - Zajęcia 7 – kursory – rozwinięcie tematu
- Zajęcia 8 – zaliczenie semestru

ZAKRES MINIMALNYCH WYMAGAŃ DOTYCZĄCYCH WIEDZY I UMIEJĘTNOŚCI PO UKOŃCZENIU PRZEDMIOTU PRZEZ STUDENTA:

- *Umiejętność posługiwania się podstawowym narzędziem MS SQL Serwer 2014 - Microsoft SQL Server Management Studio,*
- *Umiejętność pisania skryptów do tworzenia oraz usuwania bazy danych,*
- *Umiejętność pisania skryptów do wprowadzania oraz usuwania danych do/z bazy danych,*
- *Umiejętność nadawać uprawnienia do bazy danych z wykorzystaniem języka DCL,*
- *Umiejętność stosować wybrane mechanizmy wewnętrzne baz danych.*

Zaliczenie semestru 2

- Obecność na zajęciach
- Aktywne uczestniczenie w zajęciach <*zadania domowe*>
- Wykonanie własnego <*nie obowiązkowy*> projektu bazy danych:
  - Opis projektu (Dokument Word)
  - Skrypty: DDL, DCL <*struktura bazy, mechanizmy wewnętrzne bazy danych, użytkownicy bazy danych i ich uprawnienia, dane testowe bazy danych*>
- Praca końcowa – zaliczenie ćwiczenie praktyczne

Semestr 2
Zajęcia nr 1 - Powtórzenie materiału

- Zaliczenie semestr 1
  - Projekt bazy danych
  - Zapytania (SQL) do bazy danych

Powtórzenie materiału

- **Zaprojektować bazę danych rozgrywek piłkarskich. W bazie powinny znaleźć się informacje:**
  - **drużyny piłkarskie (nazwa, adres, miejscowość),**
  - **trener (trenerzy, gdy jest ich więcej) trenujący obecnie lub w przeszłości drużynę (imię, nazwisko, data urodzenia),**
  - **kolejki rozgrywek (numer kolejki, nazwa kolejki, dzień lub zakres dni kolejki),**
  - **mecze (kto, z kim, wynik meczu, data meczu),**
  - **zdarzenia: w bazie należy umieścić informacje o zdarzeniach (np.: czy były i ilość czerwonych kartek, ilość żółtych kartek, ilość rzutów karnych, zdarzenie przerwany mecz - w której minucie?", zdarzenie mecz bez publiczności" ... itp.)**

- **Projekt - powinien zawierać nazwy tabel, wykaz kolumn, typ danych, pola kluczowe, powiązania między tabelami.**

Powtórzenie materiału

Zaprojektować bazę danych rozgrywek piłkarskich. W bazie powinny znaleźć się informacje:

- **drużyny piłkarskie (nazwa, adres, miejscowość),**
- **trener (trenerzy, gdy jest ich więcej), trenujący obecnie lub w przeszłości drużynę**
- **kolejki rozgrywek (numer kolejki, nazwa kolejki, dzień / dni),**
- **mecze, (kto z kim, wynik meczu, data meczu),**
- **zdarzenia: w bazie należy umieścić informacje o zdarzeniach (np.: czy były i ilość czerwonych, ilość żółtych kartek, ilość rzutów karnych, „przerwany mecz - w której minucie?”, „mecz bez publiczności” ...itp.)**

Legia Warszawa – Górnik Zabrze

Jeden z meczy 4 kolejki (4 kolejka odbyła się w dniach 19-20.11.2015r) w którym padł wynik 2:1

W czasie tego meczu były zdarzenia:

- **2 żółte kartki**
- **1 czerwona kartka**
- **Rzut karny**
- **Mecz odbył się bez udziału publiczności**

Powtórzenie materiału

Zaprojektować bazę danych rozgrywek piłkarskich. W bazie powinny znaleźć się informacje:

- **drużyny piłkarskie (nazwa, adres, miejscowość),**
- **trener (trenerzy, gdy jest ich więcej), trenujący obecnie lub w przeszłości drużynę**
- **kolejki rozgrywek (numer kolejki, nazwa kolejki, dzień / dni),**
- **mecze, (kto z kim, wynik meczu, data meczu),**
- **zdarzenia: w bazie należy umieścić informacje o zdarzeniach (np.: czy były i ilość czerwonych, ilość żółtych kartek, ilość rzutów karnych, „przerwany mecz - w której minucie?”, „mecz bez publiczności” ...itp.)**

Legia Warszawa – Górnik Zabrze

Jeden z meczy 4 kolejki (4 kolejka odbyła się w dniach 19-20.11.2015r) w którym padł wynik 2:1

W czasie tego meczu były zdarzenia:

- **2 żółte kartki**
- **1 czerwona kartka**
- **Rzut karny**
- **Mecz odbył się bez udziału publiczności**

Śląsk Wrocław – Polonia Warszawa

Jeden z meczy 4 kolejki (kolejka odbyła się w dniach 19-20.11.2010r) w którym padł wynik 0:3

W czasie tego meczu były zdarzenia:

- **1 żółta kartka**
- **2 Rzuty karne**
- **Widzów 22 000**

Powtórzenie materiału

Zaprojektować bazę danych rozgrywek piłkarskich. W bazie powinny znaleźć się informacje:

- **drużyny piłkarskie (nazwa, adres, miejscowość),**
- **trener (trenerzy, gdy jest ich więcej), trenujący obecnie lub w przeszłości drużynę**
- **kolejki rozgrywek (numer kolejki, nazwa kolejki, dzień / dni),**
- **mecze, (kto z kim, wynik meczu, data meczu),**
- **zdarzenia: w bazie należy umieścić informacje o zdarzeniach (np.: czy były i ilość czerwonych, ilość żółtych kartek, ilość rzutów karnych, „przerwany mecz - w której minucie?”, „mecz bez publiczności” ...itp.)**

Legia Warszawa – Górnik Zabrze

Jeden z meczy 4 kolejki (4 kolejka odbyła się w dniach 19-20.11.2015r) w którym padł wynik 2:1

W czasie tego meczu były zdarzenia:

- **2 żółte kartki**
- **1 czerwona kartka**
- **Rzut karny**
- **Mecz odbył się bez udziału publiczności**

Śląsk Wrocław – Polonia Warszawa

Jeden z meczy 4 kolejki (kolejka odbyła się w dniach 19-20.11.2010r) w którym padł wynik 0:3

W czasie tego meczu były zdarzenia:

- **1 żółta kartka**
- **2 Rzuty karne**
- **Widzów 22 000**

Śląsk Wrocław – Legia Warszawa

Jeden z meczy 5 kolejki (5 kolejka odbyła się w dniach 26-27.11.2010r) w którym padł wynik 1:2

W czasie tego meczu były zdarzenia:

- **1 czerwona kartka**
- **Widzów 12 000**

Powtórzenie materiału

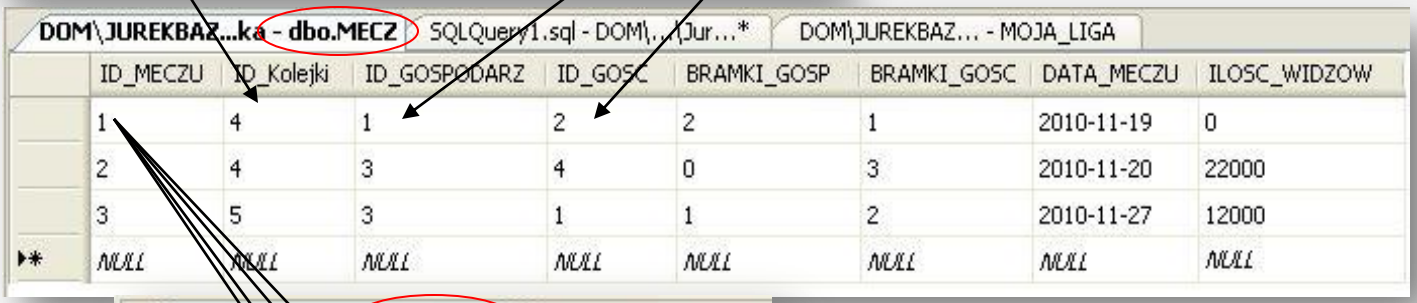
Dane do tabel

DOM\JUREKBAZA... dbo.KOLEJKI | SQLQuery1.sql - DOM\...\Jur...* | DOM\JU

ID_Kolejki	termin_kolejki	uwagi
1	10-12.10.2011	NULL
2	17-19.10.2011	Ciekawa
3	26-28.10-2011	NULL
4	19-20.10.2011	NULL
5	26-27.11.2011	nudna
NULL	NULL	NULL

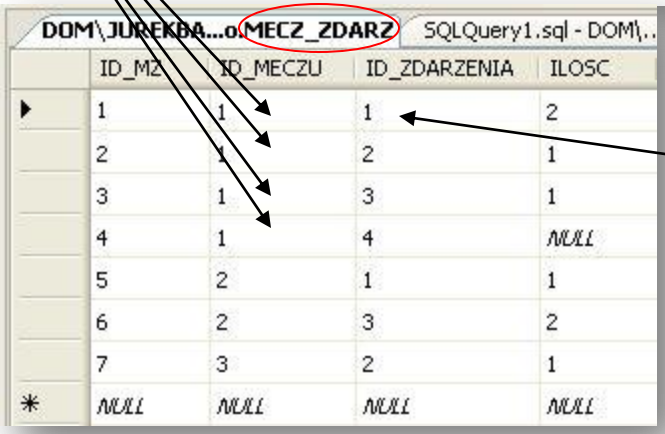
DOM\JUREKBAZ... dbo.DRUZYNA | SQLQuery1.sql - DOM\...\Jur...*

ID_Drużyny	Nazwa	Data_powstania	adres
1	LEGIA WARSZAWA	NULL	NULL
2	GÓRNIK ZABRZE	NULL	NULL
3	ŚLĄSK WROCŁAW	NULL	NULL
4	POLONIA WARSZAWA	NULL	NULL
NULL	NULL	NULL	NULL



DOM\JUREKBAZ...ka - dbo.MECZ | SQLQuery1.sql - DOM\...\Jur...* | DOM\JUREKBAZ... - MOJA_LIGA

ID_MECZU	ID_Kolejki	ID_GOSPODARZ	ID_GOSC	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	4	1	2	2	1	2010-11-19	0
2	4	3	4	0	3	2010-11-20	22000
3	5	3	1	1	2	2010-11-27	12000
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



DOM\JUREKBA...o.MECZ_ZDARZ | SQLQuery1.sql - DOM\...

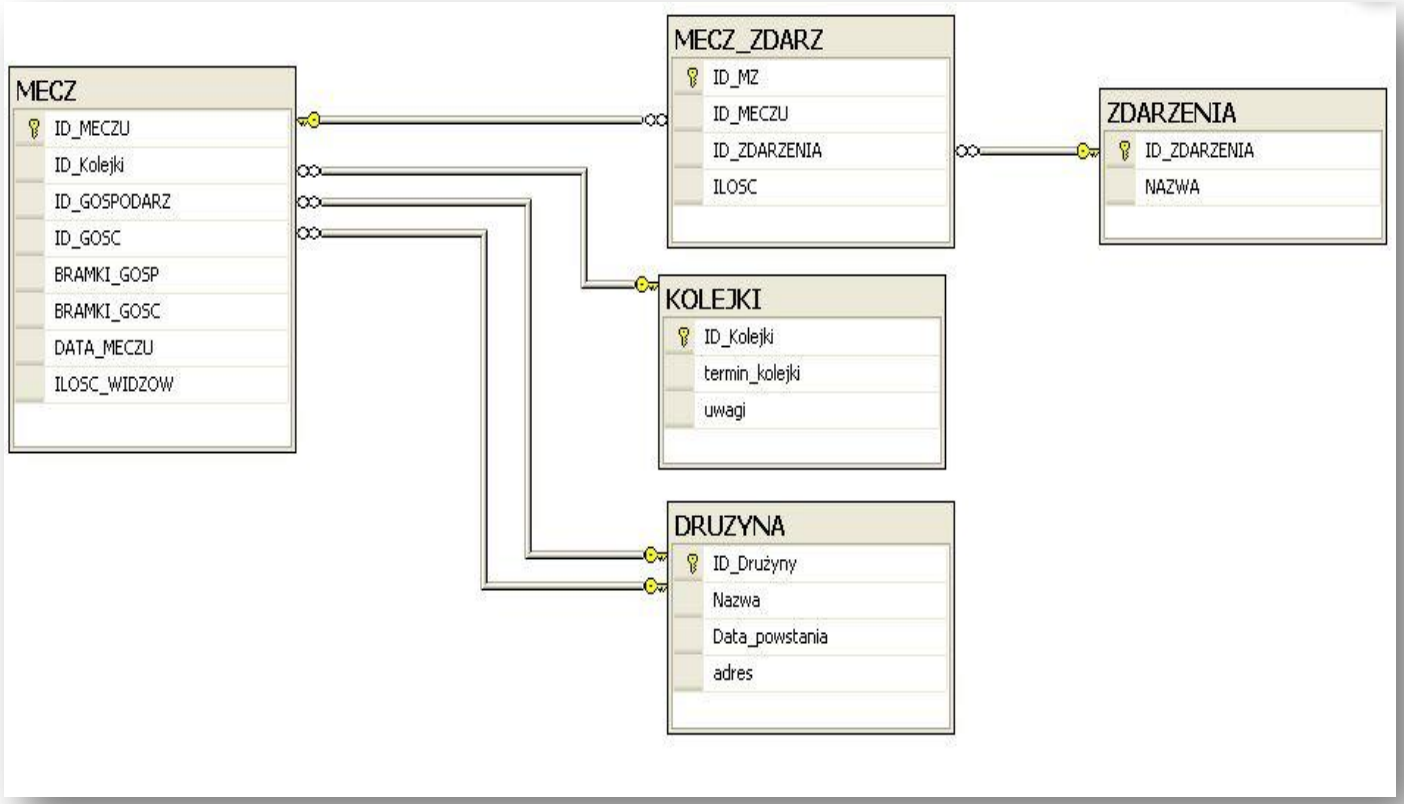
ID_MZ	ID_MECZU	ID_ZDARZENIA	ILOSC
1	1	1	2
2	1	2	1
3	1	3	1
4	1	4	NULL
5	2	1	1
6	2	3	2
7	3	2	1
NULL	NULL	NULL	NULL

DOM\JUREKBAZ...bo.ZDARZENIA | SQLQuery1.s

ID_ZDARZENIA	NAZWA
1	ŻÓŁTA KARTKA
2	CZERWONA KARTKA
3	RZUT KARNY
4	MECZ BEZ PUBLICZNOŚCI
NULL	NULL

Powtórzenie materiału

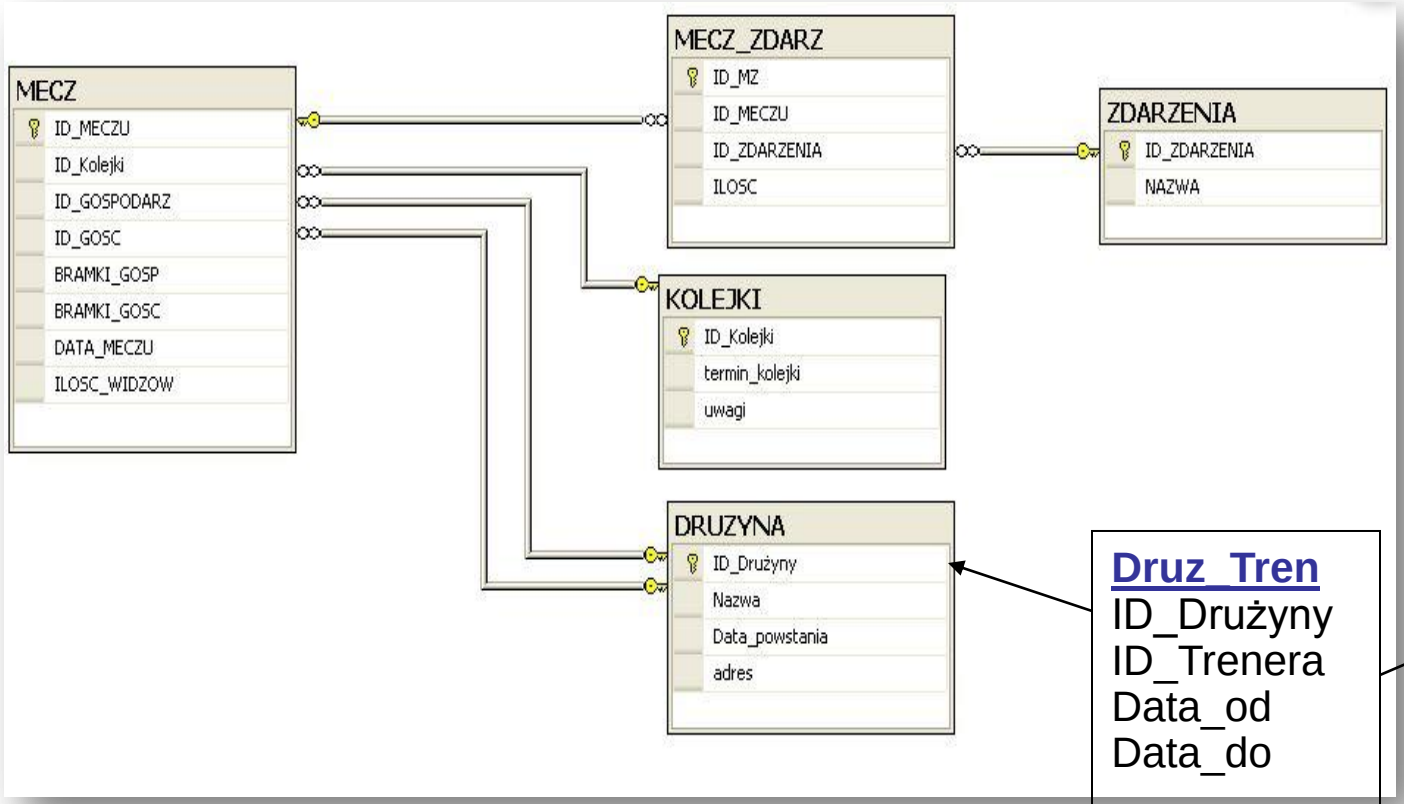
Diagram bazy danych



Czego nie zrobiłem? (sprawdzić założenia) ?

Powtórzenie materiału

Diagram bazy danych

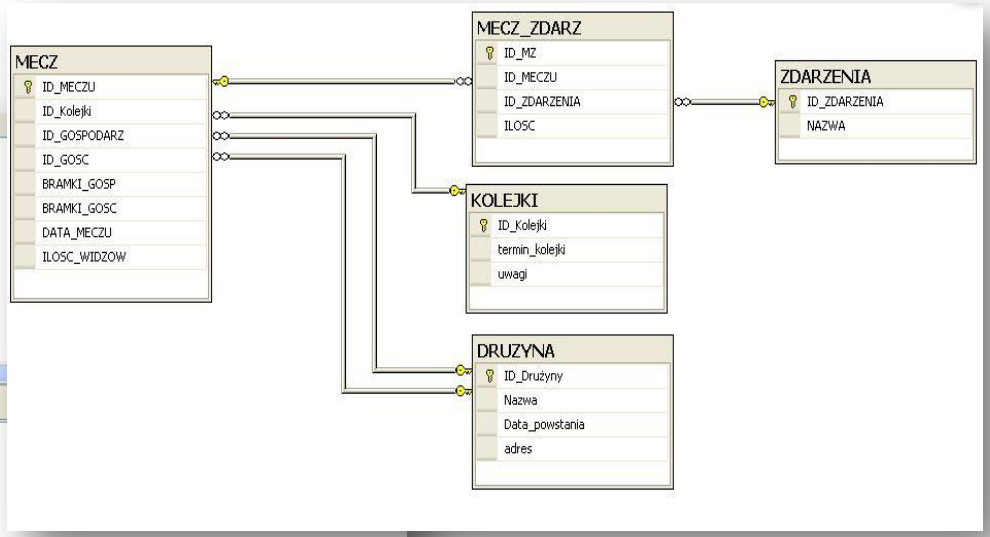


Druz_Tren
ID_Drużyny
ID_Trenera
Data_od
Data_do

Trenerzy
* ID_Trenera
Imię
Nazwisko
Data_ur

Powtórzenie materiału

ZAPYTANIE (1)



SQLQuery1.sql - DOM\...\Jur...* | DOM\JUREKBAZ... - MOJA_LIGA

```
SELECT * FROM dbo.KOLEJKI

SELECT * FROM dbo.DRUZYNA

SELECT * FROM dbo.ZDARZENIA

SELECT * FROM dbo.MECZ

SELECT * FROM dbo.MECZ_ZDARZ
```

Results | Messages

	ID_Kolejki	termin_kolejki	uwagi
1	1	10-12.10.2011	NULL
2	2	17-19.10.2011	Ciekawa
3	3	26-28.10.2011	NULL
4	4	19-20.10.2011	NULL
5	5	26-27.11.2011	nudna

	ID_Drużyny	Nazwa	Data_powstania	adres
1	1	LEGIA WARSZAWA	NULL	NULL
2	2	GÓRNIK ZABRZE	NULL	NULL
3	3	ŚLĄSK WROCŁAW	NULL	NULL
4	4	POLONIA WARSZAWA	NULL	NULL

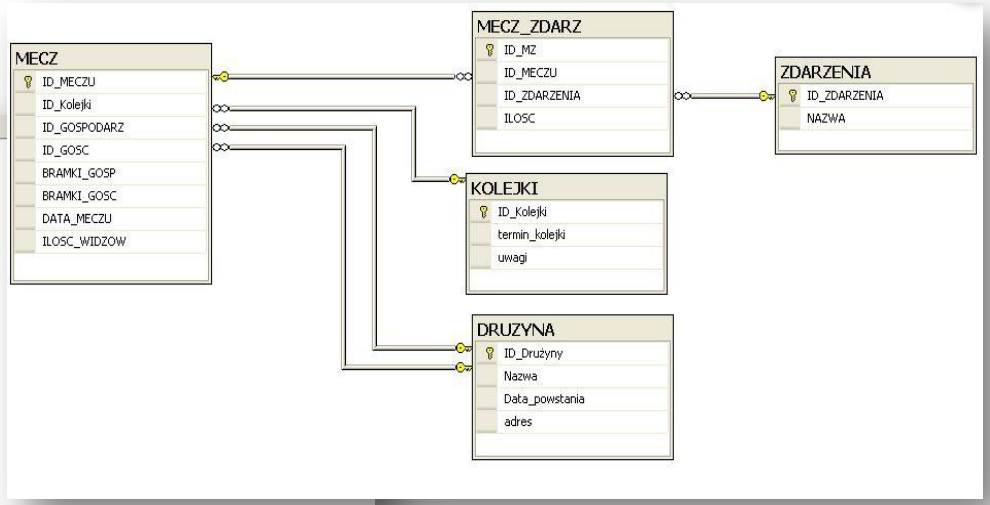
	ID_ZDARZENIA	NAZWA
1	1	ZÓŁTA KARTKA
2	2	CZERWONA KARTKA
3	3	RZUT KARNY
4	4	MECZ BEZ PUBLICZNOŚCI

	ID_MECZU	ID_Kolejki	ID_GOSPODARZ	ID_GOSC	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	1	4	1	2	2	1	2010-11-19	0
2	2	4	3	4	0	3	2010-11-20	22000
3	3	5	3	1	1	2	2010-11-27	12000

	ID_MZ	ID_MECZU	ID_ZDARZENIA	ILOSC
1	1	1	1	2
2	2	1	2	1
3	3	1	3	1
4	4	1	4	NULL
5	5	2	1	1
6	6	2	3	2
7	7	3	2	1

Powtórzenie materiału

ZAPYTANIE (2)



SQLQuery1.sql - DOM\...\Jur...* | DOM\JUREKBAZ... - MOJA_LIGA

```
SELECT * FROM KOLEJKI
    JOIN MECZ ON dbo.KOLEJKI.ID_Kolejki=dbo.MECZ.ID_Kolejki

SELECT  MECZ.ID_Kolejki,
        MECZ.ID_MECZU,
        a.Nazwa as Gospodarz,
        b.Nazwa as Gosc,
        BRAMKI_GOSP,
        BRAMKI_GOSC,
        DATA_MECZU,
        ILOSC_WIDZOW
    FROM MECZ
        JOIN DRUZYNA AS a ON MECZ.ID_Gospodarz = a.ID_Drużyny
        JOIN DRUZYNA AS b ON MECZ.ID_Gosc = b.ID_Drużyny

SELECT MECZ.ID_MECZU,NAZWA,ILOSC FROM MECZ
    JOIN MECZ_ZDARZ ON MECZ.ID_MECZU=MECZ_ZDARZ.ID_MECZU
    JOIN ZDARZENIA ON MECZ_ZDARZ.ID_ZDARZENIA=ZDARZENIA.ID_ZDARZENIA
```

Results | Messages

	ID_Kolejki	termin_kolejki	uwagi	ID_MECZU	ID_Kolejki	ID_GOSPODARZ	ID_GOSC	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	4	19-20.10.2011	NULL	1	4	1	2	2	1	2010-11-19	0
2	4	19-20.10.2011	NULL	2	4	3	4	0	3	2010-11-20	22000
3	5	26-27.11.2011	nudna	3	5	3	1	1	2	2010-11-27	12000

	ID_Kolejki	ID_MECZU	Gospodarz	Gosc	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	4	1	LEGIA WARSZAWA	GÓRNIK ZABRZE	2	1	2010-11-19	0
2	4	2	ŚLĄSK WROCŁAW	POLONIA WARSZAWA	0	3	2010-11-20	22000
3	5	3	ŚLĄSK WROCŁAW	LEGIA WARSZAWA	1	2	2010-11-27	12000

	ID_MECZU	NAZWA	ILOSC
1	1	ŻÓŁTA KARTKA	2
2	1	CZERWONA KARTKA	1
3	1	RZUT KARNY	1
4	1	MECZ BEZ PUBLICZNOŚCI	NULL
5	2	ŻÓŁTA KARTKA	1
6	2	RZUT KARNY	2
7	3	CZERWONA KARTKA	1

Legia Warszawa – Górnik Zabrze

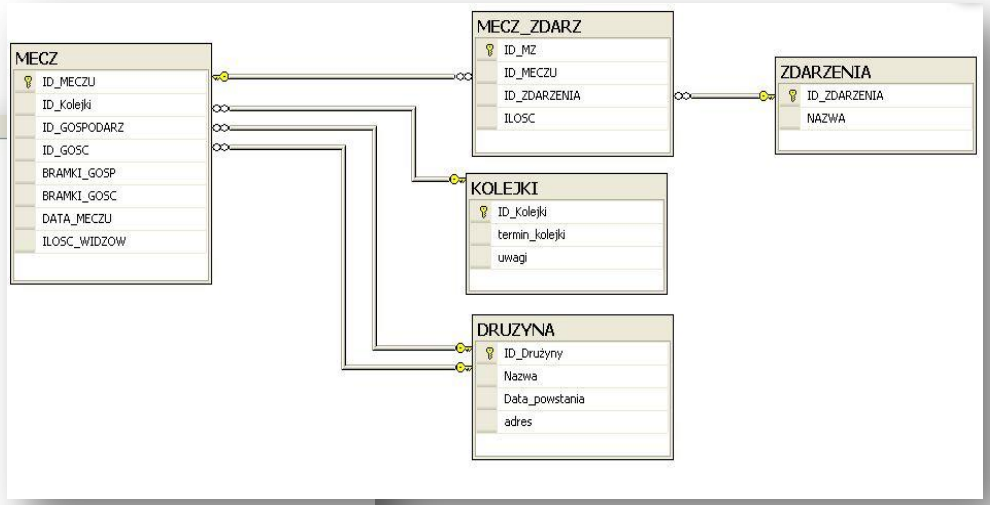
Jeden z meczy 4 kolejki (4 kolejka odbyła się w dniach 19-20.11.2010r) w którym padł wynik 2:1

W czasie tego meczu były zdarzenia:

- 2 żółte kartki
- 1 czerwona kartka
- Rzut karny
- Mecz odbył się bez udziału publiczności

Powtórzenie materiału

ZAPYTANIE (2)



SQLQuery1.sql - DOM\...\Jur...* | DOM\JUREKBAZ... - MOJA_LIGA

```
SELECT * FROM KOLEJKI
    JOIN MECZ ON dbo.KOLEJKI.ID_Kolejki=dbo.MECZ.ID_Kolejki

SELECT  MECZ.ID_Kolejki,
        MECZ.ID_MECZU,
        a.Nazwa as Gospodarz,
        b.Nazwa as Gosc,
        BRAMKI_GOSP,
        BRAMKI_GOSC,
        DATA_MECZU,
        ILOSC_WIDZOW
    FROM MECZ
        JOIN DRUZYNA AS a ON MECZ.ID_Gospodarz = a.ID_Drużyny
        JOIN DRUZYNA AS b ON MECZ.ID_Gosc = b.ID_Drużyny

SELECT MECZ.ID_MECZU,NAZWA,ILOSC FROM MECZ
    JOIN MECZ_ZDARZ ON MECZ.ID_MECZU=MECZ_ZDARZ.ID_MECZU
    JOIN ZDARZENIA ON MECZ_ZDARZ.ID_ZDARZENIA=ZDARZENIA.ID_ZDARZENIA
```

Results | Messages

	ID_Kolejki	termin_kolejki	uwagi	ID_MECZU	ID_Kolejki	ID_GOSPODARZ	ID_GOSC	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	4	19-20.10.2011	NULL	1	4	1	2	2	1	2010-11-19	0
2	4	19-20.10.2011	NULL	2	4	3	4	0	3	2010-11-20	22000
3	5	26-27.11.2011	nudna	3	5	3	1	1	2	2010-11-27	12000

	ID_Kolejki	ID_MECZU	Gospodarz	Gosc	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	4	1	LEGIA WARSZAWA	GÓRNIK ZABRZE	2	1	2010-11-19	0
2	4	2	ŚLĄSK WROCŁAW	POLONIA WARSZAWA	0	3	2010-11-20	22000
3	5	3	ŚLĄSK WROCŁAW	LEGIA WARSZAWA	1	2	2010-11-27	12000

	ID_MECZU	NAZWA	ILOSC
1	1	ŻÓŁTA KARTKA	2
2	1	CZERWONA KARTKA	1
3	1	RZUT KARNY	1
4	1	MECZ BEZ PUBLICZNOŚCI	NULL
5	2	ŻÓŁTA KARTKA	1
6	2	RZUT KARNY	2
7	3	CZERWONA KARTKA	1

Śląsk Wrocław – Polonia Warszawa

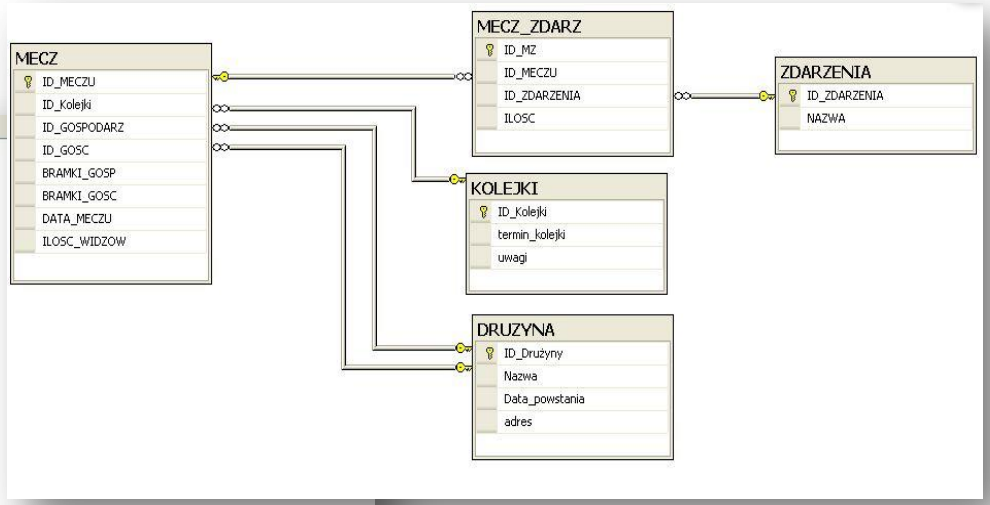
Jeden z meczy 4 kolejki (kolejka odbyła się w dniach 19-20.11.2010r) w którym padł wynik 0:3

W czasie tego meczu były zdarzenia:

- 1 żółta kartka
- 2 Rzuty karne
- Widzów 22 000

Powtórzenie materiału

ZAPYTANIE (2)



SQLQuery1.sql - DOM\...\Jur...* | DOM\JUREKBAZ... - MOJA_LIGA

```
SELECT * FROM KOLEJKI
    JOIN MECZ ON dbo.KOLEJKI.ID_Kolejki=dbo.MECZ.ID_Kolejki

SELECT  MECZ.ID_Kolejki,
        MECZ.ID_MECZU,
        a.Nazwa as Gospodarz,
        b.Nazwa as Gosc,
        BRAMKI_GOSP,
        BRAMKI_GOSC,
        DATA_MECZU,
        ILOSC_WIDZOW
    FROM MECZ
        JOIN DRUZYNA AS a ON MECZ.ID_Gospodarz = a.ID_Drużyny
        JOIN DRUZYNA AS b ON MECZ.ID_Gosc = b.ID_Drużyny

SELECT MECZ.ID_MECZU,NAZWA,ILOSC FROM MECZ
    JOIN MECZ_ZDARZ ON MECZ.ID_MECZU=MECZ_ZDARZ.ID_MECZU
    JOIN ZDARZENIA ON MECZ_ZDARZ.ID_ZDARZENIA=ZDARZENIA.ID_ZDARZENIA
```

Results | Messages

	ID_Kolejki	termin_kolejki	uwagi	ID_MECZU	ID_Kolejki	ID_GOSPODARZ	ID_GOSC	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	4	19-20.10.2011	NULL	1	4	1	2	2	1	2010-11-19	0
2	4	19-20.10.2011	NULL	2	4	3	4	0	3	2010-11-20	22000
3	5	26-27.11.2011	nudna	3	5	3	1	1	2	2010-11-27	12000

	ID_Kolejki	ID_MECZU	Gospodarz	Gosc	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	4	1	LEGIA WARSZAWA	GÓRNIK ZABRZE	2	1	2010-11-19	0
2	4	2	ŚLĄSK WROCŁAW	POLONIA WARSZAWA	0	3	2010-11-20	22000
3	5	3	ŚLĄSK WROCŁAW	LEGIA WARSZAWA	1	2	2010-11-27	12000

	ID_MECZU	NAZWA	ILOSC
1	1	ZÓŁTA KARTKA	2
2	1	CZERWONA KARTKA	1
3	1	RZUT KARNY	1
4	1	MECZ BEZ PUBLICZNOŚCI	NULL
5	2	ZÓŁTA KARTKA	1
6	2	RZUT KARNY	2
7	3	CZERWONA KARTKA	1

Śląsk Wrocław – Legia Warszawa

Jeden z meczy 5 kolejki (5 kolejka odbyła się w dniach 26-27.11.2010r) w którym padł wynik 1:2

W czasie tego meczu były zdarzenia:

1 czerwona kartka

Widzów 12 000

Powtórzenie materiału

A jak uzyskać „TABELĘ WYNIKÓW” wszystkich drużyn na podstawie danych tabeli MECZ?

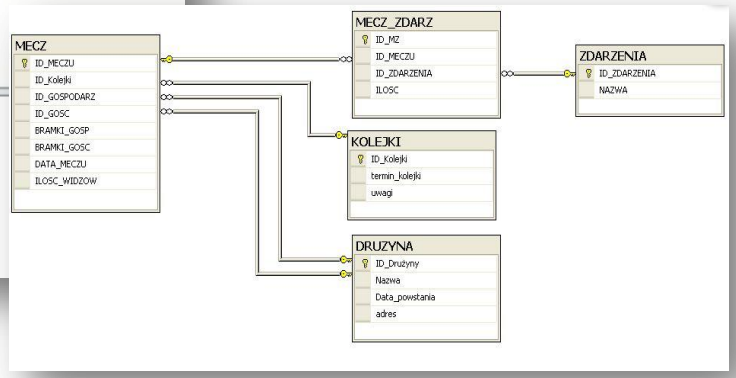
Założenia:

- Drużyna (jak widać w tabeli MECZ) gra mecze „u siebie” (gospodarz) i „na wyjeździe” (gość),
- Drużyna za zwycięstwo ma 3 pkt., za remis ma 1 pkt., za porażkę ma 0 pkt.,
- Tabela wyników Ligi piłkarskiej zawiera m.in.. pola:
  - ✓ Nazwa drużyny,
  - ✓ Łączna ilość rozegranych meczy,
  - ✓ Łączna ilość zdobytych punktów,
  - ✓ Łączna ilość strzelonych bramek,
  - ✓ Łączna ilość straconych bramek.

Results | Messages

	ID_MECZU	ID_Kolejki	ID_GOSPODARZ	ID_GOSC	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	1	4	1	2	2	1	2011-11-19	0
2	2	4	3	4	0	3	2011-11-20	22000
3	3	5	3	1	1	2	2011-11-27	12000

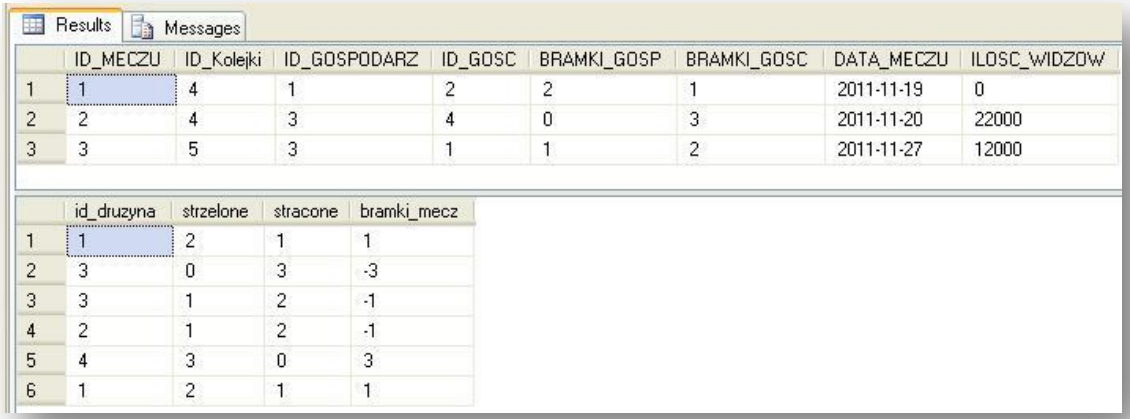
	nazwa	id_druzyna	ilosc_meczy	liczba_punktów	strzelone	stracone
1	LEGIA WARSZAWA	1	2	6	4	2
2	POLONIA WARSZAWA	4	1	3	3	0
3	GÓRNIK ZABRZE	2	1	0	1	2
4	ŚLĄSK WROCŁAW	3	2	0	1	5



Powtórzenie materiału

Może zastosować CTE? (1)

- CTE, czyli wspólne wyrażenia tablicowe, zostały wprowadzone po raz pierwszy w SQL Server 2005, jako rozszerzenie składni T-SQL
- Upraszczają i poprawiają przejrzystość kodu SQL
- W tym zakresie, ich stosowanie nie ma wpływu na wydajność zapytań, tylko na jego czytelność
- Oprócz funkcji czysto estetycznej, posiadają jeszcze jedną, specjalną właściwość – ich struktura pozwala na realizację rekurencji
- CTE, możemy stosować praktycznie wszędzie: w widokach, funkcjach, procedurach składowanych
- Definicję CTE otwiera słowo kluczowe WITH z nazwą zbioru, który zostanie za jej pomocą utworzony. Do niej będziemy odwoływać się w kwerendzie, tak jak do zwykłej tabeli.
- Obowiązują te same reguły jak w przypadku każdego obiektu tabelarycznego, do którego chcemy się w jakikolwiek sposób odnosić w zapytaniu.


Results Messages

	ID_MECZU	ID_Kolejki	ID_GOSPODARZ	ID_GOSC	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	1	4	1	2	2	1	2011-11-19	0
2	2	4	3	4	0	3	2011-11-20	22000
3	3	5	3	1	1	2	2011-11-27	12000

	id_druzyna	strzelone	stracone	bramki_mecz
1	1	2	1	1
2	3	0	3	-3
3	3	1	2	-1
4	2	1	2	-1
5	4	3	0	3
6	1	2	1	1

```
WITH mecz_wygrane_CTE
AS
(   -- określenie unikalnych nazw kolumn
    SELECT id_gospodarz as id_druzyna,
           bramki_gosp as strzelone,
           bramki_gosc as stracone,
           bramki_gosp-bramki_gosc as
       bramki_mecz
    FROM mecz
    union all
    SELECT id_gosc,
           bramki_gosc,
           bramki_gosp,
           bramki_gosc-bramki_gosp
    FROM mecz
)
SELECT * from mecz_wygrane_CTE
```

Powtórzenie materiału

Może zastosować CTE? (2)

Results | Messages

	ID_MECZU	ID_Kolejki	ID_GOSPODARZ	ID_GOSC	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	1	4	1	2	2	1	2011-11-19	0
2	2	4	3	4	0	3	2011-11-20	22000
3	3	5	3	1	1	2	2011-11-27	12000

	id_druzyna	strzelone	stracone	bramki_mecz
1	1	2	1	1
2	3	0	3	-3
3	3	1	2	-1
4	2	1	2	-1
5	4	3	0	3
6	1	2	1	1

Results | Messages

	ID_MECZU	ID_Kolejki	ID_GOSPODARZ	ID_GOSC	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	1	4	1	2	2	1	2011-11-19	0
2	2	4	3	4	0	3	2011-11-20	22000
3	3	5	3	1	1	2	2011-11-27	12000

	id_druzyna	strzelone	stracone	bramki_mecz	punkty
1	1	2	1	1	3
2	3	0	3	-3	0
3	3	1	2	-1	0
4	2	1	2	-1	0
5	4	3	0	3	3
6	1	2	1	1	3

```
WITH mecz_wygrane_CTE
AS
(   -- określenie unikalnych nazw kolumn
    SELECT id_gospodarz as id_druzyna,
           bramki_gosp as strzelone,
           bramki_gosc as stracone,
           bramki_gosp-bramki_gosc as
      bramki_mecz
    FROM mecz
    union all
    SELECT id_gosc,
           bramki_gosc,
           bramki_gosp,
           bramki_gosc-bramki_gosp
    FROM mecz
),
punkty_cte
as
(
SELECT *,
case
when bramki_mecz>0 then 3
when bramki_mecz=0 then 1
else 0
end as punkty
from mecz_wygrane_CTE
)
SELECT * from punkty_CTE
```

Powtórzenie materiału

Może zastosować CTE? (3)

Results | Messages

	ID_MECZU	ID_Kolejki	ID_GOSPODARZ	ID_GOSC	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	1	4	1	2	2	1	2011-11-19	0
2	2	4	3	4	0	3	2011-11-20	22000
3	3	5	3	1	1	2	2011-11-27	12000

	id_druzyna	strzelone	stracone	bramki_mecz	punkty
1	1	2	1	1	3
2	3	0	3	-3	0
3	3	1	2	-1	0
4	2	1	2	-1	0
5	4	3	0	3	3
6	1	2	1	1	3

	id_druzyna	ilosc_meczy	liczba_punktów	strzelone	stracone
1	1	2	6	4	2
2	4	1	3	3	0
3	2	1	0	1	2
4	3	2	0	1	5

```
WITH mecz_wygrane_CTE
AS
(   -- określenie unikalnych nazw kolumn
    SELECT id_gospodarz as id_druzyna,
           bramki_gosp as strzelone,
           bramki_gosc as stracone,
           bramki_gosp-bramki_gosc as bramki_mecz
    FROM mecz
    union all
    SELECT id_gosc,
           bramki_gosc,
           bramki_gosp,
           bramki_gosc-bramki_gosp
    FROM mecz
),
punkty_CTE
as
(
SELECT *,
case
when bramki_mecz>0 then 3
when bramki_mecz=0 then 1
else 0
end as punkty
from mecz_wygrane_CTE
)
SELECT id_druzyna,
       count(*) as ilosc_meczy,
       sum(punkty) as liczba_punktów,
       sum(strzelone) as strzelone,
       sum(stracone) as stracone
from punkty_CTE
group by id_druzyna
order by liczba_punktów desc
```

Powtórzenie materiału

Może zastosować CTE? (4)

Results | Messages

	ID_MECZU	ID_Kolejki	ID_GOSPODARZ	ID_GOSC	BRAMKI_GOSP	BRAMKI_GOSC	DATA_MECZU	ILOSC_WIDZOW
1	1	4	1	2	2	1	2011-11-19	0
2	2	4	3	4	0	3	2011-11-20	22000
3	3	5	3	1	1	2	2011-11-27	12000

	id_druzyna	strzelone	stracone	bramki_mecz	punkty
1	1	2	1	1	3
2	3	0	3	-3	0
3	3	1	2	-1	0
4	2	1	2	-1	0
5	4	3	0	3	3
6	1	2	1	1	3

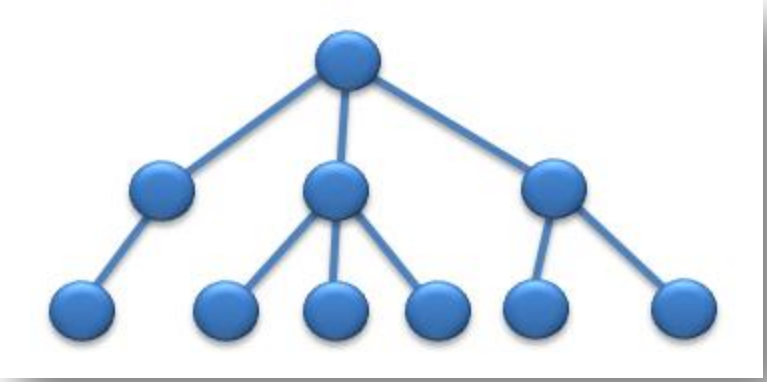
	id_druzyna	ilosc_meczy	liczba_punktów	strzelone	stracone
1	1	2	6	4	2
2	4	1	3	3	0
3	2	1	0	1	2
4	3	2	0	1	5

	nazwa	id_druzyna	ilosc_meczy	liczba_punktów	strzelone	stracone
1	LEGIA WARSZAWA	1	2	6	4	2
2	POLONIA WARSZAWA	4	1	3	3	0
3	GÓRNIK ZABRZE	2	1	0	1	2
4	ŚLĄSK WROCŁAW	3	2	0	1	5

```
WITH mecz_wygrane_CTE
AS
(   -- określenie unikalnych nazw kolumn
    SELECT id_gospodarz as id_druzyna,
           bramki_gosp as strzelone,
           bramki_gosc as stracone,
           bramki_gosp-bramki_gosc as bramki_mecz
    FROM mecz
    union all
    SELECT id_gosc,
           bramki_gosc,
           bramki_gosp,
           bramki_gosc-bramki_gosp
    FROM mecz
),
punkty_CTE
as
(
SELECT *,
case
when bramki_mecz>0 then 3
when bramki_mecz=0 then 1
else 0
end as punkty
from mecz_wygrane_CTE
)
SELECT d.nazwa,
       p.id_druzyna,
       count(*) as ilosc_meczy,
       sum(punkty) as liczba_punktów,
       sum(strzelone) as strzelone,
       sum(stracone) as stracone
from punkty_CTE p join druzyna d on
       p.id_druzyna=d.id_druzyny
group by d.nazwa,p.id_druzyna
order by liczba_punktów desc
```

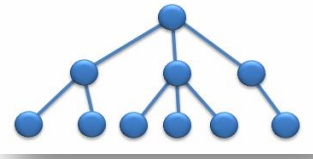
Grafy Proste (acykliczne, nieskierowane) (1)

Graf prosty – to drzewo bez cykli, krawędzie nie są skierowane (nie ma ograniczeń kierunkowych) a pomiędzy każdą parą dowolnych wierzchołków, jest tylko jedna ścieżka. Dodanie jakiejkolwiek nowej krawędzi, spowoduje powstanie cyklu a usunięcie istniejącej doprowadzi do niespójności (podział grafu).

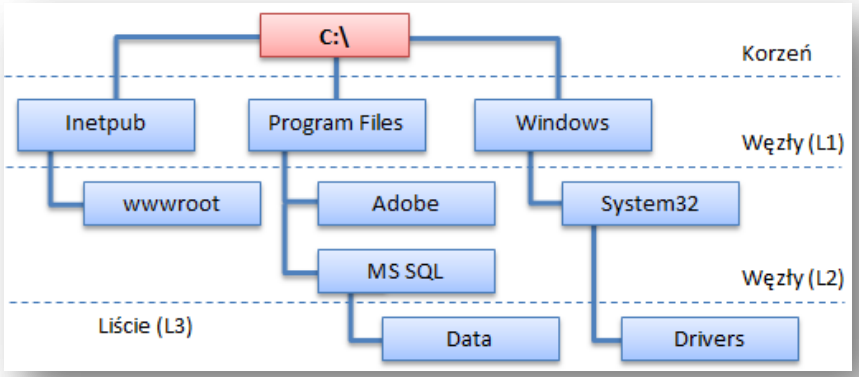


- W przypadku, gdy jeden z wierzchołków, jest wyszczególniony (korzeń), mówimy o drzewie ukorzenionym.
- Grafy proste, ukorzenione są łatwe w implementacji w relacyjnych bazach.
- Tworzenie zapytań do takich struktur nie stwarza większych problemów. Tego typu graf, to na przykład organizacja systemu plikowego lub hierarchia pracownicza – przełożony/podwładny.

Grafy Proste (acykliczne, nieskierowane) (2)

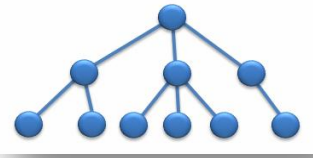


- W typowej organizacji plików i katalogów zakładamy, że dany obiekt (plik lub katalog) nie może być w dwóch różnych miejscach jednocześnie.
- Do każdego węzła, jest tylko jedna ścieżka, prowadząca ,od korzenia grafu.
- Patrząc na tą strukturę, całość stanowi jedno spójne drzewo – graf prosty, ukorzeniony.

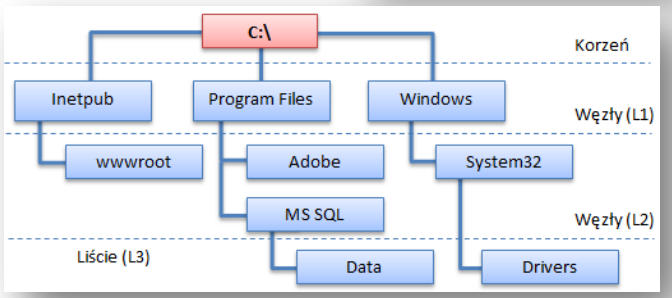


- Podobnie w strukturze zatrudnienia. Dany pracownik zazwyczaj nie może mieć na tym samym poziomie dwóch bezpośrednich szefów. Droga eskalacji od podwładnego do przełożonego najwyższego szczebla, jest jedna.
- Implementacja takiej struktury w bazie danych może być bardzo prosta. Realizuje się ją zazwyczaj, jako SELF JOIN, czyli złączenie tabeli samej ze sobą.

Grafy Proste (acykliczne, nieskierowane) (3)



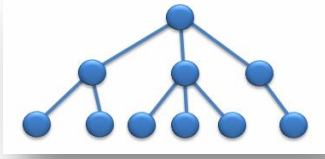
```
Create Database GRAFY
GO

USE GRAFY
GO

CREATE TABLE dbo.Folders
(
 ID     INT          NOT NULL PRIMARY KEY IDENTITY(1,1)
 Folder VARCHAR(100) NOT NULL,
 Parent INT          NULL REFERENCES dbo.Folders( ID )
);

INSERT INTO dbo.Folders ( Folder ) VALUES ('C:');

INSERT INTO dbo.Folders ( Folder, Parent )
VALUES  ('Inetpub',1),
('Program Files',1),
('Windows',1),
('wwwroot',2),
('Adobe',3),
('MS SQL',3),
('Data',7),
('system32',4),
('Data',9);

select * from dbo.Folders
```



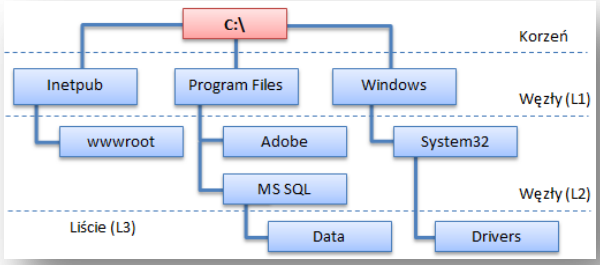
Results | Messages

	ID	Folder	Parent
1	1	C:	NULL
2	2	Inetpub	1
3	3	Program Files	1
4	4	Windows	1
5	5	wwwroot	2
6	6	Adobe	3
7	7	MS SQL	3
8	8	Data	7
9	9	system32	4
10	10	Data	9

Grafy Proste (acykliczne, nieskierowane) (4)



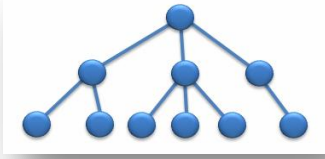
- Zapytania SQL do takiego grafu, również nie są szczególnie skomplikowane.
- Wystarczy wykorzystać rekurencję wspólnych wyrażeń tablicowych (CTE Common Table Expressions) i przeszukiwać drzewo w głąb od korzenia do liści.




```
WITH Paths AS
(
    SELECT Id,
           Folder,
           CAST(null AS VARCHAR(8000)) AS Parent,
           CAST(Folder + '\' AS VARCHAR(8000)) FullFolderName,
           0 as Poziom
    FROM dbo.Folders
    WHERE Parent IS NULL

    UNION ALL

    SELECT f.Id,
           f.Folder,
           p.FullFolderName,
           CAST(ISNULL(p.Parent,'')+ p.Folder + '\' + f.Folder + '\' AS VARCHAR(8000)),
           p.Poziom +1
    FROM dbo.Folders f INNER JOIN Paths p
                                ON f.Parent= p.ID
)
SELECT * FROM Paths
Order by FullFolderName
```

Powtórzenie materiału

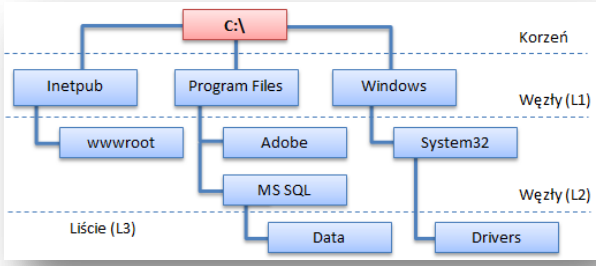
Grafy Proste (acykliczne, nieskierowane) (5)



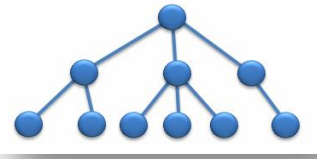
```
WITH Paths AS
(
    SELECT Id, Folder, CAST(null AS VARCHAR(8000)) AS Parent,
           CAST(Folder + '\' AS VARCHAR(8000)) FullFolderName, 0 as Poziom
    FROM dbo.Folders
    WHERE Parent IS NULL

    UNION ALL

    SELECT f.Id, f.Folder, p.FullFolderName,
           CAST(ISNULL(p.Parent, '')  + p.Folder + '\'
        +  f.Folder + '\'  AS VARCHAR(8000)) , p.Poziom +1
    FROM dbo.Folders f INNER JOIN Paths p
                               ON f.Parent= p.ID
)
SELECT * FROM Paths
Order by FullFolderName
```

Results | Messages

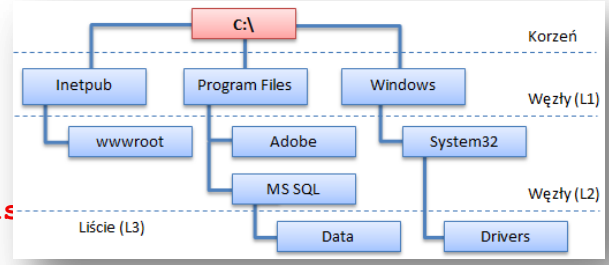
	Id	Folder	Parent	FullFolderName	Poziom
1	1	C:	NULL	C:\	0
2	2	Inetpub	C:\	C:\Inetpub\	1
3	5	wwwroot	C:\Inetpub\	C:\Inetpub\wwwroot\	2
4	3	Program Files	C:\	C:\Program Files\	1
5	6	Adobe	C:\Program Files\	C:\Program Files\Adobe\	2
6	7	MS SQL	C:\Program Files\	C:\Program Files\MS SQL\	2
7	8	Data	C:\Program Files\MS SQL\	C:\Program Files\MS SQL\Data\	3
8	4	Windows	C:\	C:\Windows\	1
9	9	system32	C:\Windows\	C:\Windows\system32\	2
10	10	Data	C:\Windows\system32\	C:\Windows\system32\Data\	3



Grafy Proste (acykliczne, nieskierowane) (6)



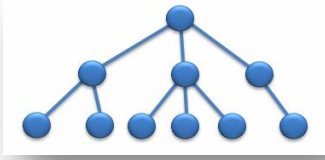
- Takie podejście, załatwia również sprawę, w sytuacji, gdy mamy do czynienia ze zbiorem grafów prostych (las drzew).
- Zakotwiczenie CTE, identyfikuje wszystkie grafy i rekurencyjnie rozpina ja analogicznie jak poprzednio. Drzewa możemy ponumerować, do późniejszej analizy.



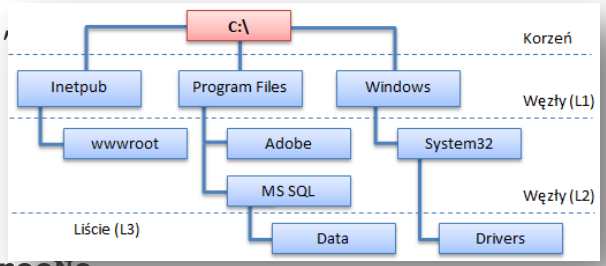

```
-- dodajmy nowe drzewo - dysk D:
Insert into dbo.Folders(Folder) Values('D:');

Insert into dbo.Folders ( [Folder], [Parent] )
VALUES('Music',11),('Docs',11),('Mike Oldfield',12),('Tres Lunas

WITH Paths AS
(
   SELECT Id,
          Folder,
          CAST(null AS VARCHAR(8000)) AS Parent,
          CAST(Folder + '\' AS VARCHAR(8000)) FullFolderName,
          0 as Poziom,
          ROW_NUMBER() OVER(ORDER BY FOLDER) as TreeNo
    FROM dbo.Folders
          WHERE Parent IS NULL
    UNION ALL

   SELECT f.Id,
          f.Folder,
          p.FullFolderName,
          CAST(ISNULL(p.Parent, '')  + p.Folder + '\' +  f.Folder + '\'  AS VARCHAR(8000)) ,
   p.Poziom +1, TreeNo
    FROM dbo.Folders f INNER JOIN Paths p
                ON f.Parent= p.ID
)
SELECT * FROM Paths
Order by FullFolderName
```

Powtórzenie materiału

Grafy Proste (acykliczne, nieskierowane) (7)



```
-- dodajmy nowe drzewo - dysk D:
Insert into dbo.Folders(Folder) Values('D:');

Insert into dbo.Folders ( [Folder], [Parent] )
VALUES('Music',11),('Docs',11),('Mike Oldfield',12),('Tres Lunas',14);

WITH Paths AS
(
    SELECT Id, Folder, CAST(null AS VARCHAR(8000)) AS Parent,
CAST(Folder + '\' AS VARCHAR(8000)) FullFolderName, 0 as Poziom,
ROW_NUMBER() OVER(ORDER BY FOLDER) as TreeNo
    FROM dbo.Folders
    WHERE Parent IS NULL

    UNION ALL

    SELECT f.Id, f.Folder, p.FullFolderName,
          CAST(ISNULL(p.Parent, '')  + p.Folder + '\'
        +  f.Folder + '\'  AS VARCHAR(8000)) , p.Poziom +1, TreeNo
    FROM dbo.Folders f INNER JOIN Paths p
                              ON f.Parent= p.ID
)
SELECT * FROM Paths
Order by FullFolderName
```



Results | Messages

	Id	Folder	Parent	FullFolderName	Poziom	TreeNo
1	1	C:	NULL	C:\	0	1
2	2	Inetpub	C:\	C:\Inetpub\	1	1
3	5	wwwroot	C:\Inetpub\	C:\Inetpub\wwwroot\	2	1
4	3	Program Files	C:\	C:\Program Files\	1	1
5	6	Adobe	C:\Program Files\	C:\Program Files\Adobe\	2	1
6	7	MS SQL	C:\Program Files\	C:\Program Files\MS SQL\	2	1
7	8	Data	C:\Program Files\MS SQL\	C:\Program Files\MS SQL\Data\	3	1
8	4	Windows	C:\	C:\Windows\	1	1
9	9	system32	C:\Windows\	C:\Windows\system32\	2	1
10	10	Data	C:\Windows\system32\	C:\Windows\system32\Data\	3	1
11	11	D:	NULL	D:\	0	2
12	13	Docs	D:\	D:\Docs\	1	2
13	12	Music	D:\	D:\Music\	1	2
14	14	Mike Oldfield	D:\Music\	D:\Music\Mike Oldfield\	2	2
15	15	Tres Lunas	D:\Music\Mike Oldfield\	D:\Music\Mike Oldfield\Tres Lunas\	3	2