



Wprowadzenie do projektowania i wykorzystania baz danych

Relacje i elementy projektowania baz

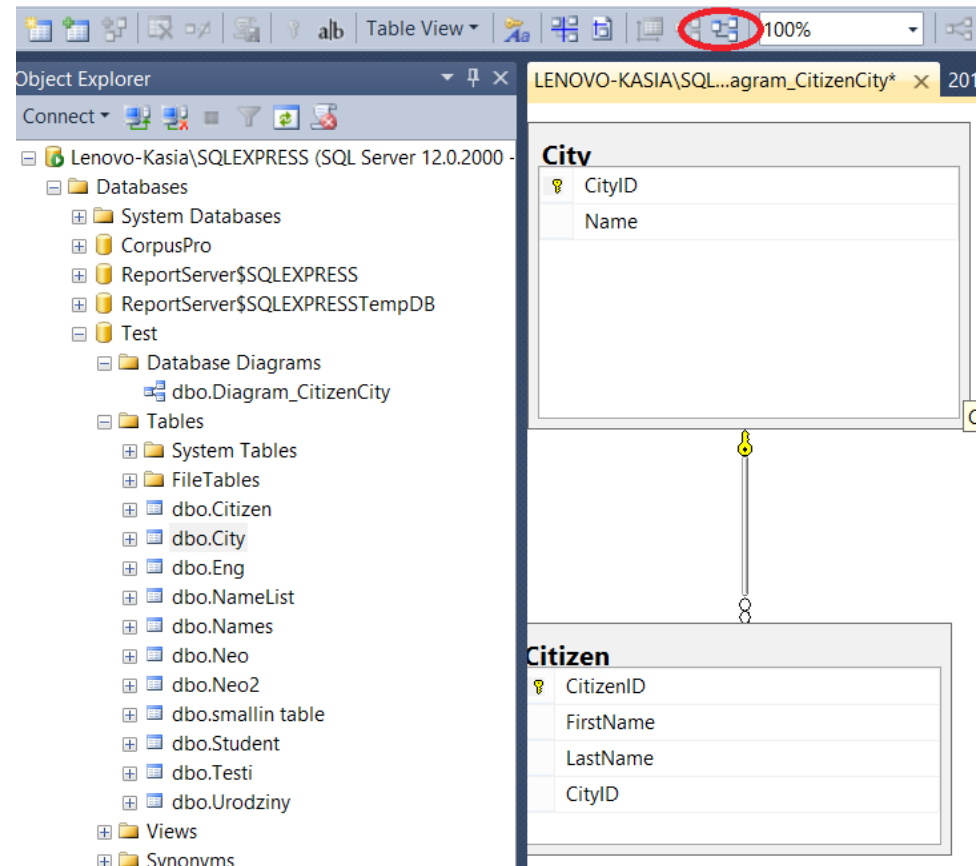
Katarzyna Klessa



RELACJE – dwa sposoby tworzenia

Tworzenie relacji:

- “ręcznie” za pomocą odpowiednich zapytań (ALTER TABLE)
- W edytorze, za pomocą narzędzi graficznych





RELACJE - typy

- Relacja jeden do wielu
- Relacje wiele-do-wielu
- Relacje jeden do jednego

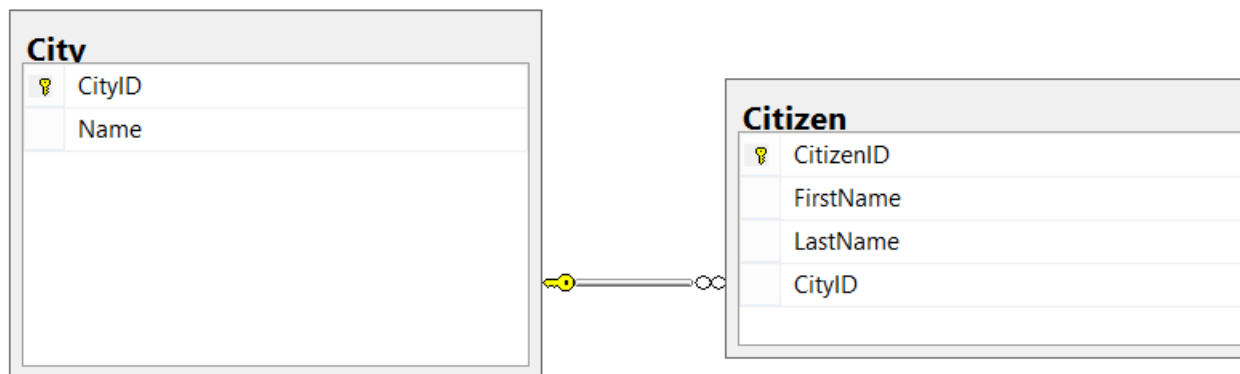


RELACJA jeden do wielu

- Relacja jeden do wielu to najpopularniejszy typ relacji.
- Zakładając relację (wszelkich typów) należy pamiętać, że relacje możemy zakładać na kolumny, które mają taki sam typ danych, w przeciwnym razie SSMS zgłosi błąd
- W edytorze graficznym relację wiele do wielu zakładamy ciągnąc myszką od kolumny będącej kluczem głównym w jednej tabeli do odpowiadającej jej typem kolumny w drugiej tabeli
- **Ważne jest zatem właściwe zaprojektowanie tabel bazy danych i zdefiniowanie typów danych przed nakładaniem relacji.**



RELACJA jeden do wielu





RELACJA wiele do wielu, przykład projektowania tabeli użytej w relacji, ustawienie klucza gł.

The screenshot shows the Microsoft SQL Server Enterprise Designer interface. The Object Explorer on the left shows the database structure. The main window displays the table design for 'dbo.Citizen' with columns: TeacherID (int), Name (nvarchar(50)), Surname (nvarchar(50)), and DepartmentID (int). The 'Column Properties' pane is open for the 'TeacherID' column, showing the 'Identity Specification' section where the 'Is Identity' checkbox is checked, and the 'Identity Increment' is set to 1. Red circles highlight the 'Identity Specification' section and the 'Is Identity' checkbox.

Column Name	Data Type	Allow Nulls
TeacherID	int	<input type="checkbox"/>
Name	nvarchar(50)	<input type="checkbox"/>
Surname	nvarchar(50)	<input type="checkbox"/>
DepartmentID	int	<input type="checkbox"/>

Column Properties	
(General)	
(Name)	TeacherID
Allow Nulls	No
Data Type	int
Default Value or Binding	
Table Designer	
Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	
Has Non-SQL Server Subscriber	No
Identity Specification	
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1

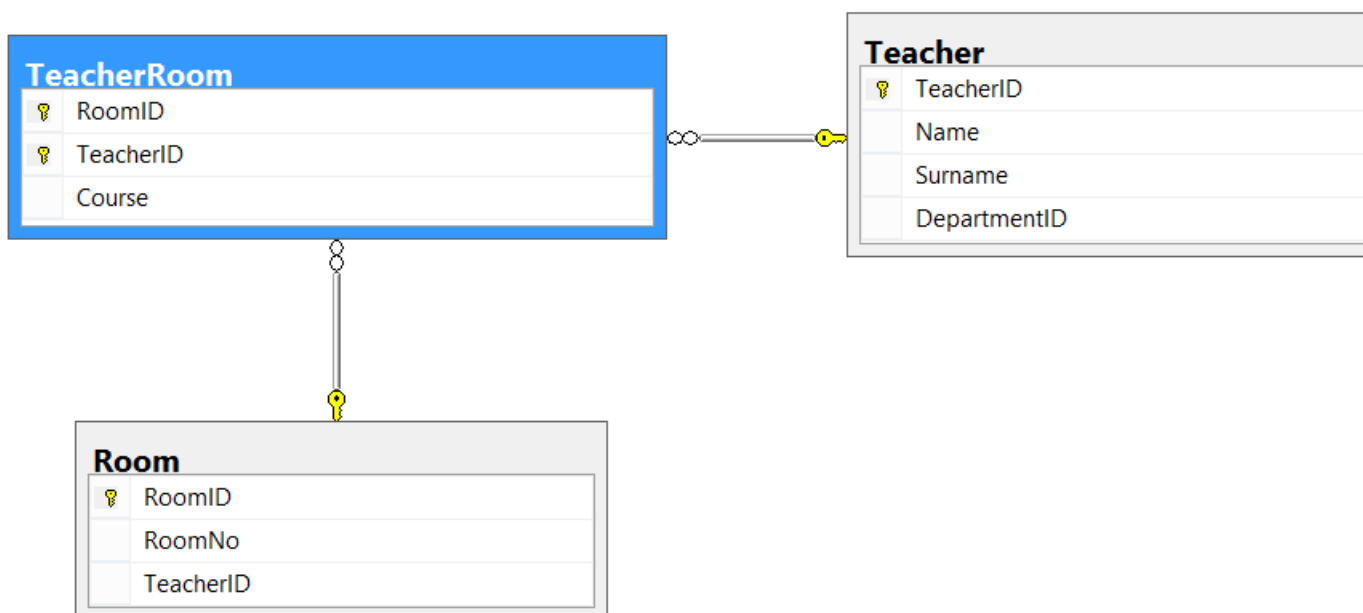


RELACJA wiele do wielu

- Relacja wiele do wielu jest drugim dość często spotykanym typem relacji w bazach danych.
- Takiej relacji nie da się nałożyć bezpośrednio między dwiema tabelami, potrzebna jest dodatkowa, odpowiednio zaprojektowana tabela – pośrednik (np. zawierająca kolumny-klucze obce odpowiadające kolumnom-kluczom głównym z tabel, które chcemy połączyć relacją wiele do wielu).



RELACJA wiele do wielu





RELACJA wiele do wielu, projektowanie tabeli - “pośrednika”

The screenshot shows the SQL Server Enterprise Designer interface. The Object Explorer on the left shows the database structure, with 'dbo.ReaderBook' selected. The main pane displays the table design with columns: ReaderID (int), BookID (int), Date (datetime), WorkerID (int), and IsReturned (bit). The Column Properties pane for 'BookID' is open, showing various settings. The 'Identity Specification' property is highlighted with a red circle and has a red text annotation 'dlaczego tutaj nie?' next to it.

Column Name	Data Type	Allow Nulls
ReaderID	int	<input type="checkbox"/>
BookID	int	<input type="checkbox"/>
Date	datetime	<input type="checkbox"/>
WorkerID	int	<input type="checkbox"/>
IsReturned	bit	<input type="checkbox"/>

Property	Value
(Name)	BookID
Allow Nulls	No
Data Type	int
Default Value or Binding	
Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	No
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No



Rodzaje baz danych cd. Aplikacje dla baz danych

- Bazy danych można dzielić na różne sposoby (zob. też o korpusach np. http://katarzyna.klessa.pl/wp-content/uploads/2014/10/Bazy_danych_Klessa_05_BazyJezykowev2.pdf)
- Ze względu na sposób wykorzystania danych w bazie można wyróżnić także: bazy **operacyjne i analityczne**
- W związku z powyższym rozróżnieniem wyróżnia się też dwa rodzaje aplikacji służących do pracy na bazie i do analizy danych w niej zawartych
(Zob też tutaj: <http://datawarehouse4u.info/OLTP-vs-OLAP.html>)



Operacyjne bazy danych

To bazy robocze, produkcyjne, używane na co dzień w działaniu różnych instytucji i firm.

Dane przechowywane w tych bazach podlegają zmianom, które odzwierciedlają zmiany w świecie rzeczywistym, którego model stanowi baza.

Np. bazy danych dla działających kont bankowych, systemy sprzedaży w sklepach, sieciowa obsługa firm, np. stacji paliw (dane do faktury dostępne na wszystkich stacjach danej sieci), systemy sprzedaży i rezerwacji biletów, połączone z obsługą urządzeń ('biletomatów'), systemy fiskalne, systemy śledzące i in.



Analityczne bazy danych

To bazy bardziej statyczne - np. statyczne archiwa językowe, bazy wykorzystywane do przechowywania danych dotyczących określonego momentu/okresu w czasie lub stanu rzeczy.

Przechowywane w nich dane są rzadko modyfikowane i odzwierciedlają pewien ustalony stan rzeczywistości, baza stanowi bardziej statyczny model zdarzeń lub stanu wiedzy.



Oprogramowanie dla baz - Online Transaction Processing (OLTP)

- Z bazami operacyjnymi często powiązane są aplikacje z kategorii tzw. Online Transaction Processing (OLTP) (bieżące przetwarzanie transakcji). Zwykle charakteryzują się one wykorzystaniem dużej liczby stosunkowo prostych zapytań np. SELECT, INSERT, UPDATE, DELETE.
- Ze względu na dynamikę działania - często w czasie rzeczywistym - istotnym parametrem dla tych aplikacji będzie szybkość wykonywania zapytań, ich optymalizacja przy jednoczesnym zapewnieniu integralności danych.



Oprogramowanie dla baz OnLine Analytical Processing (OLAP)

- Z bazami analitycznymi często powiązane są aplikacje z kategorii tzw. OnLine Analytical Processing (OLAP) (oprogramowanie do analiz wielowymiarowych).
- Zwykle charakteryzują się one wykorzystaniem mniejszej liczby zapytań, ale o bardziej złożonej strukturze, mogą one wspierać tworzenie wielowymiarowych widoków i hierarchii dla danych zawartych w bazie.
- Dzięki temu możliwa jest efektywniejsza analiza danych lub ich wybranych podzbiorów. Istotnym parametrem dla tych aplikacji będzie np. czas odpowiedzi.



Dziękuję za uwagę!

