informatyka+

Algorytmika i programowanie

Bazy danych

Multimedia, grafika i technologie internetowe

Sieci komputerowe

Tendencje w rozwoju informatyki i jej zastosowań

Człowiek – najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Warszawska Wyższa Szkoła Informatyki

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

informatyka+

Wszechnica Poranna: Bazy danych

Bazy danych – jak je ugryźć

Andrzej Ptasznik

Zenon Gniazdowski

Człowiek – najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Warszawska Wyższa Szkoła Informatyki

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Bazy danych – jak je ugryźć

Rodzaj zajęć: Wszechnica Poranna
Tytuł: Bazy danych – jak je ugryźć
Autor: mgr inż. Andrzej Ptasznik, dr hab. inż. Zenon Gniazdowski

Redaktor merytoryczny: prof. dr hab. Maciej M Sysło

Zeszyt dydaktyczny opracowany w ramach projektu edukacyjnego **Informatyka+** — ponadregionalny program rozwijania kompetencji uczniów szkół ponadgimnazjalnych w zakresie technologii informacyjno-komunikacyjnych (ICT).
www.informatykaplus.edu.pl
kontakt@informatykaplus.edu.pl

Wydawca: Warszawska Wyższa Szkoła Informatyki
ul. Lewartowskiego 17, 00-169 Warszawa
www.wwsi.edu.pl
rektorat@wwsi.edu.pl

Projekt graficzny: FRYCZ I WICHA

Warszawa 2009

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Warszawska Wyższa Szkoła Informatyki

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Bazy danych – jak je ugryźć

Andrzej Ptasznik
Warszawska Wyższa Szkoła Informatyki
aptaszni@wwsi.edu.pl

Zenon Gniazdowski
Warszawska Wyższa Szkoła Informatyki
gniazd@ite.waw.pl

Streszczenie

Wykład. Na początku zostaną wprowadzone podstawowe definicje związane bazami danych i relacyjnym modelem danych i omówione zostaną cechy tabeli relacyjnej. Opisany zostanie Systemu Zarządzania Bazami Danych oraz wybrane mechanizmy odpowiedzialne za spójność i integralność danych. Przedstawiony zostanie przykładowy model bazy danych i omówione zostaną podstawowe cechy dobrego projektu. Następnie zdefiniowany zostanie klucz podstawowy i obcy i omówione zostanie ich znaczenie, zwłaszcza w kontekście powiązania danych zapisanych w różnych tabelach. Przedstawione zostaną przykłady wykorzystania baz danych w życiu codziennym oraz zaprezentowany zostanie przykład wykorzystania wyszukiwarki internetowej nowej generacji – **wolframalpha.**

Warsztaty. W ramach warsztatów przedstawiony zostanie sposób instalacji MS SQL Server 2008 Express Edytion a następnie uczniowie zostaną zapoznani z SQL Server Menagement Studio, narzędziem klienckim umożliwiającym korzystanie i administrowanie SQL Serwerem. W kolejnych ćwiczeniach będzie tworzona pierwsza baza danych, zdefiniujemy tabele oraz dla wybranych przypadków określimy reguły poprawności. Następnie zdefiniujemy klucze podstawowe i obce oraz reguły integralności referencyjnej. Nauczymy się zapisywać dane w tabelach oraz wykonywać modyfikacje danych już istniejących. Sprawdzimy jak działają zdefiniowane wcześniej reguły poprawności w trakcie modyfikacji danych. Napiszemy pierwsze proste zapytania do utworzonej bazy danych.

Spis treści

Wykład

Warsztaty

WYKŁAD

1 KILKA DEFINICJI NA DOBRY POCZĄTEK

Wykład jest wprowadzeniem do tematyki baz danych ze szczególnym uwzględnieniem baz opartych na relacyjnym modelu danych. Wprowadzimy na początku kilka definicji wyjaśniających podstawowe pojęcia, którymi będziemy się posługiwali w dalszej części.

- **Dane to liczby, znaki, symbole (i cokolwiek innego) zapisane w celu ich przetwarzania.**
- **Informacja to taki czynnik, któremu człowiek może przypisać określony sens (znaczenie), aby móc ją wykorzystywać do różnych celów.**
- **Wiedza to, zgodnie z prastarą definicją Platona, ogół wiarygodnych informacji o rzeczywistości wraz z umiejętnością ich wykorzystywania.**

Można teraz uporządkować te pojęcia w kontekście baz danych i sposobu ich wykorzystania. Dane zbieramy i zapisujemy, by na ich podstawie uzyskiwać informacje, które będą stawały się wiedzą, gdy uzupełnimy je o sposoby i możliwości ich praktycznego wykorzystania. Nie trzeba chyba udowadniać tezy, że współczesne społeczeństwo, społeczeństwo informacyjne, opiera swoje działania i podstawy rozwoju na wiedzy, która jest między innymi pozyskiwana z baz danych.

Zgodnie z powyższymi definicjami dane to prawie wszystko co może być zapisane i dlatego jednym z podstawowych zadań, żeby zbiór danych mógł stać się bazą danych, jest zapewnienie odpowiedniego sposobu uporządkowania. Bazy danych mogą gromadzić gigantyczne ilości danych zapewniając właściwy sposób ich uporządkowania. Teraz możemy spróbować zdefiniować pojęcie bazy danych :

Baza danych to zbiór danych zapisanych w ściśle określony sposób w strukturach odpowiadających założonemu modelowi danych.

2 DANE I BAZY DANYCH

Danymi mogą być dowolne wartości znakowe, liczbowe, graficzne i jakiekolwiek inne, które przechowujemy na dowolnych nośnikach w celu ich przetwarzania. Za pomocą różnych danych można opisywać obiekty i zdarzenia zachodzące w rzeczywistości. Przykładowym „obiektem" opisywanym za pomocą danych może być uczeń i jego opis może mieć następującą postać:

'Janina' 'Maczek' '12-09-1992' '92091298787' 'Opole' '0504098765'

Dla tak przedstawionych danych o uczniu konieczne jest podanie znaczenia poszczególnych wartości. Co znaczy słowo **'Opole'** – czy jest to pseudonim opisywanego ucznia, miejsce jego urodzenia a może miejsce zamieszkania? Przedstawiony przykład ilustruje, że dane pozbawione możliwości ich jednoznacznej interpretacji są zupełnie bezużyteczne. Co więcej, istotna jest nie tylko możliwość interpretacji danych, ale także zapewnienie sposobu ich uporządkowania, szczególnie w sytuacji gdy przechowujemy bardzo dużo różnych danych.

Za pomocą danych możemy opisywać także pewne zdarzenia i w takim przypadku problemy związane z interpretacja danych mogą być jeszcze większe – co może oznaczać następująca porcja danych:

'Jan' 'Nowak' '2009-11-09' 5 'Zenon' 'Marczak' 'angielski'

Taki zestaw danych, w zależności od interpretacji może przyjmować różne znaczenie i bez jasno określonych zasad jest praktycznie bezużyteczny.

Uczeń o nazwisku Jan Nowak 9 listopada 2009r, otrzymał ocenę 5 z języka angielskiego, która wystawił nauczyciel o nazwisku Zenon Marczak – to tylko jedna z możliwych interpretacji podanego zestawu danych.

Powróćmy do przedstawionej wcześniej definicji bazy danych. Dochodzimy do wniosku, że baza danych ma umożliwić gromadzenie dużych ilości różnych danych według ściśle określonych zasad, które muszą zapew-

nić porządek wśród zapisywanych danych i ich właściwą interpretację. W erze komputeryzacji bazy danych stały się jedną z podstawowych dziedzin zastosowań. Zdecydowana większość różnych systemów informatycznych opiera się na bazach danych, a w życiu codziennym korzystamy z baz danych nawet nie zdając sobie z tego sprawy, na przykład:

- każdemu obywatelowi Polski po urodzeniu zostaje przydzielony numer PESEL – czyli zostaje zapisany w pewnej bazie danych obywateli Naszego Kraju;
- korzystając z internetowego rozkładu jazdy odczytujemy dane z bazy danych ruchu pociągów;
- przy korzystaniu z bankomatu – bez połączenia z odpowiednią bazą danych w banku transakcja jest niemożliwa
- korzystając z pomocy w dowolnej aplikacji odczytujemy komunikaty z bazy danych, w której zgromadzono hasła pomocy.

Doszliśmy do wniosku, że bazy danych są powszechnie wykorzystywane i tym samym są bardzo ważna dziedziną informatyki, dlatego warto im się przyjrzeć bliżej by zrozumieć jak są zorganizowane i na jakich zasadach są oparte i warto także się przyjrzeć niektórym problemom, które muszą być rozwiązane, żeby baza danych mogła spełniać założone funkcje.

3 PODSTAWY RELACYJNEGO MODELU DANYCH

W dotychczasowych rozważaniach posługiwaliśmy się bardzo ogólnym pojęciem bazy danych i wystarczało nam stwierdzenie, że baza danych jest oparta na pewnych zasadach, które mają zapewnić porządek na etapie przechowywania danych i muszą zapewnić możliwości poprawnej ich interpretacji. W podanej definicji bazy danych jest stwierdzenie, że bazą danych stają się dane, zapisane w ściśle określony sposób według założonego **modelu danych**.

W naszych rozważaniach będziemy zajmować się relacyjnym modelem danych, na którym opiera się większość współczesnych baz danych. Ojcem relacyjnego modelu danych i baz danych, opartych na tym modelu, jest E.F. Codd, którego praca, opublikowana w roku 1970 rozpoczęła erę burzliwego rozwoju relacyjnych baz danych. Pierwsze komercyjne systemy baz danych oparte na tym modelu zostały zaimplementowane na początku lat 80. XX wieku i od tego momentu są w powszechnym użyciu. Model relacyjny zawdzięcza swój sukces bardzo solidnym podstawom teoretycznym, opartym na teoriach matematycznych, np. relacji i zbiorów. Podstawy matematyczne przyczyniły się do dobrej implementacji baz danych. Ponadto, matematyczną interpretację relacji można zastąpić intuicyjnym pojęciem tabeli, dzięki temu wielu specjalistów korzystających z baz danych nie ma większych trudności ze zrozumieniem podstaw tego modelu. Opis relacyjnego modelu danych można podzielić na trzy części:

- **struktury danych – czyli, w jaki sposób i według jakich zasad organizujemy przechowywanie danych oraz według jakich zasad należy je projektować;**
- **języki manipulowania danymi – czyli, w jaki sposób zapisywać, modyfikować, usuwać oraz pobierać dane znajdujące się w bazie danych**
- **integralność danych – czyli, w jaki sposób zapewnić poprawność przechowywanych danych.**

Wszystkie dane w bazie relacyjnej są przedstawione w postaci dwuwymiarowych **tabel**, zwanych **relacjami**. Projektowanie bazy danych sprowadza się do zdefiniowania grupy tabel opisujących dziedzinę, dla której chcemy utworzyć bazę danych. Pojęcie tabeli jest dobrze znane, ale nie każda tabela może zostać uznana za spełniającą cechy modelu relacyjnego. Dla tabeli relacyjnej sformułowano pewne zasady, które muszą być przestrzegane, żeby można ją było uznać za poprawną z punktu widzenia założeń modelu relacyjnego. Spróbujmy przyjrzeć się niektórym z tych zasad.

- **Każda tabela w bazie danych ma jednoznaczną nazwę.**
  Zasada ta nie powinna budzić wątpliwości, gdyż w przeciwnym wypadku nie można by było jednoznacznie identyfikować tabeli

- **Każda kolumna tabeli ma jednoznaczną nazwę w obrębie tej tabeli.**
  Powód i znaczenie tej zasady jest analogiczny jak powyżej, gdyż jeżeli tabela zawiera wiele kolumn to musimy je rozróżniać.

- **Wszystkie wartości w kolumnie są tego samego typu.**
  Jeżeli w pewnej tabeli istnieje kolumna o nazwie ***DataUrodzenia***, to znaczy, że chcemy tam przechowywać dane określające datę i wszystkie wartości zapisane w takiej kolumnie muszą być poprawną datą. W innym przypadku będziemy mieli poważne problemy z wykorzystaniem i interpretacją zapisanych danych. Nieprzestrzeganie tej zasady mogłoby doprowadzić do sytuacji pokazanej w poniższej tabeli.

Nazwisko	Imie	Pesel	DataUrodzenia	IloscRodzenstwa
Kot	Jan	92091187345	1992-09-11	2
Lisek	125-12	2009-12-12	12 grudzień 1993	brat i siostra
Wojtek	92071287765	sierpień 1992	Marczak	dwóch braci

Widać wyraźnie, że tabela z taką zawartością nie nadaje się do poważnych zastosowań – na pierwszy rzut oka możemy zauważyć totalny bałagan i trudno byłoby prawidłowo interpretować takie dane. Przykładowa tabelka spełniająca ten postulat modelu relacyjnego mogłaby wyglądać tak jak poniżej.

Nazwisko	Imie	Pesel	DataUrodzenia	IloscRodzenstwa
Kot	Jan	92091187345	1992-09-11	2
Lisek	Barbara	93121287634	1993-12-12	2
Marczak	Wojciech	92061223871	1992-06-12	2

- **W tabeli nie mogą istnieć dwa identyczne wiersze, każdy wiersz jest różny, tabela może istnieć bez wierszy.**

Nazwisko	Imie	DataUrodzenia	IloscRodzenstwa
Kot	Jan	1992-09-11	2
Kot	Jan	1992-09-11	2
Marczak	Wojciech	1992-06-12	2

Dwa pierwsze wiersze w powyższej tabeli są identyczne (teoretycznie mogą istnieć dwie osoby opisywane takimi samymi danymi), a tym samym przeczą jednej z podstawowych zasad tabeli relacyjnej. Z omawianej zasady jednoznacznie wynika, że w tabeli relacyjnej musi istnieć kolumna (lub zbiór kolumn), która dla każdego wiersza przyjmuje inną wartość. Taka kolumnę nazywamy **kluczem podstawowym** tabeli (ang. *primary key*). Dodając do omawianej tabeli dodatkową kolumnę Pesel (każdy uczeń ma inny numer Pesel) uzyskujemy klucz podstawowy a tym samym tabela spełnia kolejna cechę tabeli relacyjnej.

Pesel	Nazwisko	Imie	DataUrodzenia	IloscRodzenstwa
92091177329	Kot	Jan	1992-09-11	2
92091108431	Kot	Jan	1992-09-11	2
92061281268	Marczak	Wojciech	1992-06-12	2

Dodanie klucza podstawowego (Pesel) spowodowało, że pozornie dwa identyczne wiersze jednak się różnią.

- **W tabeli relacyjnej są przechowywane dane oparte na typach prostych (dane elementarne)**

Sytuacją niepożądana, z punktu widzenia tej cechy tabeli relacyjnej, jest zapisywanie w jednej komórce tabeli wielu danych. Przedstawiona poniżej tabela nie spełnia tak sformułowanej cechy.

Nazwisko	Imię	Adres			Języki obce	Rodzeństwo
		Kod	Ulica	Miasto		
Kot	Jasio	12-098	Nowa 33/21	Opole	Angielski, francuski, hiszpański	brat Staś, siostra Mariola
Lis	Hania	65-987	Cicha 17/2	Sopot	Angielski, niemiecki	brak
Żuk	Piotrek	33-093	Miła 4/3	Gdynia	Nie zna	brat Jaś, brat Staś, siostra Hania

W dalszej części wykładu, przy omawianiu przykładowego projektu bazy danych, zostanie pokazany sposób rozwiązania tego problemu.

- **Kolejność wierszy i kolejność kolumn w tabeli relacyjnej nie ma żadnego znaczenia – czyli położenie danej w tabeli nie wpływa na jej znaczenie.**

nazwisko	imie	pesel	data_urodzenia
Kotek	Katarzyna	92031275446	1992-03-12
Piesek	Jan	92051587746	1992-05-15
Lisek	Kasia	92022277654	1992-02-22

imie	pesel	data_urodzenia	nazwisko
Katarzyna	92031275446	1992-03-12	Kotek
Jan	92051587746	1992-05-15	Piesek
Kasia	92022277654	1992-02-22	Lisek

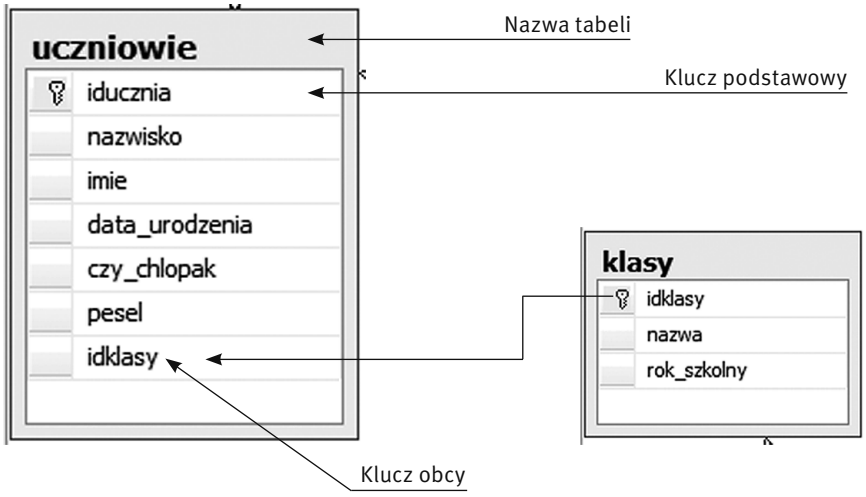
data_urodzenia	pesel	nazwisko	imie
1992-03-12	92031275446	Kotek	Katarzyna
1992-05-15	92051587746	Piesek	Jan
1992-02-22	92022277654	Lisek	Kasia

Ponieważ w modelu relacyjnym kolejność kolumn i wierszy nie ma żadnego znaczenia, to widoczne trzy postaci tabel są identyczne i można z nich pobrać dokładnie te same informacje.

4 MODELOWANIE Z WYKORZYSTANIEM TABEL RELACYJNYCH

Jedyną strukturą danych w modelu relacyjnym jest tabela ale jedna tabela może zawierać dane tylko na określony temat (dane o uczniu, dane o planowanych wizytach lekarskich itp.). Nie można utożsamiać tabeli z bazą danych, ponieważ baza danych jest pojęciem szerszym a tabele są elementami składowymi bazy danych. To, jakie tabele będzie zawierała baza danych, jest określone na etapie jej **projektowania**. W trakcie projektowania baz danych za pomocą dwuwymiarowych tabel opisujemy wybrany fragment rzeczywistości (bank, szkoła, kolekcja płyt). Teraz spróbujemy omówić niektóre aspekty i zasady projektowania baz danych, zdając sobie sprawę, że proces projektowania baz jest daleko bardziej złożony.

Zanim przystąpimy do projektowania musi zostać określona dziedzina, dla której tworzymy bazę danych. Zakładamy, że chcemy zaprojektować bazę danych umożliwiającą rejestrowanie ocen wystawianych uczniom, czyli pewien fragment szkolnej rzeczywistości. Jednym z podstawowych zadań, na etapie projekto-

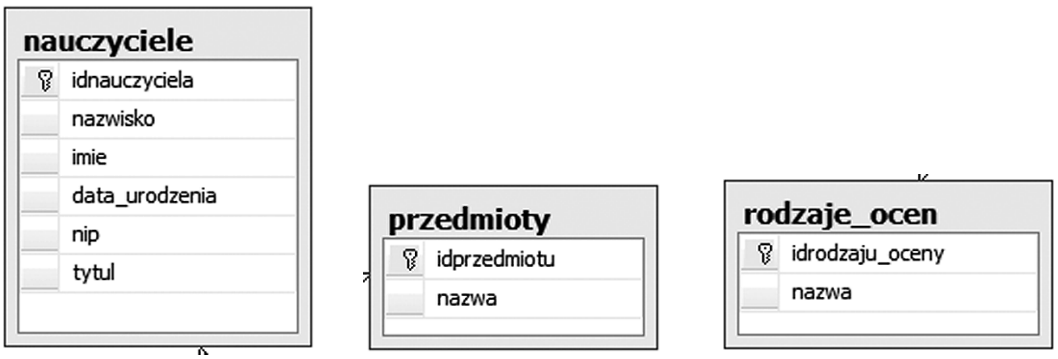


wania bazy danych, jest określenie podstawowych obiektów występujących w dziedzinie, dla której te bazę projektujemy. W naszym przypadku dość szybko dojdziemy do wniosku, że nasza baza powinna opisywać uczniów oraz podstawową jednostkę organizacyjną szkoły czyli klasy. Poniżej przedstawiamy propozycję tabel opisujących te dwa podstawowe elementy.

Wyjaśnimy teraz niektóre elementy zaproponowanego rozwiązania.

- Każda tabela musi mieć przypisaną nazwę i nazwa ta powinna określać rodzaj danych, jaki planujemy w tej tabeli przechowywać.
- Zgodnie z cechami modelu relacyjnego, każda tabela musi zawierać klucz podstawowy. Dla zaproponowanych tabel kluczami podstawowymi są kolumny o nazwach ***iducznia*** i ***idklasy*** – na pierwszy rzut oka nazwy te wydaja się dziwne ale odpowiada to pewnej przyjętej praktyce polegającej na tym, że projektując tabele ustala się tzw. **sztuczny klucz podstawowy**. W naszym przypadku kolumna ***iducznia*** (identyfikator ucznia) będzie zawierała unikatowe numery przypisywane każdemu uczniowi, który zostanie w tabeli zapisany. W bazach danych istnieją mechanizmy, które automatycznie generują unikatowe numery dla tak zdefiniowanych kluczy.
- W tabeli ***uczniowie*** znajduje się kolumna o nazwie ***idklasy*** (na rysunku nazwana kluczem obcym). Wymaga to wyjaśnienia tym bardziej, że dotykamy istoty projektowania relacyjnych baz danych. Sprawą oczywistą jest to, że każdy uczeń jest przypisany do określonej klasy. Klasy, jako takie, są zapisywane w oddzielnej tabeli o nazwie ***klasy*** w której kluczem podstawowym jest kolumna ***idklasy***. Fakt umieszczenia w tabeli ***uczniowie*** klucza obcego (czyli kolumny ***idklasy***, która w innej tabeli pełni role klucza podstawowego) tworzy związek (powiązanie) pomiędzy tabelami ***uczniowie*** i ***klasy***. Spróbujmy wyjaśnić dlaczego w ten sposób projektuje się elementy relacyjnych baz danych.
  - Faktem jest, że każdy uczeń jest przypisany do określonej klasy i teoretycznie moglibyśmy umieścić w tabeli ***uczniowie*** dodatkowe kolumny opisujące tę klasę (nazwę klasy i rok szkolny), ale w tej sytuacji dla uczniów tej samej klasy dane takie musiałyby być powtórzone, czyli to samo byłoby zapisywane w tabeli wielokrotnie. Sytuacja taka sprzyja powstawaniu błędów i niejednoznaczności, których przyczyną mogą być zwykłe błędy (literówki) na etapie zapisywania danych do tabeli.
  - Zapisując dane o klasach w osobnej tabeli zapewniamy, że dana klasa opisana jest tylko jeden raz.
  - Umieszczenie w tabeli ***uczniowie*** klucza obcego ***idklasy*** zapewnia powiązanie danych o uczniu z danymi o klasie do której został dowiązany.
  - Jeżeli w pewnym wierszu tabeli ***uczniowie*** mamy zapisane dane ucznia i przykładowo w kolumnie ***idklasy*** zapisana jest liczba 5, to taki zapis interpretujemy w ten sposób: uczeń związany jest z klasą (zapisaną w tabeli ***klasy***) o wartości klucza 5. Ponieważ ***idklasy*** w tabeli ***klasy*** jest kluczem podstawowym to mamy gwarancję, że nasza przykładowa wartość 5, odpowiada dokładnie jednemu wierszowi tabeli ***klasy*** zawierającemu interesujący nas opis.

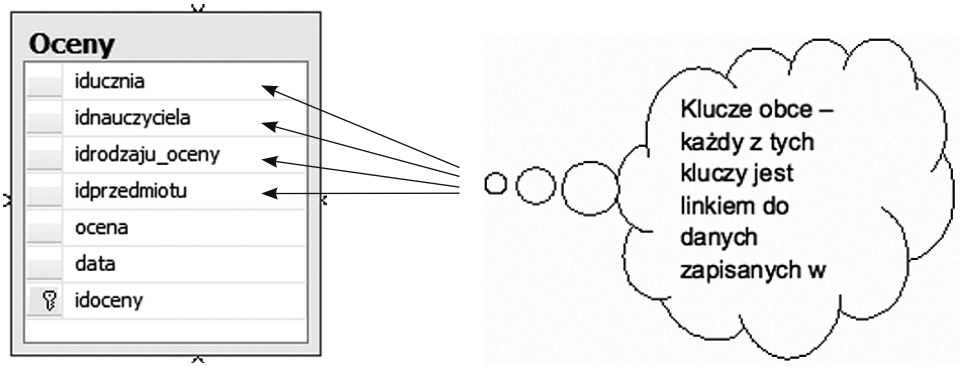
Kontynuujemy nasz projekt i kolejnym elementem może być tabela opisująca nauczycieli, ponieważ nie można wyobrazić sobie procesu wystawiania ocen bez wiedzy o nauczycielu, który taką ocenę wystawił. Proponowana tabela ***nauczyciele*** nie wprowadza żadnych nowych elementów do rozważań.



Kilka słów wyjaśnienia przy propozycji kolejnych tabel w naszym projekcie. Zaproponowane tabele o nazwach ***przedmioty*** i ***rodzaje_ocen*** są tak zwanymi **tabelami słownikowymi**, czyli takimi, które będą przechowywać zbiory pewnych pojęć. Cel dla którego projektujemy tego typu tabele wydaje się oczywisty – będziemy

wykorzystywać klucze podstawowe z tych tabel jako klucze obce w innych tabelach zawierających informacje o przedmiocie lub rodzaju wystawionej oceny i. podobnie jak we wcześniej opisywanym przypadku (***uczniowie*** i ***klasy***), dzięki takiemu podejściu zapewnimy jednoznaczność zapisywanych danych.

Na koniec tabela, która jest najważniejsza z punktu widzenia zaplanowanej bazy danych, czyli tabela w której będziemy przechowywali dane opisujące wystawione oceny.

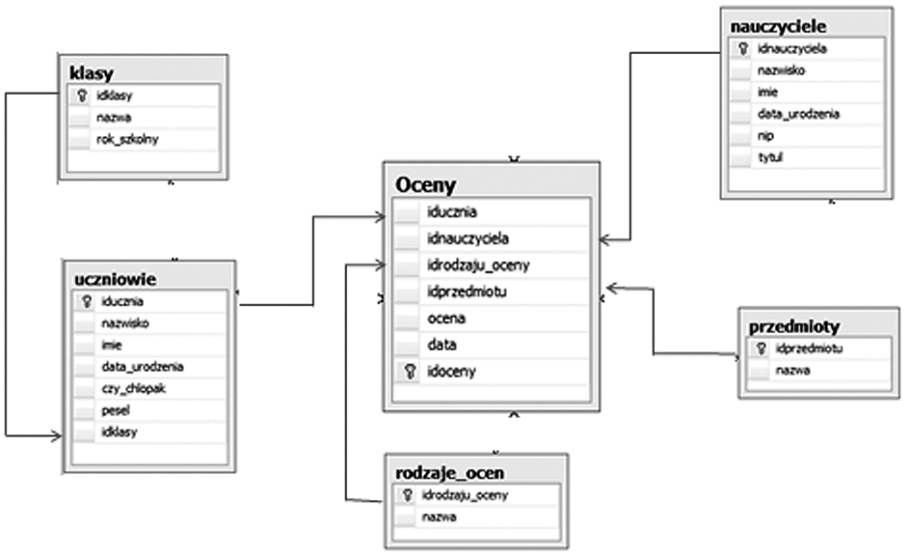


W tabeli ***Oceny*** są przechowywane dane o pewnych zdarzeniach – wystawionych ocenach. Należy się spodziewać, że ta tabela będzie centralnym punktem naszej bazy danych. Zawartość tej tabeli można opisać następująco:

Pewien uczeń (***iducznia***) od jakiegoś nauczyciela (***idnauczyciela***) otrzymał pewien rodzaj oceny (***idrodzaju_oceny***) z pewnego przedmiotu (***idprzedmiotu***) o wartości oceny (***ocena***) wystawionej pewnego dnia (***data***). Dodatkowo, w tabeli ***Oceny*** jest kolumna ***idoceny***, czyli sztuczny klucz podstawowy, który już poznaliśmy.

Na zakończenie tej części naszych rozważań przedstawmy w całości nasz projekt bazy danych.

Baza danych „Elektroniczny dziennik ocen”



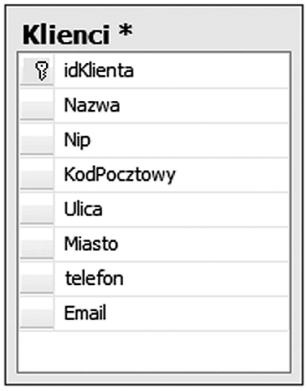
Możemy uznać, że tak zaprojektowana baza danych będzie spełniać rolę miejsca, w którym gromadzone będą dane o wystawianych ocenach. Nic nie stoi na przeszkodzie, aby gromadzić w niej dane o wszystkich ocenach wystawianych w danej szkole. Rodzi się jednak pytanie – a jeśli chcielibyśmy zapisywać w takiej bazie danych oceny wystawiane w różnych szkołach. Czy taki projekt bazy byłby wystarczający? Jak należałoby zmodyfikować ten projekt, aby umożliwić zapisywanie ocen wystawianych w różnych szkołach?

Realizacja bazy ocen nie jest wystarczającym doświadczeniem, by móc uznać, że jesteśmy już projektantami baz danych. Problemy związane z projektowaniem baz są dalece bardziej złożone i wymagają oprócz wiedzy bardzo dużego doświadczenia. Zrobiliśmy tylko pierwszy krok. Dla ugruntowania naszych umiejęt-

ność przeprowadzimy jeszcze analizę innego przypadku, aby pokazać, że przy projektowaniu baz danych pojawia się wiele problemów, można je jednak właściwie rozwiązywać.

Baza danych klientów firmy

Wyobraźmy sobie sytuację, że w pewnej bazie danych istnieje tabela o nazwie ***Klienci*** o następującej strukturze:

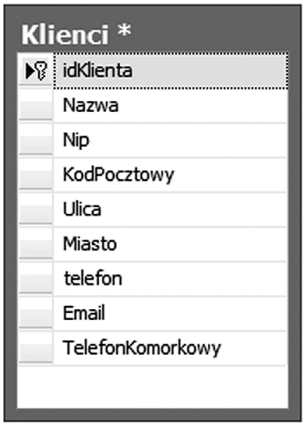


Na pierwszy rzut oka w pokazanej tabeli jest wszystko w porządku i nie widać problemów. Przykładowa zawartość takiej tabeli mogłaby wyglądać tak, jak poniżej :

idKlienta	Nazwa	Nip	KodPocztowy	Ulica	Miasto	telefon	Email
1	Fiat Auto Poland	7161954205	04-985	Krzywa 15	Tychy	032 567 34 55	firma@fiat.pl
2	Wedel S.A.	5129087671	00-978	Prosta 22/24	Warszawa	022 698 34 99	*NULL*
3	Goplana S.A	6789873452	66-432	Wawrzyńca 18	Poznań	*NULL*	*NULL*

W dalszym ciągu nie widać problemów – zapisy w tabeli wyglądają zupełnie sensownie, jedynie zwraca uwagę brak pewnych danych, co jest oznaczane wartością ***NULL*** – nie jest to błąd, taka wartość jest dopuszczalna w bazach danych.

Może pojawić się problem, gdy użytkownicy takiej bazy danych będą natrafiać na sytuacje, których nie przewidzieliśmy na etapie projektowania. Na przykład: użytkownicy bazy danych chcieliby przechowywać dodatkowe dane, będące numerem telefonu komórkowego. Tak postawiony problem można rozwiązać bardzo prosto poprzez dodanie do tabeli nowej kolumny przeznaczonej do przechowywania numerów telefonów komórkowych. Zmodyfikowana tabela miałaby teraz strukturę, jak poniżej.



Zawartość tej tabeli wskazuje, że problem został rozwiązany, pomijając fakt, że dodanie nowej kolumny do tabeli w już istniejącej i działającej bazie danych wcale nie jest sprawą tak prosta jak tutaj pokazujemy. Pozostaje jeszcze pytanie, czy ta modyfikacja rozwiązuje problem na dłużej, czy mamy pewność, że w trakcie dal-

szej eksploatacji tej bazy danych nie będzie potrzeby dodania kolejnych kolumn, np. aby zapisać więcej niż jeden numer telefonu albo adres strony www, numer faksu, numer GG ... itd.

idKlienta	Nazwa	Nip	KodPocztowy	Ulica	Miasto	telefon	Email	TelefonKomorkowy
1	Fiat Auto Poland	7161954205	04-985	Krzywa 15	Tychy	032 567 34 55	firma@fiat.pl	696 456789
2	Wedel S.A.	5129087671	00-978	Prosta 22/24	Warszawa	022 698 34 99	*NULL*	*NULL*
3	Goplana S.A	6789873452	66-432	Wawrzyńca 18	Poznań	*NULL*	*NULL*	504 044784

Widać wyraźnie, że zaproponowane przez nas rozwiązanie nie jest trwałe i elastyczne, czyli bardzo łatwo, w trakcie eksploatacji bazy danych, może się okazać, że nasze tabele nie są w stanie zapisać danych, które są potrzebne użytkownikom.

A może chwila zastanowienia i rozwiązać ten problem raz a dobrze? Widać, że problemem staje się ustalenie jakie dane mogą być nam potrzebne. Zaproponujemy w takim razie utworzenie tabeli, która będzie takie dane przechowywał – we wcześniej omawianym projekcie występowały podobne tabele, zwane tabelami słownikowymi. Poniżej pokazana jest struktura takiej tabeli oraz przykładowa jej zawartość.

idRodzajuKontaktu	Nazwa
1	Telefon stacjonarny
2	Telefon komórkowy
3	E-mail

Jeżeli będziemy dodatkowo potrzebowali przechowywać w bazie danych informacje o numerach Gadu Gadu i adresy stron WWW – to wystarczy dopisać kolejne wiersze do tabeli i otrzymamy tabelę z odpowiadającą nam zawartością. Tym razem nie wymaga to dodania nowych kolumn do istniejącej tabeli, co jest czynnością trudną i złożoną w działającej bazie danych, a jedynie dopisanie nowego wiersza lub wierszy do istniejącej tabeli, a to jest czynnością całkowicie naturalna w bazie danych. Poniżej jest pokazana tabela z uzupełnionymi wpisami.

idRodzajuKontaktu	Nazwa
1	Telefon stacjonarny
2	Telefon komórkowy
3	E-mail
4	Numer GaduGadu
5	Strona WWW

Wykonaliśmy pierwszy krok, ale do rozwiązania problemu droga jeszcze daleka. W kolejnym kroku modyfikujemy tabelę ***Klienci***, usuwając z niej wszystkie kolumny przechowujące dane o kontaktach. Struktura tabeli po modyfikacji jest pokazana poniżej.

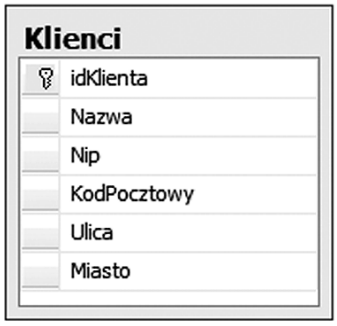


Tabela ta została trochę odchudzona i tym samym uproszczona, ale w dalszym ciągu nie mamy miejsca na zapisywanie danych o kontaktach konkretnych klientów. Ostatnią tabelą, którą musimy utworzyć, jest tabe-

la przeznaczona do przechowywania danych o kontaktach. Dane zapisane w takiej tabeli powinny zawierać uzupełnienie poniższego zdania:

Z pewnym klientem (***idklienta***) istnieje pewien rodzaj kontaktu (***idRodzajuKontaktu***), z którym związany jest numer (***numer*** – rozumiany jako numer telefonu, GG, adres e-mail itp.) oraz pewne dodatkowe uwagi (***uwagi***).

Struktura tabeli odpowiadającej takim wymaganiom jest pokazana poniżej. Ta tabela jest podobna, w podstawowych założeniach, do tabeli ***Oceny*** z wcześniej projektowanej bazy danych. Tabelę tego typu nazywamy **tabelą powiązań** (**asocjacyjną**).

Przykładowa zawartość takiej tabeli jest pokazana poniżej. Na pierwszy rzut oka dane zapisane w tej tabeli wydaja się mało czytelne, ale pamiętajmy o tym, że dzięki kluczom obcym (***id klienta***, ***idRodzajuKontaktu***) mamy dostęp do dodatkowych danych zapisanych w powiązanych tabelach.

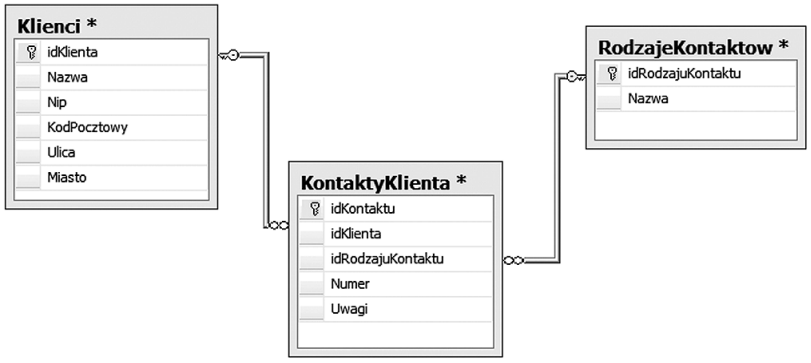
idKontaktu	idKlienta	idRodzajuKontaktu	Numer	Uwagi
4	1	1	032 567 34 55	Dzwonić do 15.00
5	2	1	022 698 34 99	Prosic Panią Basię
6	1	2	696 456789	Brak uwag
7	3	2	504 044784	Po dwóch sygnałach włacza się sekretarka
8	1	2	firma@fiat.pl	Często nie odpowiada na pocztę

W tym miejscu, wyprzedzając nieco naszą podróż po krainie baz danych, możemy pokazać, jak wyglądałyby dane z tej tabeli po ich zinterpretowaniu z wykorzystaniem kluczy obcych. Zadanie takie w bazach danych jest realizowane poprzez odpowiednie zapytania. Chociaż nie wiemy jeszcze jak takie zapytania realizować, to na kolejny rysunku jest pokazany wynik zapytania zrealizowany na pokazanych wcześniej przykładowych danych.

Klient	RodzajKontaktu	Numer	Uwagi
Fiat Auto Poland	Telefon stacjonarny	032 567 34 55	Dzwonić do 15.00
Wedel S.A.	Telefon stacjonarny	022 698 34 99	Prosic Panią Basię
Fiat Auto Poland	Telefon komórkowy	696 456789	Brak uwag
Goplana S.A	Telefon komórkowy	504 044784	Po dwóch sygnałach włacza się sekretarka
Fiat Auto Poland	E-mail	firma@fiat.pl	Często nie odpowiada na pocztę

Teraz zaprezentowane dane są bardziej czytelne – w ten sposób dane przekształciliśmy w pewną informację.

Na koniec naszych rozważań zaprezentowany został wykonany przez nas fragment projektu, w którym zamiast jednej tabeli ***Klienci*** są trzy tabele widoczne na rysunku poniżej. Pojawia się pytanie, czy „gra warta była świeczki" ale odpowiedź na takie pytanie wcale nie jest jednoznaczna. Z jednej strony bowiem poprawiliśmy projekt, który teraz jest bardziej elastyczny, a z drugiej – projekt stał się bardziej złożony i jego obsługa, przy korzystaniu z bazy danych, też będzie bardziej skomplikowana. Tego typu pytania są codziennością w trakcie projektowania i eksploatowania baz danych – na ogół, gdy poprawiamy jeden aspekt rozwiązania, to na ogół kosztem innego aspektu, zatem podstawowym wyzwaniem etapu projektowania jest wybór odpowiedniego kompromisu.



Autor woli rozwiązanie elastyczne, niezależnie od kosztu jaki należy ponieść, ale jest to pogląd jednego specjalisty, a można spotkać inne.

5 PROBLEMY I ANOMALIE ZWIĄZANE Z GROMADZENIEM DANYCH W TABELACH

Po burzy mózgów związanych z projektowaniem baz danych przechodzimy teraz do zagadnień związanych z ich eksploatacją. Do tej pory zakładaliśmy, że do zaprojektowanych tabel są zapisywane poprawne dane, ale może to okazać się problemem, gdy pojawiają się błędne wpisy. Jeżeli nie zagwarantujemy wymuszenia poprawności zapisywania danych, to bardzo szybko nasza baza danych zamieni się w śmietnik bezużytecznych i nikomu nie potrzebnych zapisów. W tej części zastanowimy się nad niektórymi problemami, jakie mogą się pojawić, a dalej pokażemy sposoby ich rozwiązania.

Gromadzenie danych w tabeli nie może się odbywać bez reguł i ograniczeń – nie wystarczy samo nazwanie kolumn, bo możemy się spotkać z problemem pokazanym na kolejnym rysunku. Można sobie wyobrazić istnienie tabeli z przykładową zawartością (jak poniżej) i zastanówmy się, czy dopuszczenie do takiej sytuacji ma jakikolwiek sens.

Pesel	Nazwisko	Imię	DataUrodzenia	Płeć	Wiek
92092256787	Kotek	Janina	1992-09-22	Kobieta	17
921105au34	Lisek	Piotr	1992-07-21	Kotek	33
Wiktor	23	Lis	8 maj 91	Chłopak	OK

Jak widać, w tabeli bardzo łatwo może zapanować totalny bałagan. Możemy w tym miejscu dojść do wniosku, że bazy danych powinny zawierać mechanizmy ułatwiające wymuszanie poprawności zapisywanych danych. W pokazanym przykładzie występuje kilka problemów :

- Nazwa kolumny nie gwarantuje zapisywania w niej właściwych danych
- Gdy mamy zapisane błędne dane – baza danych traci sens.
- W powyższym przykładzie – numer Pesel powinien być zależny od daty urodzenia

Zobaczmy jeszcze jeden przykład, który będzie nawiązywał do zaprojektowanej przez nas bazy danych ***Elektroniczny dziennik ocen***. Wyobraźmy sobie istnienie w pewnej bazie danych tabeli z następującą zawartością:

idoceny	Uczen	AdresUcznia	Nauczyciel	Przedmiot	DataOceny	Ocena	RodzajOceny
1	Jan Kotek	Warszawa ul.Miła 7	Piotr Poważny	Historia	2009-10-12...	4,00	Kartkówka
2	Daria Miła	Płock ul.Naftowa 23	Maria Łaskawa	Fizyka	2009-10-11...	3,00	Praca domowa
4	Kotek Jan	Warszawa ul.Miła 7	Zenon Piguła	Matematyka	2009-10-10...	5,00	Sprawdzian
5	Daria Miła	Płock ul.Benzynowa 23	Zenon Piguła	Chistoria	2009-10-16...	3,00	Sprawdz.
7	Lisowski Paweł	Opole ul.Nowa 12	Paulina Mocna	Język polski	2009-11-22...	4,00	Kartkówka

Widać, że jest to odpowiednik naszego elektronicznego dziennika ocen, tylko utworzony na bazie jednej tabeli. Analiza zawartość tej przykładowej tabeli, dostarczy odpowiedzi na pytanie, dlaczego nie zaprojektowaliśmy tylko jednej tabeli. Analizując zawartość tej tabeli napotykamy na wątpliwości i pytań – oto niektóre z nich:

1. Czy Jan Kotek i Kotek Jan – to ta sama osoba?
2. Czy Daria Miła mieszka na ulicy Naftowej czy Benzynowej?
3. Czy Sprawdzian i Sprawdz. to ten sam rodzaj oceny?
4. Czy Historia i Chistoria to ten sam przedmiot?

Warto w tym miejscu odnieść te problemy do naszego rozwiązania i spróbować odpowiedzieć na postawione pytanie.

Ad 1. W naszym projekcie ten problem nie występuje, ponieważ uczniowie są zapisani w odrębnej tabeli i zapisując ocenę podajemy tylko klucz podstawowy z tabeli ***Uczniowie***, czyli ***iducznia***, a to jednoznacznie wiąże ocenę z jednym konkretnym uczniem (cecha klucza podstawowego).

Ad 2. Problem nie istnieje z tego samego powodu.

Ad 3. Problem nie istnieje, ponieważ zaprojektowaliśmy tabelę słownikową Rodzaje_ocen i zapisując ocenę podajemy tylko wartość klucza, który jednoznacznie jest związany z jednym konkretnym rodzajem oceny.

Ad 4. Problem nie istnieje, ponieważ zaprojektowaliśmy tabelę słownikową ***Przedmioty*** i zapisując ocenę podajemy tylko wartość klucza, który jednoznacznie jest związany z jednym konkretnym przedmiotem.

Z analizy tego przykładu widać, że jednym ze sposobów zapewnienia poprawności przechowywanych danych jest dobry jej projekt. Mamy tutaj także odpowiedź na podstawowy dylemat projektowania, czy koszt bardziej złożonego, w stosunku do jednej tabeli, projektu jest wart poniesienia ze względu na korzyści związane z jednoznacznością zapisanych danych. W omawianej sytuacji odpowiedź wydaje się oczywista. Nie wszystkie problemy daje się jednak rozwiązać poprzez właściwy projekt, czego dowodem jest prezentowana w tej części pierwsza tabela, w której problem był związany z koniecznością wymuszenia i zapewnienia właściwego zapisywania danych – do sposobów rozwiązania tego typu nieprawidłowości wrócimy w dalszych częściach tych zajęć.

6 SYSTEMY ZARZĄDZANIA BAZAMI DANYCH

Dotychczasowe rozważania prowadziliśmy w oderwaniu od technologii, czyli od sposób realizacji. Koncentrowaliśmy się na teorii – a teraz przyszła pora na praktykę. W tym celu zdefiniujemy nowe pojecie:

Systemem Zarządzania Bazami Danych(SZBD) nazywamy specjalistyczne oprogramowanie umożliwiające tworzenie baz danych oraz ich eksploatację.

Wydaje się oczywiste, że tworzenie i działanie baz danych musi być wspierane przez specjalistyczne oprogramowanie, które powinno umożliwiać realizacje pewnych zadań:

- definiowanie obiektów bazy danych,
- manipulowanie danymi,
- generowanie zapytań,
- zapewnienie spójności i integralności danych.

Zadania te brzmią bardzo ogólnie, obejmują jednak większość potrzeb w zakresie tworzenia i eksploatacji baz danych. Dla przybliżenia pojęcia SZBD można podać kilka nazw handlowych, pod jakimi te produkty można spotkać na rynku i w zastosowaniach: MS SQL Server 2008, Oracle, MySQL, Access, DB2 i wiele, wiele innych mniej lub bardziej popularnych.

Jednym z najważniejszych zadań stojących przed SZBD jest zapewnienie spójności i integralności danych, czyli systemowe rozwiązywanie tych problemów, które rozważaliśmy wcześniej. SZBD dostarczają mechani-

zmy służące do zapewnienia spójności i integralności danych, czyli mówiąc innymi słowami, zapewnienia logicznej poprawności danych zapisanych w bazie. Podstawowe mechanizmy, realizujące te zadania to :

- deklaracja typu,
- definicje kluczy,
- reguły poprawności dla kolumny,
- reguły poprawności dla wiersza,
- reguły integralności referencyjnej

I krótkie podsumowanie pierwszego spotkania z pojęciem SZBD:

- Przedstawione wcześniej problemy oraz wiele innych, których nie zdążymy tutaj omówić, są poważnym wyzwaniem dla twórców baz danych.
- Bez rozwiązania tych problemów bazy danych byłyby niewiarygodne.
- W dalszej części wykładu omówimy niektóre sposoby zapewnienia poprawności przechowywanych danych

7 SPÓJNOŚĆ I INTEGRALNOŚĆ DANYCH

W tabelach relacyjnych sa przechowywane dane różnego typu (liczby, teksty, znaki, daty ...). Z przedstawionych wcześniej cech modelu relacyjnego wynika, że każda kolumna w tabeli musi mieć określony typ przechowywanych danych. Deklaracja typu jest pierwszym sposobem zapewnienia poprawności danych – w ujęciu matematycznym jest to określenie dziedziny wartości dla kolumny. SZBD udostępniają zbiór typów, które mogą być wykorzystane w definicji kolumn.

Przykładowe typy danych w SQL Server 2008

Dla danych znakowych

- ***char(n)*** – ciąg n znaków o stałej długości (np. jeżeli kolumna ma określony typ ***char(25)*** a wpiszemy słowo „kot" – to i tak zostanie ono zapisane pomocą 25 znaków – uzupełnione spacjami);
- ***varchar(n)*** – ciąg n znaków o zmiennej długości (np. jeżeli kolumna ma określony typ ***varchar(25)*** i wpiszemy słowo „kot" –zostanie ono zapisane za pomocą 3 znaków)
- ***varchar(max)*** – ciąg znaków o zmiennej długości do 2 GB

W tym miejscu można spróbować odpowiedzieć na następujące pytanie: Skoro typ ***char*** w porównaniu z ***varchar*** wykorzystuje więcej pamięci do zapisywania danych (uzupełnianie spacjami), to jakie korzyści możemy osiągnąć w przypadku wykorzystania typu ***char***.

Dla danych liczbowych – liczby całkowite

- ***tinyint*** – liczba całkowita z zakresu [0 – 255], przechowywana w jednym bajcie;
- ***smallint*** – liczba całkowita z zakresu [–32768 – 32767], przechowywana na dwóch bajtach;
- ***int*** – liczba całkowita z zakresu [–2147483648 – 2147483647], przechowywana na czterech 4 bajtach;
- ***bigint*** – liczba całkowita z zakresu [–9223372036854775808 – 922337203685477580], przechowywana na ośmiu bajtach.

Dla danych liczbowych – liczby z ułamkiem

- ***real, float*** – do zapisywania liczb zmiennopozycyjnych;
- ***decimal, numeric*** – do zapisywania liczb zmiennopozycyjnech o określonej precyzji;
- ***money*** – do zapisywania liczb wyrażających kwoty pieniężne.

Dla danych– daty i czasu

- ***date*** – do zapisywania dat, np. 2009-08-22;
- ***time*** – do zapisywania czasu, np. 19:22:07.2345644;
- ***datetime*** – do zapisywania łącznie daty i czasu, np. 2009-08-22 19:22:07.2345644.

Typy różne

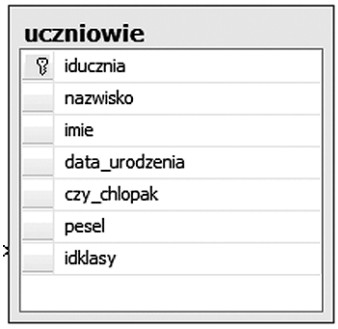
- ***bit*** – do zapisywania wartości logicznych (true, false lub 0,1);
- ***varbinary(n)*** – do zapisywania danych binarnych o długości n bajtów;
- ***varbinary(max)*** – do zapisywania danych binarnych o długości do 2 GB (np. obrazy, dźwięki)
- ***Timestamp*** – specjalny znacznik który automatycznie zmienia swoja wartość przy modyfikacji wiersza

Nie wymieniliśmy wszystkich dostępnych typów danych, widać jednak, że dostępny jest dość duży zbiór typów danych i od decyzji projektanta zależy właściwy ich wybór. Krótkie podsumowanie :

- Każda kolumna w tabeli musi mieć określony typ danych, w jakim będą w tej kolumnie zapisywane dane.
- Decyzja o wyborze odpowiedniego typu danych jest pierwszym etapem zapewnienia spójności danych.
- Wybór typu jest równoznaczny z określeniem dziedziny wartości dla danych zapisywanych w danej kolumnie.

Kolejnym mechanizmem związanym z zapewnieniem spójności i integralności danych są definicje kluczy. W każdej tabeli relacyjnej powinien być zdefiniowany klucz podstawowy – taka definicja zapewnia, że każda wartość w kolumnie klucza podstawowego musi przyjąć inną wartość. W SZBD istnieją mechanizmy nadające kolumnom klucza podstawowego automatycznie unikatowe wartości (autonumeracja), a samo zdefiniowanie kolumny jako klucza podstawowego wymusza jednoznaczność wprowadzanych danych, czyli system nie zezwoli na to, żeby w danej tabeli pojawiły się dwie identyczne wartości w kolumnie klucza podstawowego. Można także wymusić jednoznaczność kolumn, które nie są kluczem podstawowym, czyli definiować tzw. **klucze potencjalne.**

Omawiana wcześniej deklaracja typu określa dziedzinę wartości dla kolumny, ale często jest to dziedzina zbyt szeroka, czyli nie wszystkie wartości z tak określonej dziedziny możemy uznać za poprawne z punktu widzenia zawartości analizowanej kolumny. Dodatkowym utrudnieniem mogą być także zależności logiczne pomiędzy danymi zapisanymi w różnych kolumnach jednego wiersza. Jeżeli jesteśmy w stanie zdefiniować takie zależności, to korzystając z mechanizmu, dostarczanego przez SZBD, definiowania reguł poprawności dla kolumny lub wiersza. Problemy związane z regułami poprawności przeanalizujemy na przykładzie kolumny ***pesel*** z pokazanej na rysunku przykładowej tabeli:



Załóżmy, że dla kolumny ***pesel*** został zdefiniowany typ danych ***char(11)***. Wydaje się, że jest to definicja poprawna, ale rodzi się wiele problemów związanych z zapewnieniem poprawności danych zapisywanych w tej kolumnie, czyli musimy wymusić, żeby każda wartość zapisana w tej kolumnie była poprawnym numerem pesel.

Z punktu widzenia deklaracji typu ***char(11)***, poprawnymi wartościami są wszystkie ciągi znakowe o długości nie przekraczającej 11 znaków i w tym momencie widzimy, jak daleko jeszcze do pełnej poprawności. Gdybyśmy pozostali na takim zdefiniowaniu tej kolumny, to za poprawne dane mogłyby uchodzić nawet tak bezsensowne dane: ‘Ala ma kota’, ‘W45991AS’, ‘brak Pesel’ - każdy z tych przykładowych ciągów znakowych jest poprawny, ponieważ nie przekracza 11 znaków.

Zadanie 1 – ograniczyć definicję typu tak, żeby jako poprawne były traktowane tylko ciągi składające się z dokładnie jedenastu znaków. W celu realizacji tego zadania można skorzystać z mechanizmu definiowania ograniczeń. Ograniczenie dziedziny wartości sprowadza się do zdefiniowania wyrażenia logicznego, które, jeśli jest spełnione, uznaje dane za poprawne. W naszym przypadku takie wyrażenie mogłoby mieć następująca postać:

Ile_znakow(pesel) = 11

Załóżmy, że istnieje funkcja o nazwie ***ile_znakow***, której jako parametr przekazujemy ciąg znaków (w naszym przypadku zawartość kolumny ***pesel***), a w wyniku otrzymujemy liczbę znaków, z których składa się przekazany parametr. Jeżeli funkcja o takiej nazwie nie istnieje w SZBD, to możemy skorzystać z istniejącej o innej nawie ale wykonującej podobne działanie albo utworzyć własną funkcję. Ponieważ w ramach tego wykładu zajmujemy się głównie problemami, to szczegóły realizacji są w tym przypadku mniej istotne. Możemy jedynie zapewnić, że praktycznie w każdym SZBD można taki warunek zdefiniować.

W tym miejscu uznajemy, że potrafimy zdefiniować taka regułę i ... to dopiero początek długiej drogi, ponieważ po tej definicji za poprawne zapisy uznane zostaną następujące ciągi znakowe:

'aswedfcxsdr' – bo zawiera dokładnie 11 znaków,

'a234543234j' – bo też zawiera dokładnie 11 znaków.

Zadanie 2 – zapewnić, że dane w kolumnie ***pesel*** składają się z dokładnie 11 cyfr. W tej sytuacji zadanie sprowadza się do uściślenia naszego wcześniejszego wyrażenia. Posłużymy się w tej sytuacji również hipotetyczna funkcją. Nasze wyrażenie logiczne mogłoby mieć teraz następująca postać :

Ile_znakow(pesel) = 11 and ile_cyfr(pesel)=11

To wyrażenie zapewni, że jako poprawne zostaną uznane tylko ciągi znaków złożone z 11 cyfr i ... byłoby już prawie dobrze, gdyby nie fakt, że w numerze ***pesel*** ostatnia cyfra nie jest cyfrą przypadkową, tylko tak zwaną **sumą kontrolną**. Mechanizmy cyfr kontrolnych są stosowane przez różnego typu identyfikatory (pesel, nip, numer bankowy, numer dowodu osobistego) dla zapobiegania przypadkowym błędom przy wprowadzaniu danych. Jeżeli chcemy traktować bazę poważnie, to powinniśmy także zapewnić sprawdzanie cyfry kontrolnej, a to polega na tym, że według pewnego algorytmu korzystając z pierwszych dziesięciu cyfr numeru Pesel wykonujemy obliczenia, których wynik musi być równy ostatniej cyfrze numeru Pesel i tylko wtedy taki Pesel jest poprawny.

Zadanie 3 – zapewnić sprawdzenie poprawności cyfry kontrolnej, czyli rozbudować nasze wyrażenie sprawdzające poprawność danych i zgodnie z przyjętymi przez nas zasadami odnośnie założenia istnienia takich funkcji, które są w danej chwili potrzebne. Zmodyfikowana postać takiego wyrażenia mogłaby wyglądać następująco:

Ile_znakow(pesel) = 11 and ile_cyfr(pesel)=11

and dziesiata_cyfra(pesel)=cyfra_kontrolna(pesel)

Nasza definicja staje się coraz bardziej złożona, ale po jej określeniu to SZBD będzie sprawdzał i wymuszał poprawność zapisywanych danych. Napracowaliśmy się dużo i ... nagle olśnienie, że przecież pierwsze 6 cyfr w numerze Pesel także nie może być dowolnych ale musi odpowiadać dacie urodzenia, tym bardziej, ze w naszej tabeli obok kolumny ***pesel*** jest także kolumna ***Data_urodzenia*** i niedopuszczalne byłoby zaniechanie synchronizacji miedzy tymi danymi.

Zadanie 4 – zapewnić, żeby numer Pesel był zgodny z zapisaną w tym samym wierszu datą urodzenia. Zgodnie z naszą tradycją rozbudujemy wyrażenie logiczne o kolejny składnik, który będzie odpowiedzialny za sprawdzenie zgodności numeru pesel z datą urodzenia. Po modyfikacji wyrażenie logiczne przyjmie następującą postać :

Ile_znakow(pesel) = 11 and ile_cyfr(pesel)=11

and dziesiata_cyfra(pesel)=cyfra_kontrolna(pesel) and

data_z_pesel(pesel)=data_urodzenia

W tym miejscu chcielibyśmy zwrócić uwagę, że po raz pierwszy wyrażenie odwołuje się do różnych kolumn tabeli (oczywiście ma to sens w obrębie jednego konkretnego wiersza). Moglibyśmy pewnie już spocząć na laurach, gdyby nie jeszcze jedna wiadomość mówiąca o tym, że dziesiąta cyfra numeru Pesel także nie jest przypadkowa, gdyż jest to cyfra określająca płeć (parzysta kobiety, nieparzysta mężczyźni) a pamiętamy, że w naszej tabeli jest kolumna o dziwnej nazwie ***czy_chlopa***, w której zapisujemy dane określające płeć.

Zadanie 5 – Zapewnić sprawdzenie poprawności oznaczenia płci w numerze Pesel. W tej sytuacji została nam, mamy nadzieje ostatnia modyfikacji naszej reguły poprawności do następującej postaci :

Ile_znakow(pesel) = 11 and ile_cyfr(pesel)=11

and dziesiata_cyfra(pesel)=cyfra_kontrolna(pesel) and

data_z_pesel(pesel)=data_urodzenia and

sprawdz_plec(czy_chlopak, pesel)=1

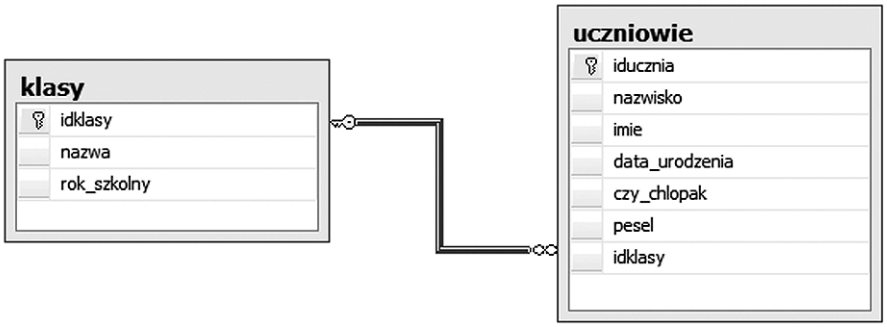
W tym przypadku założyliśmy, że istnieje funkcja o nazwie ***sprawdz_plec***, która przyjmuje dwa parametry (***czy_chlopak*** i ***pesel***) i jeżeli zwróci wartość 1, to znaczy, że płeć w numerze ***pesel*** jest prawidłowa.

Przyznać trzeba, że rozważany problem został rozwiązany w zbyt skomplikowany sposób. Równie dobrze można było założyć istnienie jednej funkcji, która realizuje wszystkie nasze zadania – funkcja o nazwie ***sprawdz_pesel***, której przekażemy trzy parametry (***pesel***, ***data-urodzenia***, ***czy_chlopak***) i jeżeli taka funkcja zwróci wartość 1, to uznamy dane zapisane w wierszu za poprawne:

sprawdz_pesel(pesel, data_urodzenia, czy_chlopak) = 1

Niezależnie od ostatecznej postaci wyrażenia wymuszającego poprawność, w trakcie omawiania problemów związanych z poprawnością numeru Pesel, wykazaliśmy, że zagadnienia zapewnienia spójności i integralności danych są złożone, ale jednocześnie bardzo istotne, gdyż tylko wymuszenie poprawności danych może gwarantować przydatność całej bazy danych.

Na koniec jeszcze krótkie rozważania o problemie integralności danych zapisanych w różnych tabelach. W sytuacji, gdy pewne zależności między danymi obejmują dane zapisane w różnych tabelach, to mówimy o **integralności referencyjnej.** Najczęściej problemy integralności referencyjnej dotyczą pary: klucza podstawowego – klucza obcego. Zastanówmy się chwilę nad znanym nam już układem tabel :



Spróbujmy odpowiedzieć na poniższe pytania :

1. Czy można usunąć z tabeli ***klasy*** wiersz, którego wartość klucza podstawowego (***idklasy***) jest już zapisana w tabeli ***uczniowie***, czyli czy można usunąć z danych o klasie, gdy jest choć jeden uczeń dowiązany do tej klasy?
2. Czy możemy, zapisując dane ucznia w tabeli ***uczniowie***, wprowadzić w kolumnie ***idklasy*** wartość, która nie występuje jako klucz podstawowy w tabeli ***klasy***?
3. Czy możemy zmienić wartość zapisaną w kolumnie ***idklasy*** tabeli ***klasy***, gdy taka wartość została już przypisana do ucznia?

Na każde z tych pytań powinniśmy odpowiedzieć przecząco, gdyż dopuszczenie do takich działań mogłoby spowodować, że klucz obcy nie miałby swojego odpowiednika w tabeli, gdzie występuje on jako klucz podstawowy, czyli uczeń byłby dowiązany do klasy, która w bazie danych nie istnieje.

Dobrą wiadomością jest to, że SZBD dostarczają mechanizmów, dzięki którym możemy zapewnić poprawność odwołań klucza obcego do podstawowego i żadna z opisanych operacji nie zostałaby wykonana.

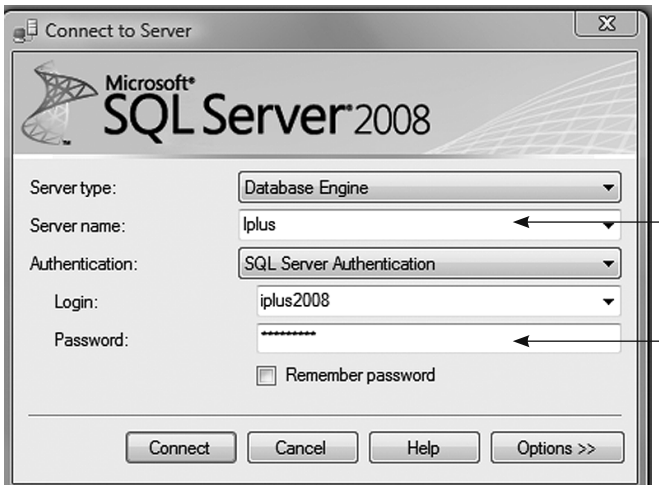
WARSZTATY

W ramach warsztatów będzie wykorzystywany System Zarządzania Bazami Danych MS SQL Server 2008 Express. Poniżej zamieszczona jest instrukcja, jak ściągnąć z Internetu i zainstalować to oprogramowanie.

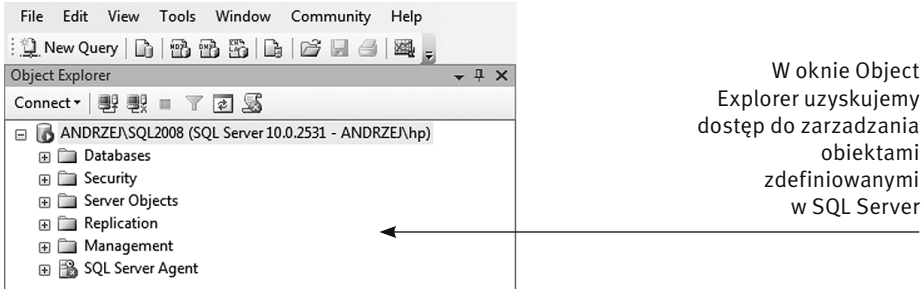
Ćwiczenie 1. Zapoznanie się ze środowiskiem MS SQL Server 2008

Wspólnie z prowadzącym warsztaty rozpoznajemy środowisko SZBD MS SQL Server 2008. Korzystanie z tego systemu umożliwia specjalne oprogramowanie SQL Server Menagement Studio.

1. Uruchamiamy SQL Server Menagement Studio (lokalizację programu poda prowadzący warsztaty).
2. Po uruchomieniu programu przechodzimy do logowania się do SQL Servera, na ekranie pojawi się okienko do logowania, w którym wpisujemy w pola wartości takie, jak podane na rysunku.



3. Po poprawnym wpisaniu powyższych wartości uruchomione zostanie oprogramowanie i pojawia się okna SQL Server Menagement Studio.

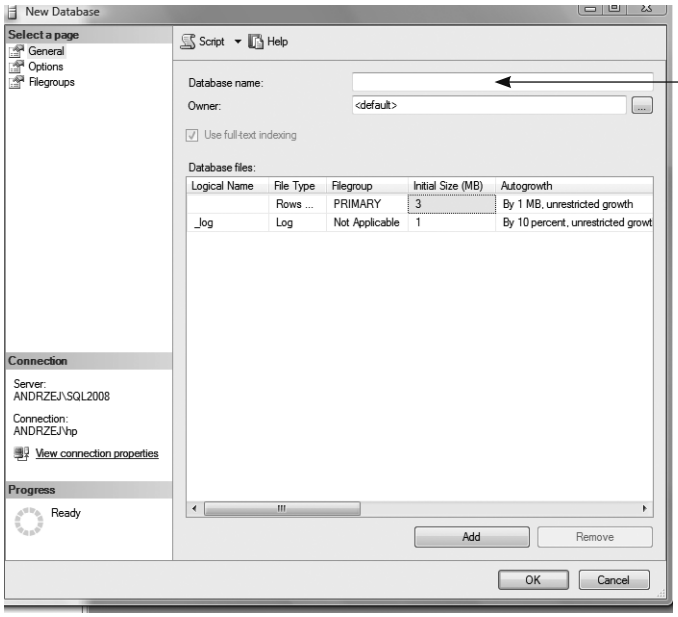


4. Wspólnie z prowadzącym warsztaty poznajemy wybrane elementy środowiska SQL Servera.

Ćwiczenie 2. Zainicjowanie pierwszej bazy danych

W trakcie tego ćwiczenia każdy uczestnik warsztatów utworzy swoją pierwszą bazę danych. W tym celu wykonujemy następujące czynności:

1. W oknie Object Explorer wybieramy folder Databases.
2. Po kliknięciu prawym klawiszem myszy pojawia się menu, z którego wybieramy opcję New Database.
3. Po wybraniu opcji New Databases pojawia się okno definiowania bazy danych.



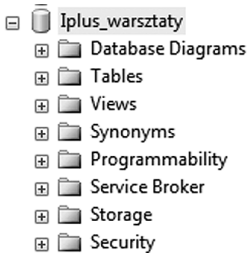
W okno Data name wpisujemy nazwę którą chcemy

Przez proces tworzenia nowej bazy danych poprowadzi prowadzący warsztaty. Jak widać, zainicjowanie tworzenia nowej bazy danych nie jest zbyt złożone (w standardowej wersji). Po utworzeniu nowej bazy możemy przystąpić do definiowania tabel.

Ćwiczenie 3. Utworzenie pierwszej tabeli

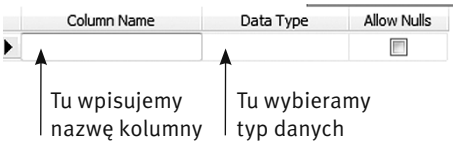
Po utworzeniu nowej bazy danych jest ona pusta, czyli nie zawiera tabel (za wyjątkiem tabel systemowych). Utworzymy teraz nową tabelę.

1. W oknie Object Explorer rozwijamy folder Databases.
2. Po pojawieniu się listy dostępnych baz danych, wybieramy bazę o nazwie ***Iplus_warsztaty*** i rozwijamy folder – pojawi się lista elementów, które można definiować.

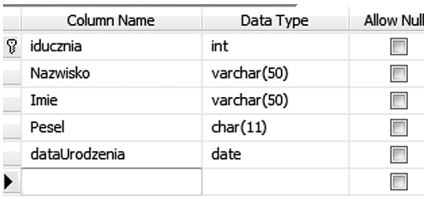


Prowadzący wyjaśni krótko widoczne elementy.

3. Po kliknięciu prawym klawiszem myszy na folderze Tables, pojawi się menu, z którego wybieramy opcję New Table
4. Pojawi się okno wspomagające proces definiowania tabeli. Utworzenie tabeli polega głównie na zdefiniowaniu poszczególnych kolumn.



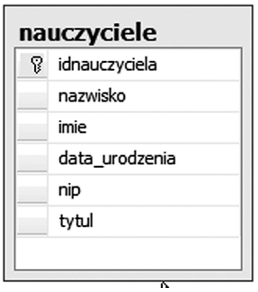
5. Po zakończeniu tworzenia tabeli, okno będzie wyglądało jak poniżej:



6. Po zakończeniu definiowania kolumn zamykamy okno i nadajemy nazwę tabeli.

Ćwiczenie 4. Utworzenie tabeli na podstawie projektu

W ramach tego ćwiczenia każdy uczestnik samodzielnie zdefiniuje tabelę według poniższego schematu:



Zadanie polega na utworzeniu tabeli według pokazanego schematu. W ramach zadania należy:

1. Zdefiniować poszczególne kolumny.
2. Określić typ danych dla definiowanych kolumn.
3. Zdefiniować klucz podstawowy.
4. Po zakończeniu definiowania – zapisać nową tabelę (jako nazwę tabeli podajemy swoje imię i nazwisko).
5. Przedyskutować z prowadzącym poprawność definicji.
6. Po wykonaniu zadania usuwamy tabelę z bazy danych.

Ćwiczenie 5. Projektowanie i definiowanie tabeli

Każdy uczestnik samodzielnie projektuje i definiuje tabelę. Zadanie należy przeprowadzić według następujących kroków:

1. Wymyślić sobie przeznaczenie dla definiowanej tabeli.
2. Ustalić, jakie kolumny powinna zawierać tabela – dla celów następnego ćwiczenia, w tabeli powinna być przynajmniej jedna kolumna typu ***date***.
3. Zdefiniować w bazie danych zaprojektowaną tabelę.

4. Ustalić i zdefiniować klucz podstawowy.
5. Zapisać tabelę (nazwa tabeli to nazwisko i imię uczestnika).
6. Przedyskutować z prowadzącym poprawność definicji.
7. Wpisać do utworzonej tabeli kilka przykładowych wierszy.

Czy proces projektowania własnej tabeli okazał się być łatwy?

Wprowadzanie danych do tabeli realizujemy następująco :

- W folderze Tables wybieramy naszą tabelę.
- Po kliknięciu prawym klawiszem myszy wybieramy opcje Edit Top 200 Rows.
- Pojawia się okienko do wprowadzania danych.

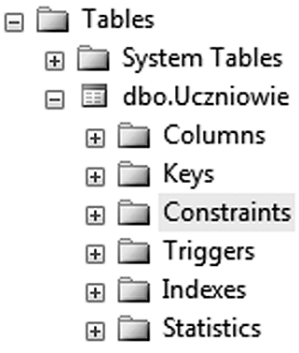
	iducznia	Nazwisko	Imie	Pesel	dataUrodzenia
*	NULL	NULL	NULL	NULL	NULL

Ćwiczenie 6. Definiowanie reguł poprawności

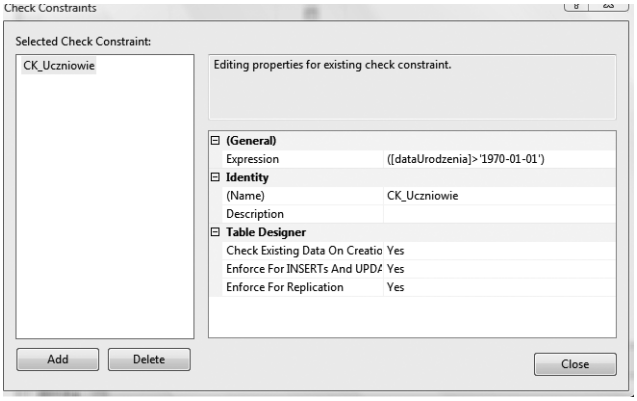
W tabeli utworzonej w poprzednim ćwiczeniu zdefiniujemy regułę poprawności dla kolumny, która ma określony typ danych jako ***date***. Dla celów zademonstrowania działania reguł poprawności zdefiniujemy wymaganie, aby zapisywane daty były większe od 01-01-1970.

Zadanie realizujemy według następującego schematu :

1. Rozwijamy folder Tables.
2. Wybieramy własną (zdefiniowana w poprzednim ćwiczeniu tabelę).
3. Rozwijamy folder tej tabeli – poniżej przykładowa zawartość takiego folderu.



4. Wybieramy folder Constraints i po kliknięciu prawym klawiszem myszy wybieramy opcję New Constraint – pojawia się okienko definiowania reguły:



5. W polu Expression wpisujemy wyrażenie logiczne według następującej formuły :

Nazwa_kolumny ›'1970-01-01'

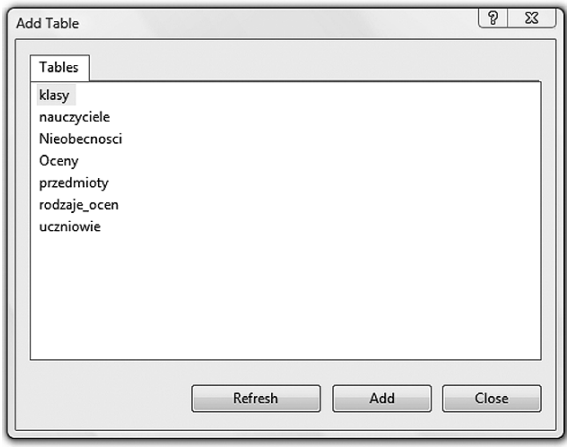
W miejsce pola ***Nazwa_kolumny*** wpisujemy faktyczną nazwę kolumny w tabeli, dla której definiujemy regułę.

6. Po zapisaniu wyrażenia naciskamy przycisk Close.
7. Po poprawnym wykonaniu zadania – określona reguła zaczyna działać.
8. Sprawdzić działanie reguły poprzez próby wprowadzania danych, które tej reguły nie spełniają.

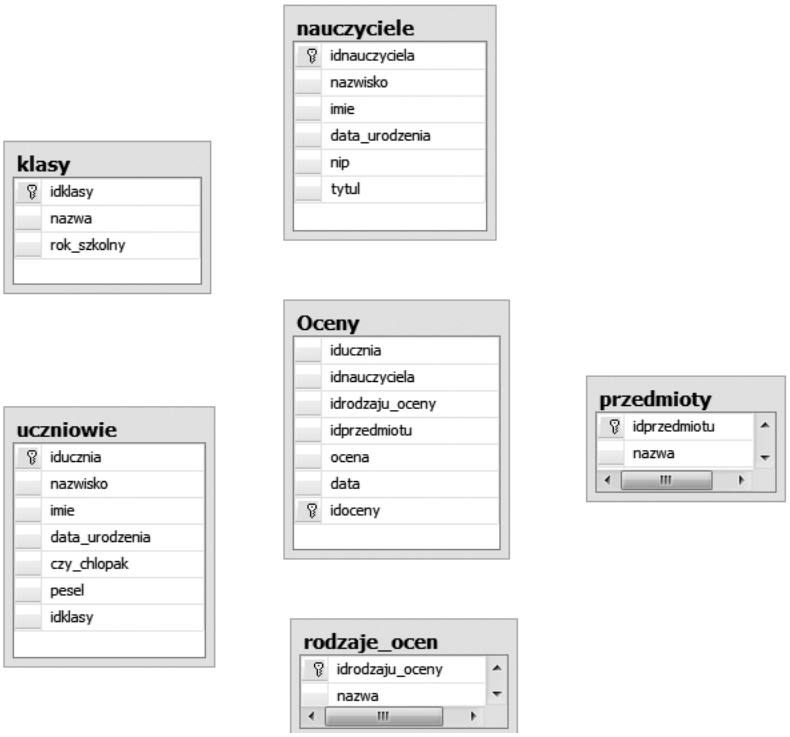
Ćwiczenie 7. Definiowanie reguł integralności referencyjnej – tworzenie diagramu bazy danych

W ramach tego ćwiczenia zdefiniujemy regułę integralności referencyjnej oraz utworzymy diagram bazy danych.

1. Wybieramy bazę danych o nazwie *_wszechnica*.
2. Rozwijamy folder tej bazy danych.
3. Wybieramy folder Database Diagrams i po kliknięciu prawym klawiszem myszy wybieramy opcję New Database Diagram – pojawi się okno z listą tabel zdefiniowanych w bazie danych:

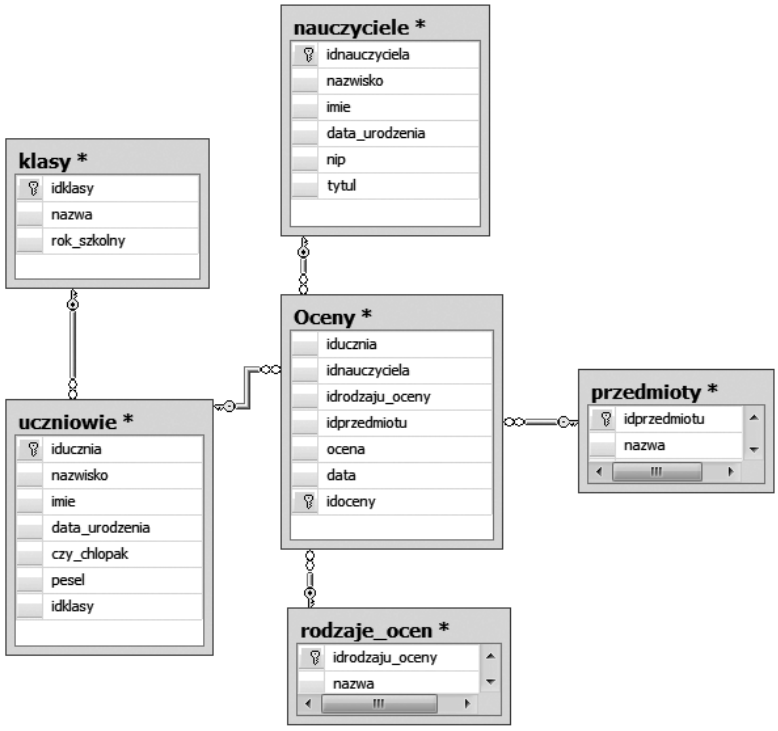


4. Zaznaczamy wszystkie tabele i naciskamy przycisk Add – pojawi się obszar diagramu. Widoczne tabele odpowiadają projektowanej w trakcie wykładu bazie „Elektroniczny dziennik ocen”.



5. Widoczne tabele są zdefiniowane w bazie danych bez określenia powiązań pomiędzy kluczami obcymi i podstawowymi.

6. W ramach definiowania diagramu należy przeciągnąć myszą klucz podstawowy do odpowiadającego mu klucza obcego – po wykonaniu takiej operacji na diagramie pojawi się powiązanie. Po wykonaniu zadania diagram powinien mieć postać, jak poniżej:



7. Po wykonaniu zadania zamykamy okno diagramu i nadajemy mu nazwę (nazwisko i imię wykonującego).
8. Poprawne zapisanie dokonanych zmian powoduje uruchomienie mechanizmu więzów integralności referencyjnej.
9. Sprawdzamy działanie mechanizmu poprzez próby wykonania niedozwolonych operacji.
10. Wyjaśniamy pytania i wątpliwości z prowadzącym warsztaty.

Ćwiczenie 8. Omówienie procesu instalacji MS SQL Server 2008

W ramach ostatniego ćwiczenia zostanie omówiony proces instalacji SQL Server 2008, ponieważ oprogramowanie to jest dostępne za darmo, to warto wiedzieć jak poradzić sobie z jego instalacją.

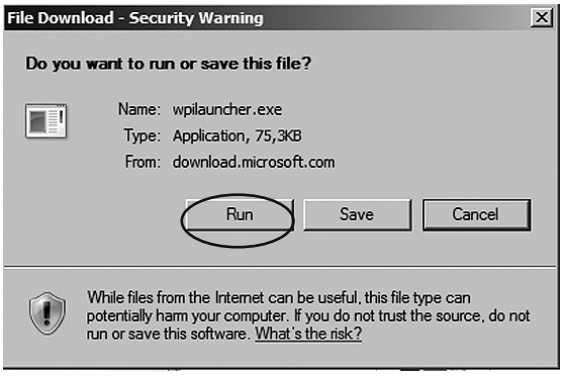
Wprowadzenie

Obok komercyjnych wersji, SQL Server występuje także w darmowym wariancie zwanym SQL Server 2008 Express. Jego instalacja jest jednak, podobnie jak pozostałych wersji, dość złożona i można popełnić przy niej wiele błędów. Aby się tego ustrzec i móc jak najszybciej zacząć wykorzystywać możliwości oferowane przez SQL Server 2008, podamy w miarę prosty i łatwy sposób zainstalowania tego systemu.

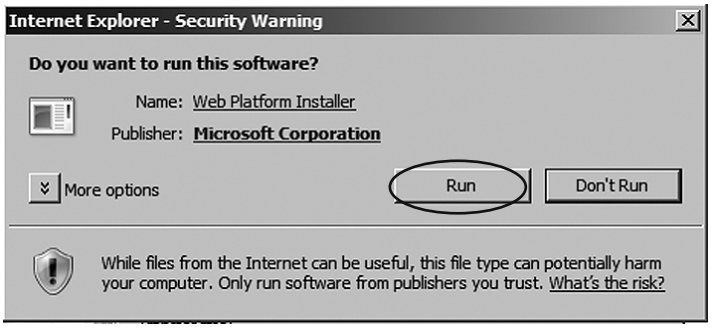
Przygotowanie do instalacji i instalacja

Aby zainstalować SQL Server 2008 Express wraz z narzędziami do zarządzania (SQL Server Management Studio) w możliwie prosty sposób, można ściągnąć ze strony WWW producenta pakiet Microsoft Web Platform Installer – http://www.microsoft.com/web/downloads/platform.aspx .

Jest to bardzo wygodne narzędzie nie tylko do instalowania systemu SQL Server 2008 Express, ale także wielu innych komponentów stosowanych do budowania rozwiązań opartych na .NET Framework 3.5 SP1. Zacznijmy więc od pobrania pakietu Microsoft Web Platform Installer. Można od razu wybrać uruchomienie pakietu bezpośrednio po ściągnięciu:

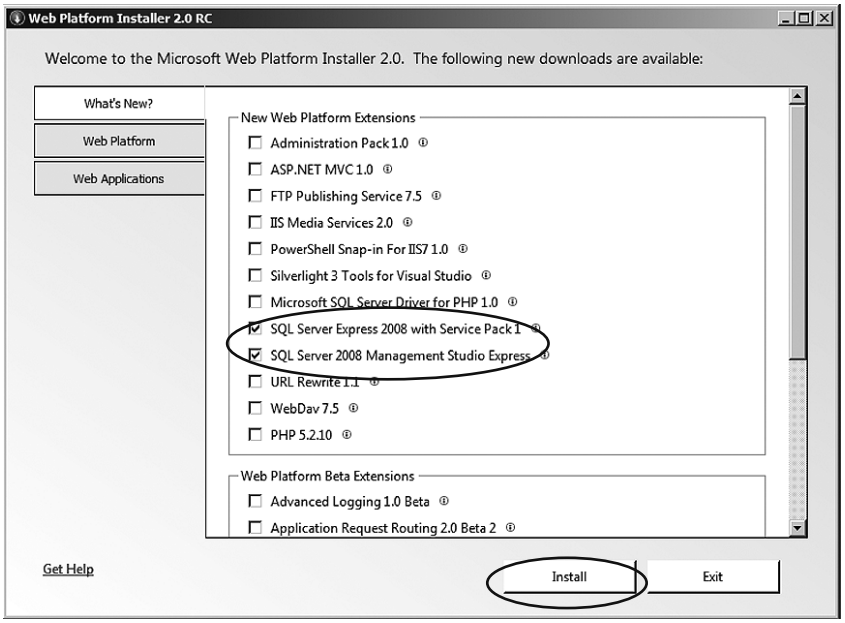


Kolejnym krokiem jest wyrażenie zgody na instalację programu pochodzącego od określonego dostawcy (Microsoft).

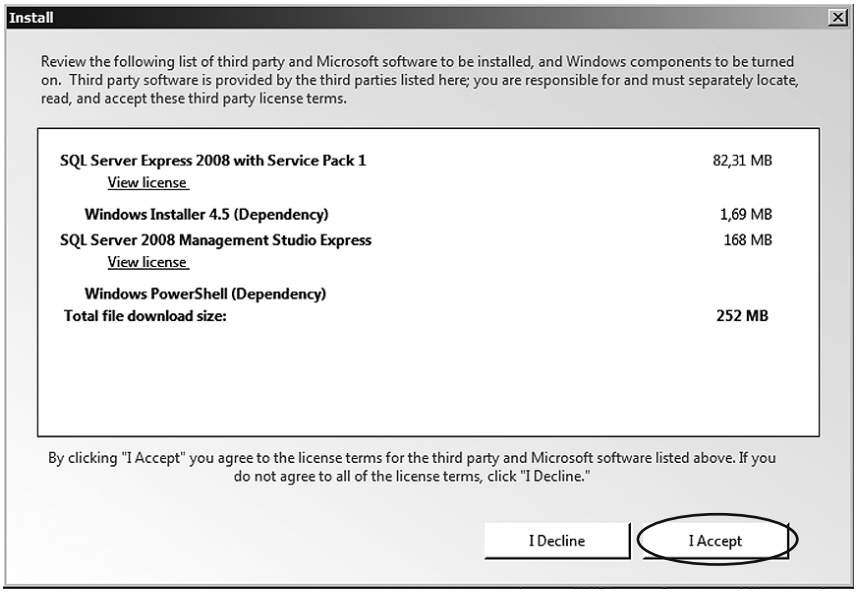


Po chwili pakiet zostaje zainstalowany i rozpoczyna się sprawdzanie najnowszych dostępnych wersji oprogramowania, które można instalować za jego pomocą. Ten proces będzie wykonywany przy każdym kolejnym uruchomieniu tego narzędzia i dzięki temu gwarantuje nam, że instalujemy zawsze aktualne oprogramowanie.

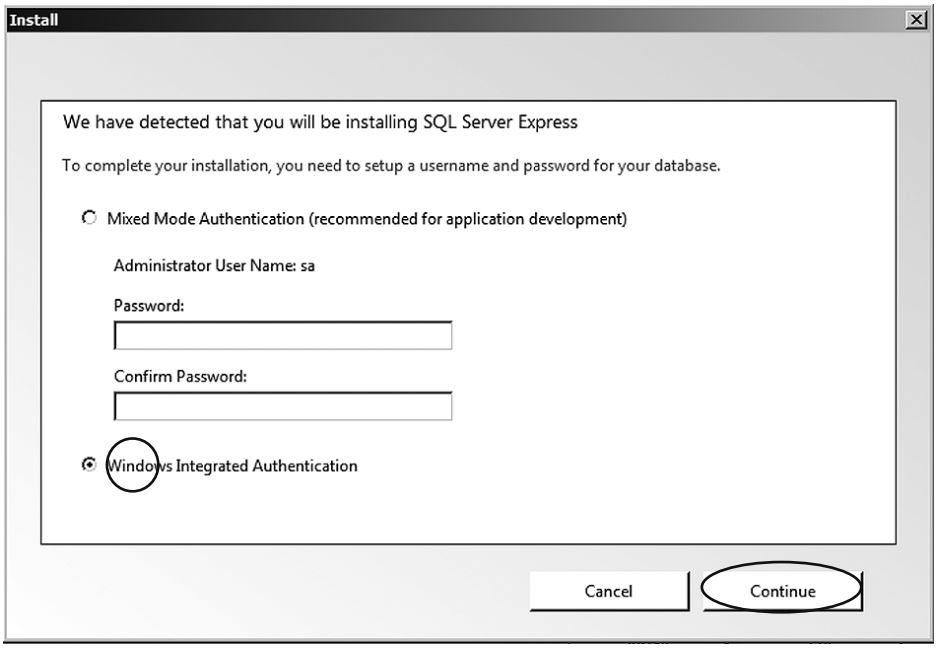
Gdy tylko ten proces się zakończy, będziemy mogli przejrzeć listę dostępnych komponentów oraz wybrać te, które mają zostać zainstalowane. Dla naszych potrzeb są to: SQL Server Express 2008 oraz SQL Server Management Studio Express:



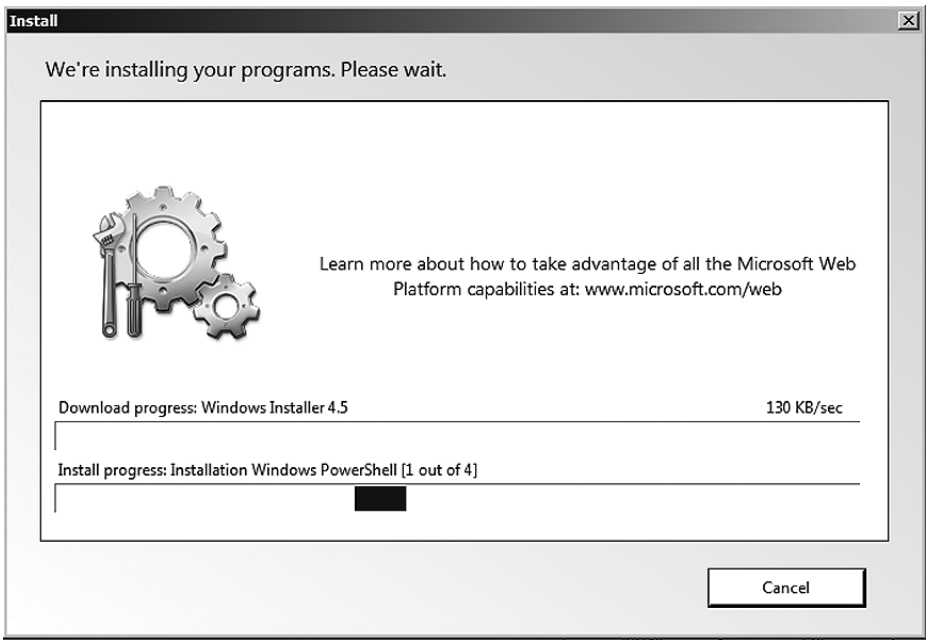
Po zaakceptowaniu tego wyboru przez kliknięcie przycisku Install zostaniemy poinformowani, jakie komponenty zostaną pobrane z Internetu i jaka jest ich wielkość. Nie należy się dziwić, że lista ta może zawierać więcej opcji niż zaznaczyliśmy – będą to tzw. zależności czyli komponenty, bez których działanie wybranych przez nas opcji nie jest możliwe. Zawartość tej listy może się zmieniać w zależności od konfiguracji komputera i zainstalowanych wcześniej programów:



Zgadzając się na warunki licencji przechodzimy do dalszego etapu instalacji – skonfigurowania trybu uwierzytelniania dla SQL Servera. Mamy do wyboru dwie opcje: Mixed Mode oraz Windows Integrated Security. Wybierzmy tę drugą – nasz serwer będzie przy uwierzytelnianiu polegał na mechanizmach systemu Windows – jeśli ktoś zaloguje się do komputera i nadamy mu uprawnienia do korzystania z serwera baz danych, to uzyska taki dostęp. Domyślnie taki dostęp będzie miał użytkownik, na którego koncie instalujemy oprogramowanie. W trybie Mixed Mode możliwe będzie dodatkowo tworzenie loginów i haseł (na poziomie SQL Server), które także będą umożliwiały uzyskiwanie dostępu do baz danych.

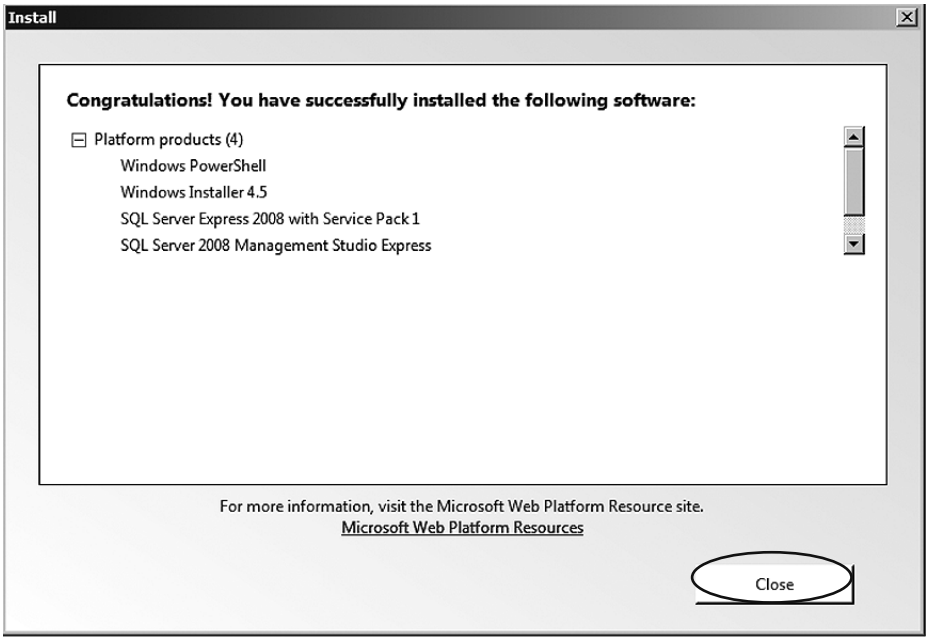


Na tym kończy się konfigurowanie naszej instancji SQL Servera. Web Platform Installer rozpoczyna teraz pobieranie poszczególnych komponentów z Internetu i instalowanie ich.

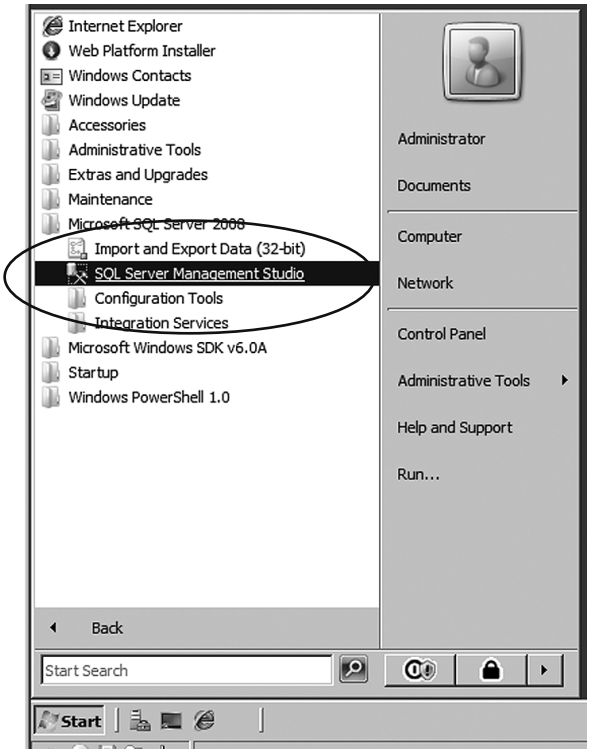


W trakcie tego procesu może się zdarzyć, że któryś z instalowanych komponentów będzie wymagał restartu systemu. Należy się wówczas na to zgodzić, a po restarcie i zalogowaniu się proces instalacji będzie kontynuowany automatycznie.

Instalowane będą kolejne komponenty, aż do ostatniego. Po zainstalowaniu wszystkich zależności oraz wybranych przez nas komponentów wyświetlone zostanie podsumowanie instalacji. Po zapoznaniu się z nim możemy zakończyć proces instalacji klikając przycisk Close.

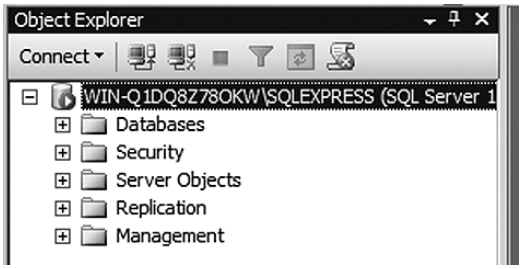


Teraz możemy zweryfikować poprawność instalacji praktycznie. Sama usługa SQL Servera (silnik baz danych) jest skonfigurowana tak, aby uruchamiała się automatycznie wraz ze startem systemu operacyjnego. W menu Start powinna pojawić się grupa Microsoft SQL Server 2008, zawierająca m.in. skrót do narzędzia SQL Server management Studio. Spróbujmy je uruchomić.



Jeśli wszystko poszło dobrze, powinno uruchomić się właśnie zainstalowane narzędzie i zapytać o serwer baz danych, z którym ma się połączyć, przy czym domyślnie wskazany będzie nasz zainstalowany SQL Server 2008 Express oraz wybrany będzie tryb uwierzytelniania Windows Authentication.

Po nawiązaniu połączenia w oknie Object Explorera powinniśmy zobaczyć drzewo, za którego pomocą można zarządzać bazami danych jak i samym SQL Serverem.



Problemy przy instalacji

Może się zdarzyć, że instalacja przebiegnie tak łatwo, jak przedstawiliśmy to powyżej. Typowym problem, na który możemy natrafić podczas instalacji jest wykrycie niezgodności któregoś z instalowanych komponentów z już zainstalowanymi na komputerze. W takim przypadku zwykle wystarcza odinstalowanie istniejącej (starszej) wersji i ponowne uruchomienie programu Web Platform Installer. Należałoby także wspomnieć, że SQL Server Express 2008 nie może być zainstalowany razem z istniejącym SQL Server Express 2005. W takim przypadku trzeba ten wcześniejszy program odinstalować przed rozpoczęciem instalacji wersji 2008.

W przypadku pojawienia się innych błędów przy instalacji nie należy wpadać w panikę. Lepiej poszukać w Internecie informacji, czy ktoś się już spotkał z takim błędem i czy ewentualnie znalazł jego rozwiązanie. Zwykle odnalezienie takich wskazówek nie jest zbyt trudne ani czasochłonne, więc w większości przypadków instalacja kończy się sukcesem. Czego i Wam życzymy :)

W projekcie **Informatyka +**, poza wykładami i warsztatami, przewidziano następujące działania:

- 24-godzinne kursy dla uczniów w ramach modułów tematycznych
- 24-godzinne kursy metodyczne dla nauczycieli, przygotowujące do pracy z uczniem zdolnym
- nagrania 60 wykładów informatycznych, prowadzonych przez wybitnych specjalistów i nauczycieli akademickich
- konkursy dla uczniów, trzy w ciągu roku
- udział uczniów w pracach kół naukowych
- udział uczniów w konferencjach naukowych
- obozy wypoczynkowo-naukowe.

Szczegółowe informacje znajdują się na stronie projektu

www.informatykaplus.edu.pl