

INFORMATYKA

– MÓJ SPOSÓB NA POZNANIE I OPISANIE ŚWIATA

PROGRAM NAUCZANIA INFORMATYKI Z ELEMENTAMI
PRZEDMIOTÓW MATEMATYCZNO-PRZYRODNICZYCH

Informatyka – poziom rozszerzony

Język SQL, czyli jak rozmawiać z bazą danych

Andrzej Ptasznik

$$\sum_{i=1}^n$$

Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



WARSZAWSKA
WYŻSZA SZKOŁA
INFORMATYKI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Tytuł: **Język SQL, czyli jak rozmawiać z bazą danych**

Autor: **Andrzej Ptasznik**

Redaktor merytoryczny: **prof. dr hab. Maciej M. Sysło**

Materiał dydaktyczny opracowany w ramach projektu edukacyjnego
Informatyka – mój sposób na poznanie i opisanie świata.
Program nauczania informatyki z elementami przedmiotów
matematyczno-przyrodniczych

www.info-plus.wwsi.edu.pl

infoplus@wwsi.edu.pl

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

Projekt graficzny: *Marzena Kamasa*

Warszawa 2013

Copyright © Warszawska Wyższa Szkoła Informatyki 2013
Publikacja nie jest przeznaczona do sprzedaży

Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



WARSZAWSKA
WYŻSZA SZKOŁA
INFORMATYKI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY





SCENARIUSZ TEMATYCZNY

JĘZYK SQL, CZYLI JAK ROZMAWIAĆ Z BAZĄ DANYCH

→ INFORMATYKA – POZIOM POZIOM ROZSZERZONY

**OPRACOWANY W RAMACH PROJEKTU:
INFORMATYKA – MÓJ SPOSÓB NA POZNANIE I OPISANIE ŚWIATA.
PROGRAM NAUCZANIA INFORMATYKI
Z ELEMENTAMI PRZEDMIOTÓW MATEMATYCZNO-PRZYRODNICZYCH**

Streszczenie

Głównym celem scenariusza jest zapoznanie się z poleceniem SELECT, przy pomocy którego w języku SQL realizowane są zapytania. Zanim przystąpimy do omawiania polecenia SELECT, wyjaśnimy istotę zapytań do relacyjnej bazy danych.

Dane w bazie relacyjnej zapisane są w postaci tabel i wynik zapytania też jest dwuwymiarową tabelą, chociaż mogą być od tego odstępstwa związane z obsługą w środowisku SQL Server 2012 danych typu XML (ang. *Extensible Markup Language*). Akronim XML można przetłumaczyć jako Rozszerzalny Język Znaczników. Zagadnienia związane z danymi zapisanymi w postaci dokumentów XML będą szczegółowo omawiane w modelu poświęconym standardom wymiany danych.

Istota zapytań polega na wybraniu z całej bazy danych takiej tabeli, która spełnia wymagania realizowanego zadania. Realizacja zapytań opiera się na trzech podstawowych operacjach wykonywanych na modelu relacyjnym:

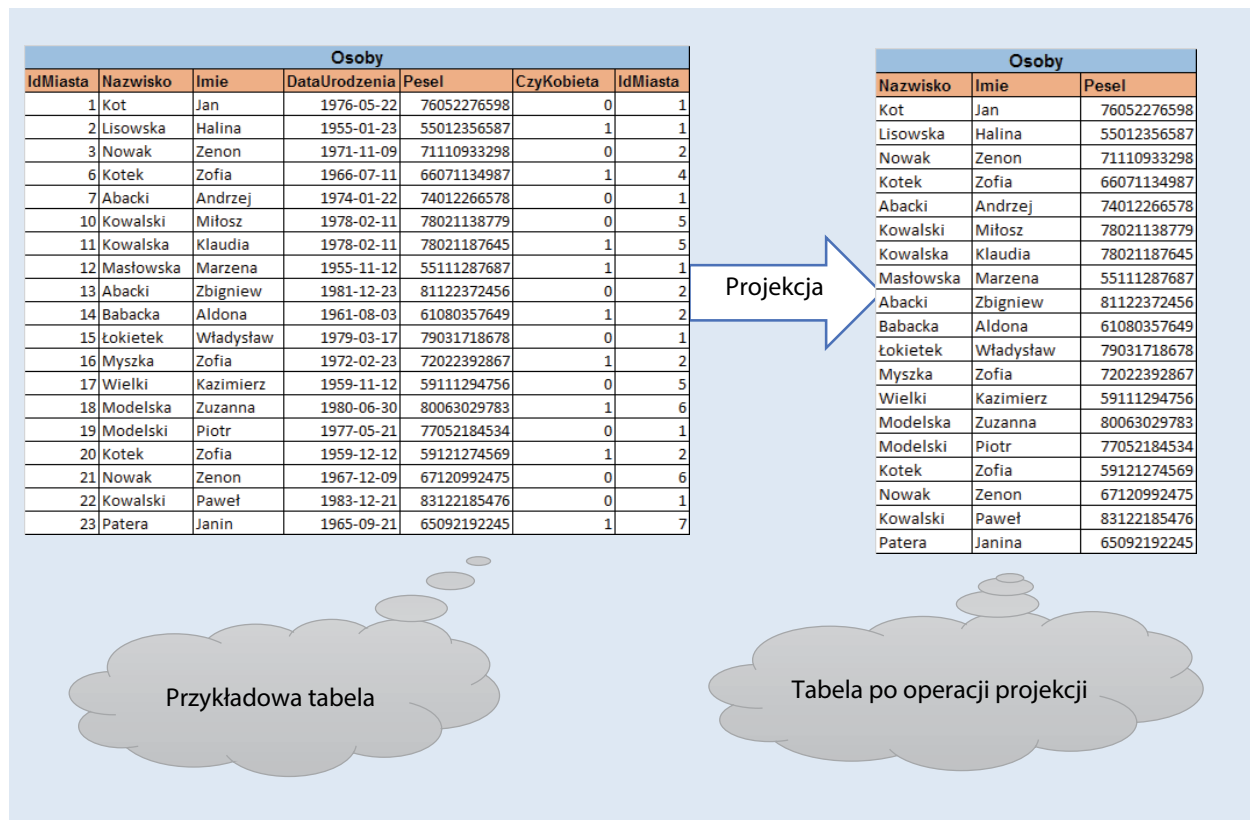
- projekcji (zwana także rzutowaniem),
- selekcji,
- łączenia.

Praktycznie każde zapytanie, realizowane za pomocą polecenia języka SQL, jest wypadkową tych trzech operacji. Wyjaśnimy teraz, na czym polegają te operacje.

Operacja projekcji

Operacja projekcji polega na wyborze podzbioru kolumn ze zbioru wszystkich kolumn dostępnych w danej tabeli, a więc wynikiem projekcji dla danej tabeli jest inna tabela, w której dostępne są tylko niektóre kolumny z tabeli wyjściowej. Przykład operacji projekcji pokazano na rysunku 1.



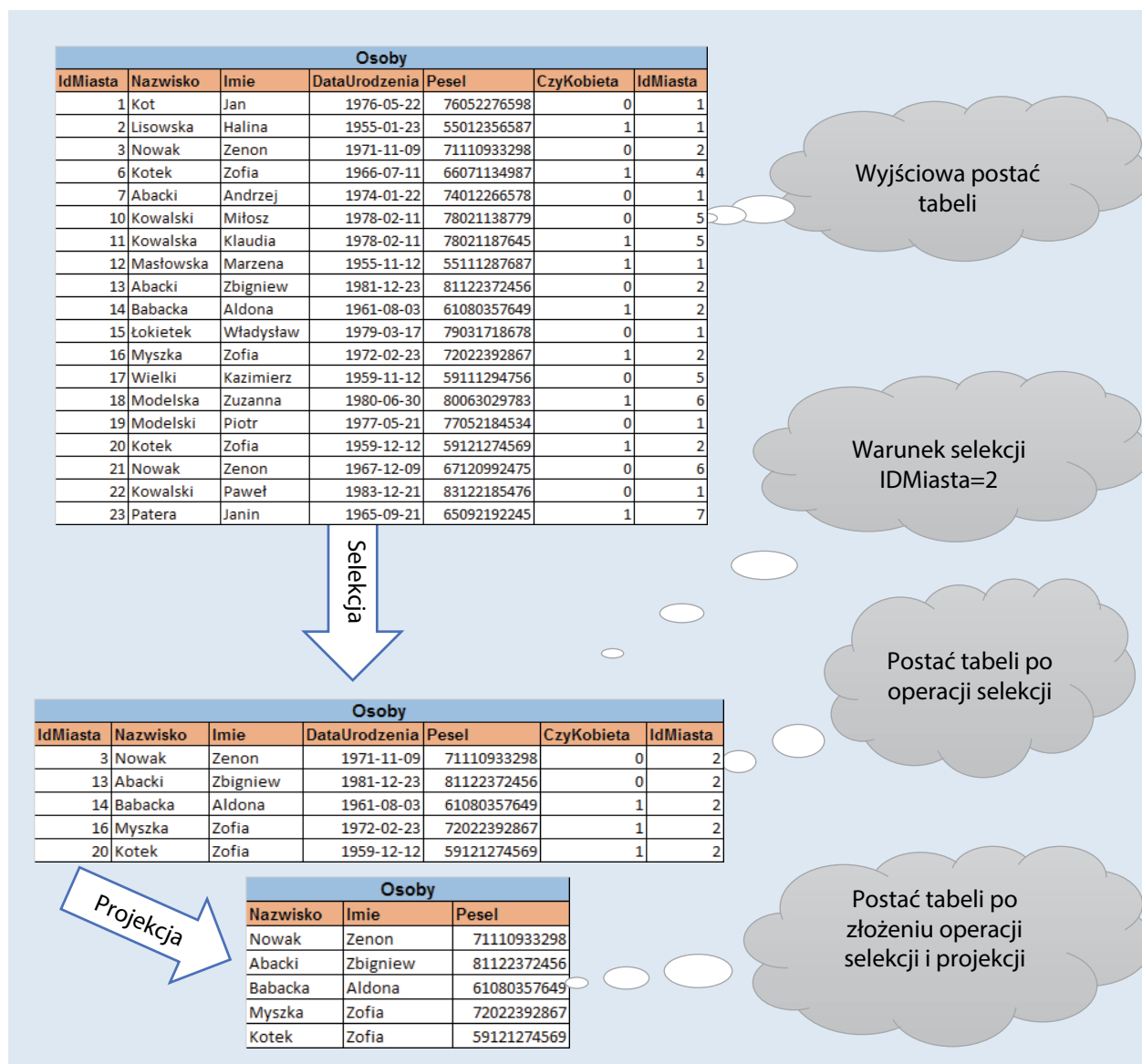


Rysunek 1. Operacja projekcji.

Operacja projekcji zmniejsza rozmiar tabeli wyjściowej poprzez wyeliminowanie kolumn, które w danym momencie uznajemy za nieistotne, natomiast liczba wierszy pozostaje bez zmian.

Operacja selekcji

Operacja selekcji polega na wyborze podzbioru wierszy ze zbioru wszystkich wierszy dostępnych w danej tabeli. Podstawą operacji selekcji jest wyrażenie logiczne, które decyduje, czy dany wiersz powinien się znaleźć w zbiorze wynikowym. Operacja selekcji również zmniejsza rozmiar tabeli wyjściowej, ale poprzez wyeliminowanie wierszy, dla których wyrażenie logiczne nie jest prawdziwe. Przykład operacji selekcji pokazano i omówiono na rysunku 2.

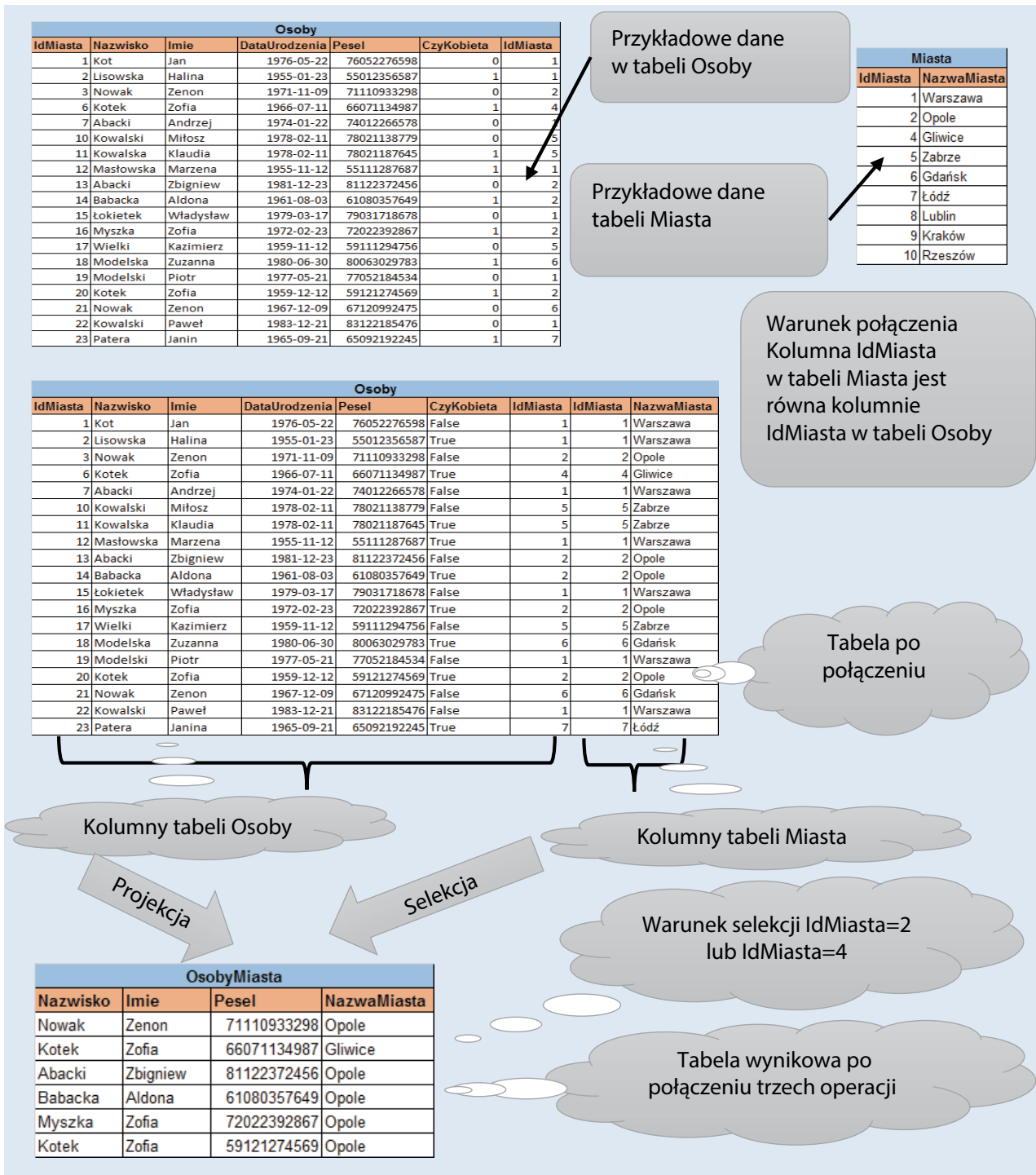


Rysunek 2. Operacja selekcji i złożenie operacji selekcji z projekcją.

Możliwe jest połączenie operacji projekcji i selekcji, co pokazano na rysunku 2, a dzięki takiemu połączeniu jesteśmy w stanie wybrać dowolny podzbiór danych z tabeli wyjściowej.

Operacja łączenia

Operacja łączenia polega na dołączeniu do tabeli wyjściowej kolumn z innej tabeli na podstawie odpowiedniego wyrażenia logicznego. W relacyjnych bazach danych operację łączenia wykonujemy (najczęściej) w oparciu o klucz obcy w tabeli wyjściowej i klucz podstawowy w tabeli dołączanej do tabeli wyjściowej. Operacja łączenia sprowadza dane zapisane w wielu tabelach do postaci jednej tabeli i od tego momentu wszystkie operacje poprawne dla jednej tabeli mogą być wykonywane na tabeli połączonej. Na rysunku 3 pokazano i omówiono operację łączenia oraz dodatkowo operację projekcji i selekcji dla połączonej tabeli:



Rysunek 3. Operacja łączenia oraz złożenie wszystkich operacji.

Operacja łączenia sprowadza dane zapisane w wielu tabelach do postaci jednej tabeli i od tego momentu wszystkie operacje poprawne dla jednej tabeli mogą być wykonywane na tabeli połączonej. Możliwe jest złączenie operacji projekcji, selekcji i łączenia, dzięki czemu możemy wybrać dowolny podzbiór z danych zapisanych w wielu różnych tabelach.



Podstawowa składnia polecenia SELECT

Język SQL, jest językiem deklaratywnym i formułując zapytanie określać będziemy jedynie to, co chcemy osiągnąć, bez konieczności znajomości szczegółów realizacji. O tym, w jaki sposób zapytanie będzie realizowane decydują mechanizmy wewnętrzne Systemu Zarządzania Relacyjnymi Bazami Danych.

UWAGA: Wynikiem zapytania do relacyjnej bazy danych jest dwuwymiarowa tabela.

Język SQL udostępnia jedno polecenie do realizacji zapytań SELECT. Polecenie to ma złożoną składnię i dzięki temu można przy jego pomocy realizować dowolnie złożone zapytania. Na rysunku 4 przedstawiona jest uproszczona postać składni polecenia SELECT.

```
SELECT ListaKolumn  
FROM ListaTabel ON  
WarunkiŁączenia  
WHERE WarunkiSelekcji  
GROUP BY ListaKolumnGrupowania  
HAVING WarunkiSelekcji  
ORDER BY ListaKolumnPorzadkowania
```

```
SELECT Nazwisko, Imie, NazwaMiasta  
FROM Osoby JOIN Miasta  
ON Osoby.IdMiasta=Miasta.IdMiasta  
WHERE Osoby.IdMiasta=2 OR  
Osoby.IdMiasta=4  
ORDER BY Nazwisko
```

Rysunek 4. Uproszczona składnia polecenia SELECT i przykład zapytania.

Czas realizacji

5 x 45 minut

Tematy lekcji

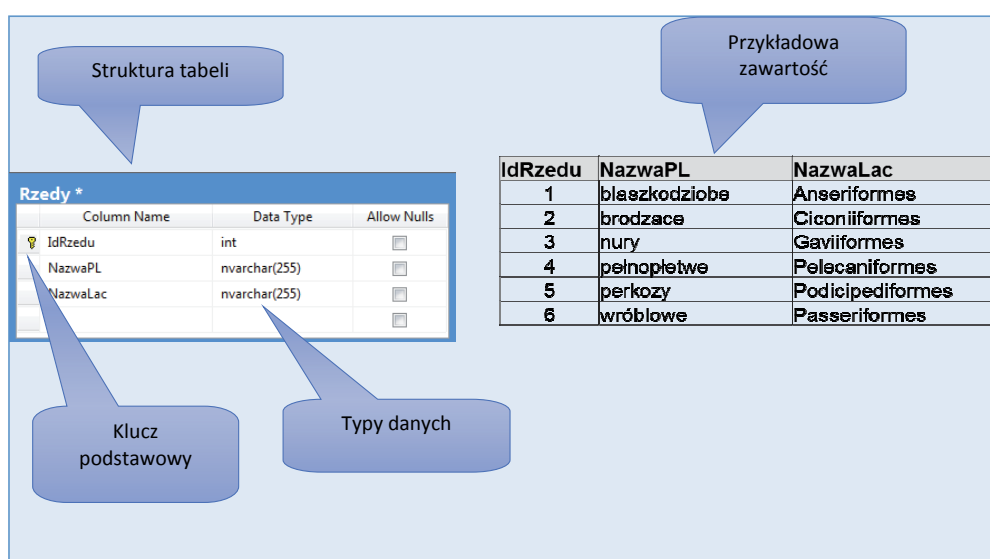
- 1 Bazy wiedzy (statyczne bazy danych)
- 2 Problemowe bazy danych (dynamiczne bazy danych)
- 3 Zapytania proste

LEKCJA NR 1

TEMAT: Bazy wiedzy (statyczne bazy danych)

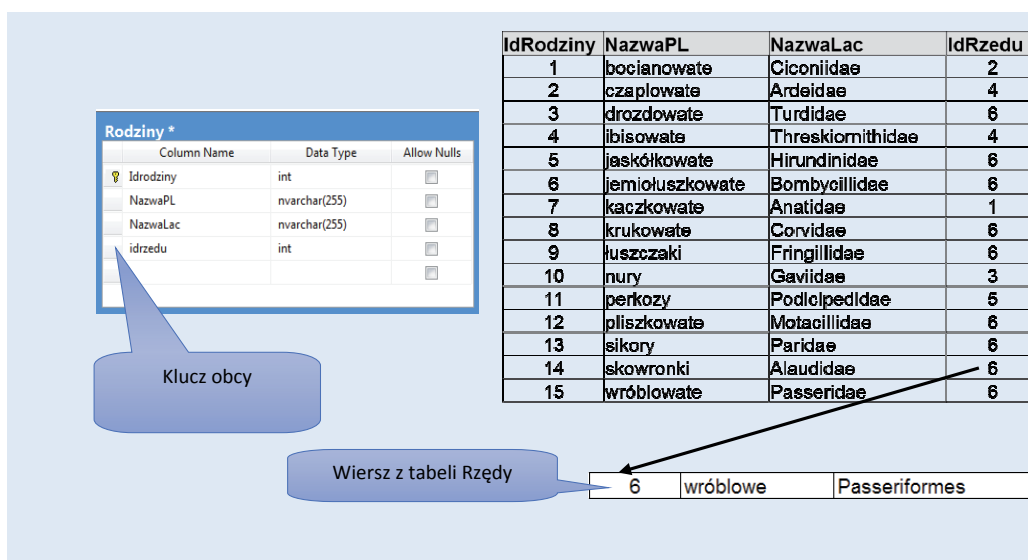
Streszczenie

Zadaniem tej lekcji jest zaprojektowanie i utworzenie statycznej bazy danych, czyli takiej bazy danych, w której dane są niezmiennie i stanowią określony zasób wiedzy z wybranej dziedziny. Przykładem takiej bazy danych może być baza opisująca gatunki ptaków występujących w Polsce. Podział gatunków ptaków na rodziny i rzędy tworzy stały zbiór danych. W ramach lekcji należy omówić przykładową bazę danych PtakiPolskie. Z rozpoznania dziedziny problemu wynika, że systematyka podziału ptaków zawiera pojęcie rzędów. Na rysunku 5 pokazano strukturę i przykładową zawartość tabeli Rzędy, w której zapisane zostaną dane opisujące rzędy ptaków występujących w Polsce.



Rysunek 5. Tabela Rzędy.

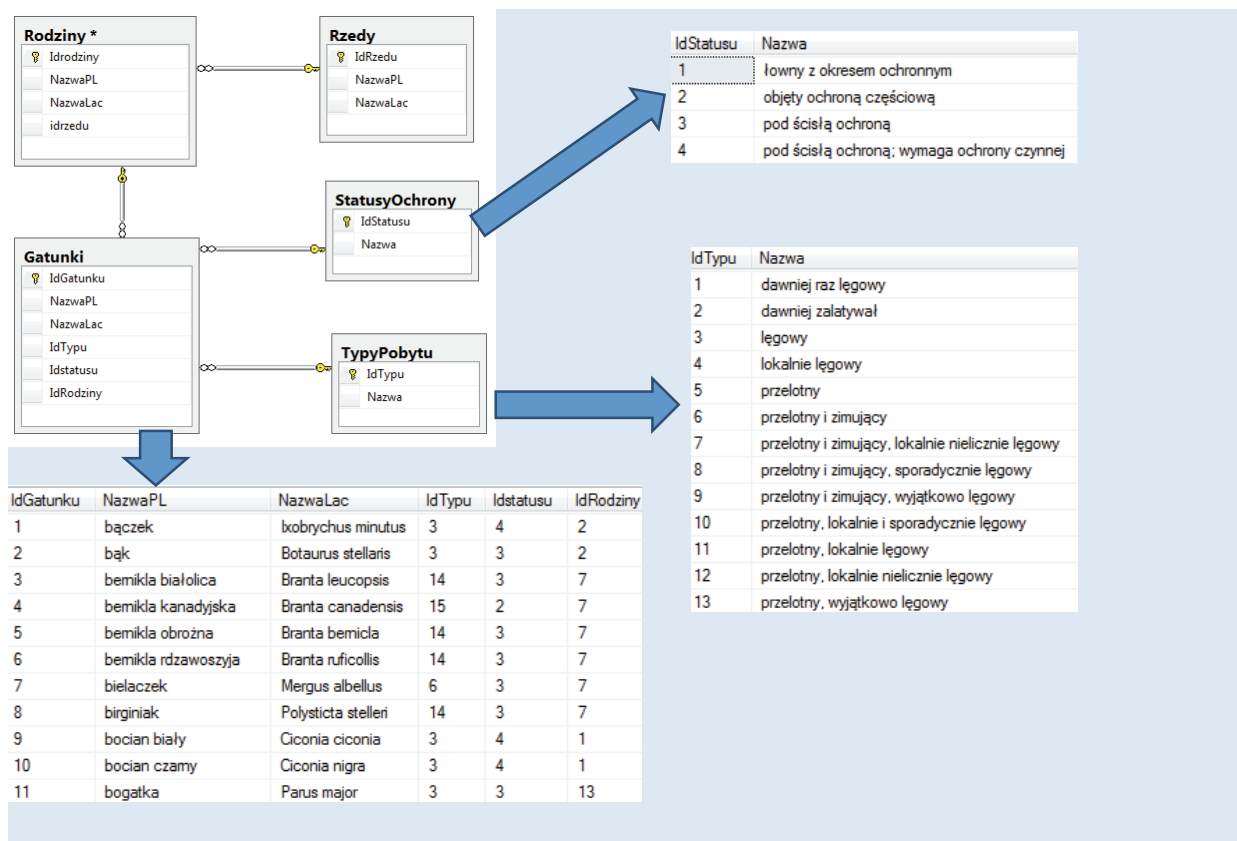
Kolejnym elementem systematyki ptaków jest pojęcie rodziny. Każda rodzina ptaków przypisana jest do określonego rzędu. Na rysunku 6 pokazano przykładową strukturę i zawartość tabeli o nazwie Rodziny.



Rysunek 6. Tabela Rodziny.

W kolejnych krokach należy przygotować tabelę opisującą gatunki ptaków, a także tabele słownikowe opisujące statusy ochrony ptaków oraz typy pobytu.

Na rysunku 7 pokazano końcowy schemat bazy danych o nazwie PtakiPolskie. W materiałach dodatkowych dla tego scenariusza zawarty jest backup bazy danych PtakiPolskie, który można wykorzystać w ramach tej jednostki lekcyjnej.



Rysunek 7. Schemat bazy danych PtakiPolskie.

Przedstawiony przykład bazy danych powinien stanowić inspirację dla uczniów i zachęcić ich do tworzenia baz wiedzy dla innych dziedzin.

Podstawa programowa

Etap edukacyjny: IV, przedmiot: informatyka (poziom rozszerzony)

Cele kształcenia – wymagania ogólne

I. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji.

Treści nauczania – wymagania szczegółowe:

Uczeń:

- 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych;
- 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL).

Cel

Podstawowym celem lekcji jest wyjaśnienie istoty statycznych baz danych, w których można gromadzić dane z różnych dziedzin nauki.

Słowa kluczowe

relacyjny model danych, baza danych, tabela słownikowa, tabela asocjacyjna, klucz obcy

Co przygotować



- Zainstalować MS SQL Server 2012 Express Edition – opis procesu instalacji opisany jest w pliku dodatkowym o nazwie Instalacja pakietu MS SQL Server 2012 Express Edition With Advanced Services – materiały pomocnicze 1



- Wprowadzenie do cyklu lekcji scenariusza omówić z wykorzystaniem prezentacji 1 Bazy wiedzy.

Przebieg zajęć

Wprowadzenie (30 minut)

W trakcie wprowadzenia wykorzystujemy prezentację o nazwie Bazy wiedzy.

Omawiamy przykładową bazę danych zawierającą dane o ptakach polskich i przypominamy podstawowe pojęcia relacyjnego modelu danych (klucz podstawowy, klucz obcy).

Dyskusja podsumowująca (10 minut)

W ramach dyskusji należy zaproponować uczniom przygotowanie, w ramach zadania dodatkowego, projektu bazy wiedzy dla wybranej dziedziny problemu (np. ssaki polskie, systematyka roślin, związki chemiczne itp.). Prace nad projektem bazy wiedzy należy prowadzić w grupach ok. 5-osobowych.

Sprawdzenie wiedzy

Podstawowe sprawdzenie wiedzy można zrealizować na podstawie testu.

Ocenianie

Oceny na podstawie testu oraz pracy grupowej.

Dostępne pliki



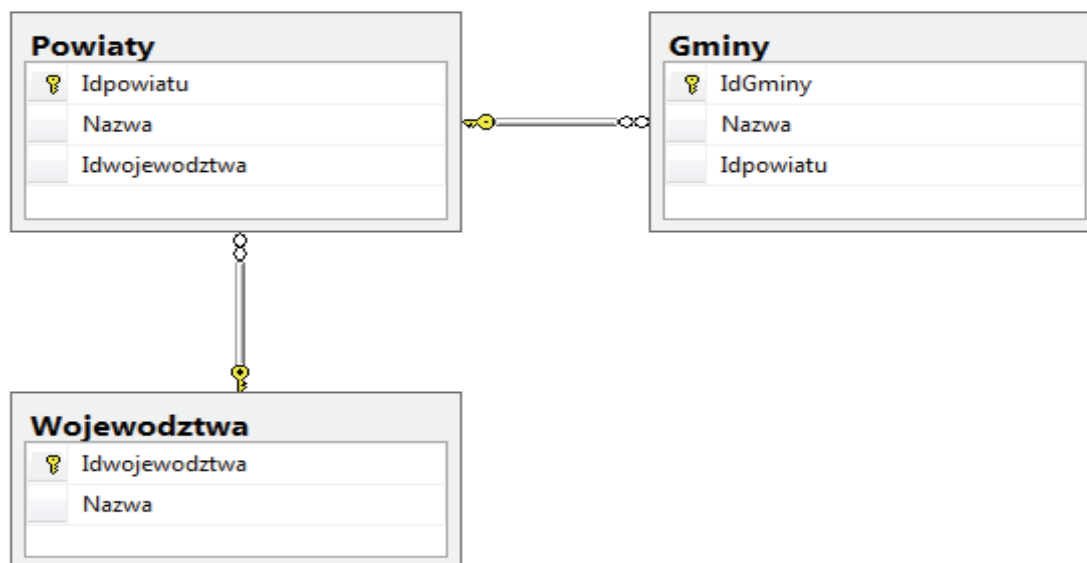
1. Opis procesu instalacji MS SQL Server 2012 – Instalacja pakietu MS SQL Server 2012 Express Edition With Advanced Services.docx (materiały pomocnicze 1)
2. Prezentacja 1 – Bazy wiedzy
3. Backup bazy danych PtakiPolskie – PtakiPolskie.bak (materiały pomocnicze 2)
4. Film instruktażowy 1:
 - Podstawy zapytań
5. Test 1

LEKCJA NR 2

TEMAT: Problemowe bazy danych (dynamiczne bazy danych)

Streszczenie

Celem lekcji jest pokazanie w jaki sposób, dodając do istniejącej bazy danych odpowiednie tabele, można rozszerzyć jej zakres tematyczny. W ramach pokazu do bazy danych PtakiPolskie dodane zostaną tabele opisujące miejscowości w Polsce oraz informacje, o powiecie i województwie, w którym dana miejscowość się znajduje. Na rysunku 8 pokazano tabele opisujące miejscowości.



Rysunek 8. Fragment bazy danych opisujący miejscowości w Polsce

W dołączonej do materiałów scenariusza bazie danych znajdują się tabele pokazane na rysunku 8 wraz z danymi, które zostały umieszczone w omawianych tabelach na podstawie bazy danych Teryt, udostępnionej przez Główny Urząd Statystyczny. W celu ułatwienia dalszego korzystania z danych opisujących miejscowości można zdefiniować w bazie danych widok, który dane będzie udostępniał w jednej (wirtualnej) tabeli o nazwie Lokalizacja.

Definicję widoku można zrealizować za pomocą następującego polecenia w języku SQL:

```

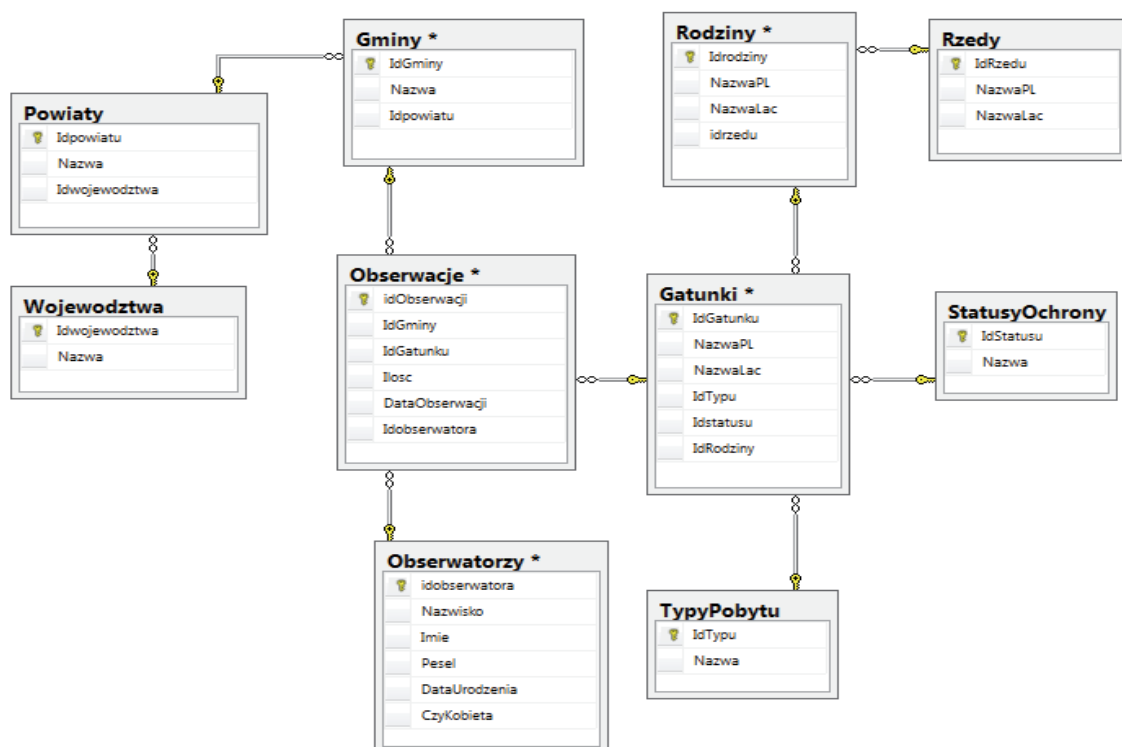
SELECT Gminy.IdGminy,
       Gminy.Nazwa AS Gmina,
       Powiaty.Nazwa AS Powiat,
       Wojewodztwa.Nazwa AS Wojewodztwo
FROM   Gminy JOIN Powiaty ON Gminy.Idpowiatu = Powiaty.Idpowiatu
       JOIN Wojewodztwa ON Powiaty.Idwojewództwa = Wojewodztwa.Idwojewództwa
    
```

Po wykonaniu tego polecenia w bazie danych będziemy mieli dostęp do tabeli o nazwie Lokalizacje, której fragment zawartości pokazano na rysunku 9.

IdGminy	Gmina	Powiat	Wojewodztwo
1	Abramów	lubartowski	LUBELSKIE
2	Adamów	lukowski	LUBELSKIE
3	Adamów	zamojski	LUBELSKIE
4	Adamówka	przeworski	PODKARPACKIE
5	Aleksandrów	bilgorajski	LUBELSKIE
6	Aleksandrów	piotrkowski	LÓDZKIE
7	Aleksandrów Kujawski	aleksandrowski	KUJAWSKO-POMORSKIE
8	Aleksandrów Łódzki	zgierski	LÓDZKIE
9	Alwemia	chrzanowski	MALOPOLSKIE
10	Andrespol	łódzki wschodni	LÓDZKIE
11	Andrychów	wadowicki	MALOPOLSKIE
12	Andrzejewo	ostrowski	MAZOWIECKIE

Rysunek 9. Fragment zawartości udostępnianej przez widok Lokalizacje.

Istotą problemu, który chcemy zaprezentować w ramach tej jednostki lekcyjnej, jest dodanie do bazy wiedzy PtakiPolskie tabel, które umożliwią rejestrowanie obserwacji ptaków. Na rysunku 10 pokazano strukturę tabel o nazwie Obserwatorzy i Obserwacje.



Rysunek 10. Schemat bazy danych do obserwacji ptaków.



Pokazana na rysunku 10 baza danych stanowić będzie podstawę do zapoznania uczniów z możliwościami polecenia SELECT języka SQL.

Dodatkowo w materiałach dołączonych do scenariusza zawarte są backupy baz danych o nazwach:

- Hurtownia
- Przychodnia
- Uczelnia

Przykładowe bazy danych mogą służyć do testowania różnych postaci zapytań realizowanych za pomocą polecenia Select języka SQL.

Podstawa programowa

Etap edukacyjny: IV, przedmiot: informatyka (poziom rozszerzony)

Cele kształcenia – wymagania ogólne

- I. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji.

Treści nauczania – wymagania szczegółowe

Uczeń:

- 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych;
- 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);

Cel

Podstawowym celem lekcji jest utworzenie bazy danych przeznaczonej do rejestracji obserwacji ptaków, która wykorzystuje, przygotowaną w ramach pierwszej lekcji bazę PtakiPolskie

Słowa kluczowe

baza danych, klucz podstawowy, klucz obcy, typy danych, widok, język SQL

Co przygotować

- Przed przystąpieniem uczniów do samodzielnej pracy należy wspólnie pokazać proces tworzenia bazy danych i definiowania przykładowej tabeli.



Przebieg zajęć

Wprowadzenie (10 minut)

W trakcie wprowadzenia należy pokazać uczniom rozszerzenie bazy wiedzy PtakiPolskie o dodatkowe tabele opisujące lokalizacje (miejscowości, powiaty, województwa) oraz obserwacje ptaków.

Praca w zespołach (30 minut)

Grupy uczniów proponują rozszerzenie przygotowanych przez zespoły baz wiedzy o tabele opisujące zdarzenia odwołujące się do statycznej bazy danych.

Dyskusja podsumowująca

W ramach dyskusji należy ocenić zrealizowane projekty i omówić ewentualne błędy.

Sprawdzenie wiedzy

W celu sprawdzenia wiedzy można wykorzystać, zamieszczony w materiałach scenariusza test 2.

Ocenianie

Ocena uczniów na podstawie testu oraz oceny zrealizowanego projektu.

Dostępne pliki



1. Test 2
2. Prezentacja 2 – Obserwacje ptaków.pptx
3. Backupy baz danych – materiały pomocnicze 2, 3, 4, 5
 - PtakiPolskie.bak
 - Hurtownia.bak
 - Przychodnia.bak
 - Uczelnia.bak

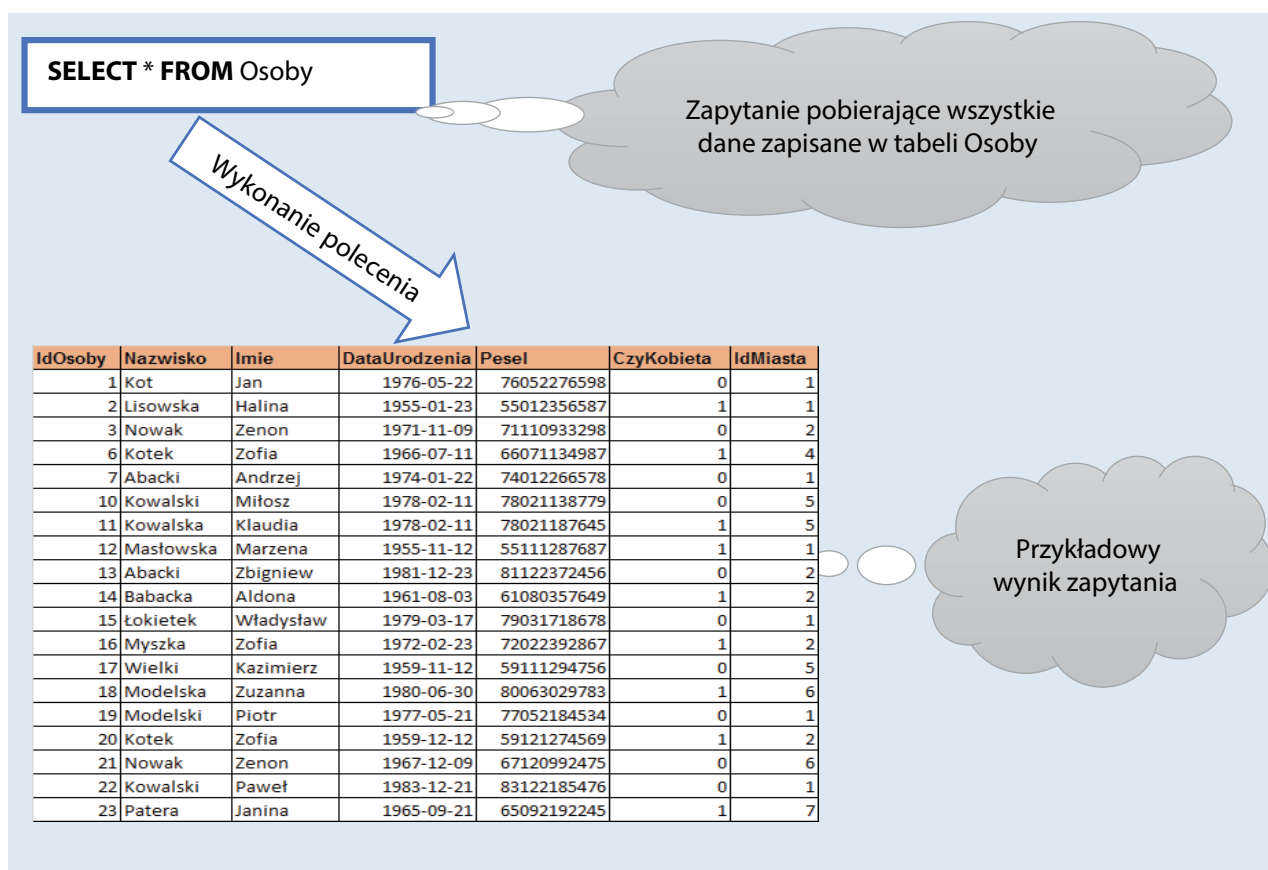
LEKCJA NR 3

TEMAT: Zapytania proste

Streszczenie

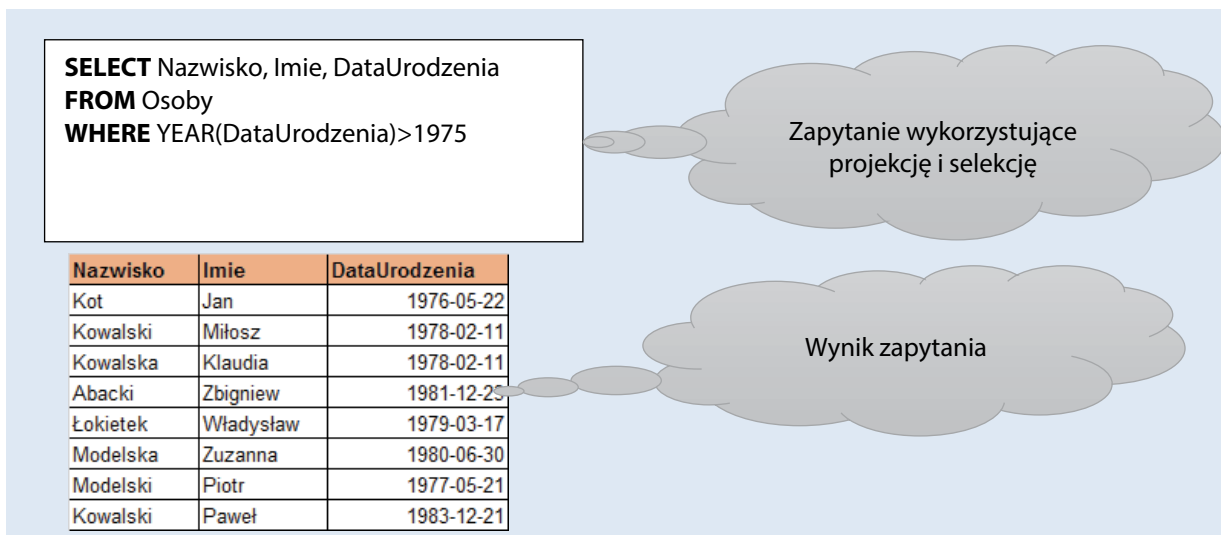
Zapytania do jednej tabeli.

Poznanie możliwości polecenia SELECT rozpoczniemy od prostych zapytań skierowanych do jednej tabeli. W nauce programowania w dowolnym języku większość książek rozpoczyna się od prezentacji najprostszego programu wyświetlającego tekst „Hello World”. Dla polecenia SELECT taką najprostszą postacią jest żądanie pobrania wszystkich danych z wybranej tabeli. Na rysunku 11 pokazano zapytanie pobierające wszystkie dane z tabeli Osoby.



Rysunek 11. Polecenie SELECT – pobieranie zawartości całej tabeli.

Jako listę kolumn po poleceniu SELECT, pokazanym na rysunku 5, zapisano znak * (gwiazdka), który należy interpretować jako wszystkie dostępne kolumny z tabeli, której nazwa jest podana w klauzuli FROM. Podstawą zapytań skierowanych do jednej tabeli jest realizacja operacji projekcji i selekcji, dzięki tym operacjom możemy wybrać dowolny fragment tabeli wyjściowej. Operację projekcji, w zapytaniach SELECT, realizujemy poprzez wymienienie listy kolumn, które powinny znaleźć się w tabeli wynikowej, a operację selekcji – poprzez zapisanie warunku logicznego w klauzuli WHERE. Na rysunku 12 pokazano i omówiono zapytanie do tabeli Osoby, w ramach którego realizowana jest operacja projekcji i selekcji.



Rysunek 12. Zapytanie wykorzystujące operację selekcji i projekcji.

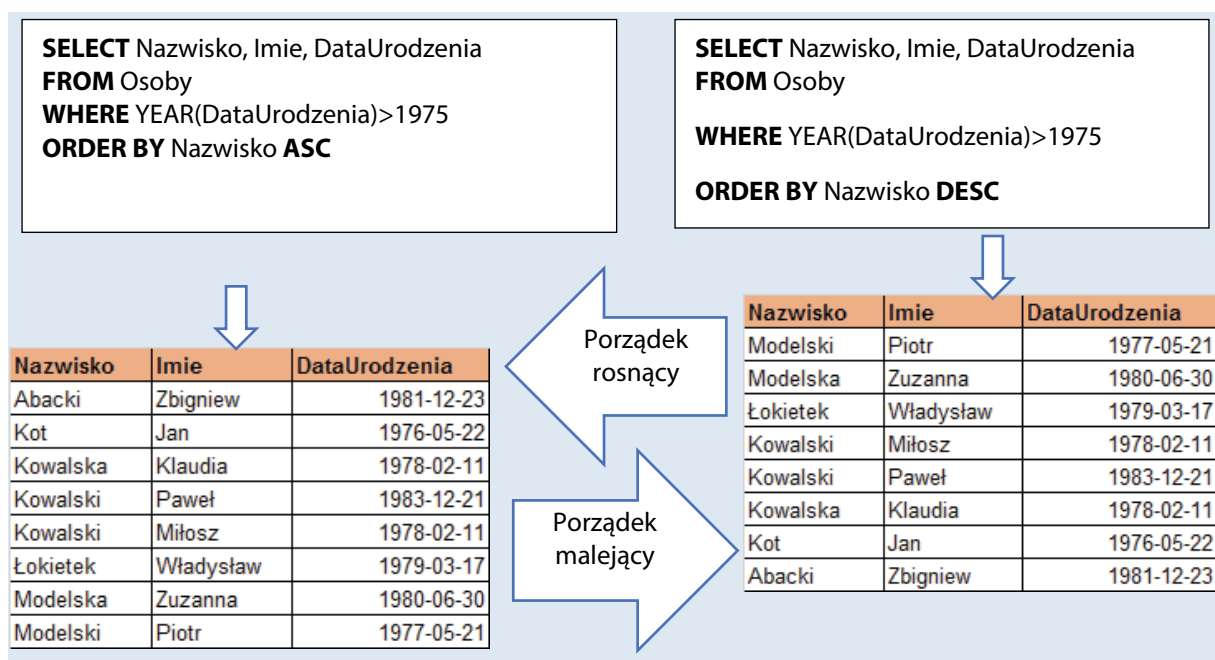
Zapytanie pokazane na rysunku 12 wybiera trzy kolumny z tabeli Osoby (operacja projekcji). W klauzuli WHERE zapytania został zapisany warunek selekcji, czyli do wyniku zapytania zostały przekazane tylko te wiersze tabeli Osoby, dla których warunek logiczny jest prawdziwy. Warunek logiczny wykorzystuje funkcję YEAR, która z danej typu date pobiera rok.

Często chcemy uzyskać wynik zapytania uporządkowany według zadanego kryterium. W poleceniu SELECT porządkowanie wyniku zapytania możemy uzyskać dzięki dołączeniu do zapytania klauzuli **ORDER BY**. Porządkowanie zapytania może być realizowane w kierunku rosnącym lub malejącym. Określenie kierunku porządkowanie dokonujemy za pomocą opcji:

ASC (ascending – rosnąco)

DESC (descending – malejąco).

Na rysunku 13 pokazano wykorzystanie klauzuli ORDER BY w zapytaniach.



Rysunek 13. Przykład zapytania wykorzystującego klauzulę ORDER BY.

W dotychczas przedstawionych przykładach, w kolumnach tabeli wynikowej, przedstawiane były dane pobrane bezpośrednio z tabeli, czyli w takiej postaci, w jakiej zostały zapisane. W zapytaniach możemy przekształcać pobrane dane do innej postaci w zależności od naszych potrzeb. Do przekształcania danych często wykorzystujemy funkcje, które są dostępne w SQL Server 2012.

Wywołując funkcję podajemy jej nazwę, w nawiasach podajemy wartości parametrów, a funkcja przekazuje wynik swojego działania.

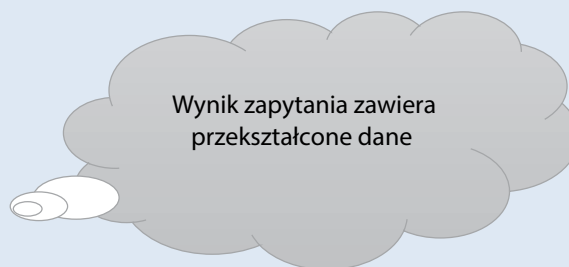
Funkcje dostępne w SQL Server 2012 można podzielić na kilka kategorii:

- Funkcje matematyczne
- Funkcje dla typu daty i czasu
- Funkcje operujące na ciągach znakowych
- Funkcje logiczne
- Funkcje systemowe

Ponieważ w SQL Server 2012 wbudowano kilkadziesiąt różnych funkcji, to zapoznanie z wszystkimi funkcjami znacznie przekraczałoby zakres scenariusza. W trakcie omawiania różnych zagadnień będą wykorzystywane i omówione tylko wybrane. Na rysunku 14 pokazano przykład zapytania, które tworzy wynik wykorzystując przekształcanie wartości danych pobranych z tabeli.

```
SELECT SUBSTRING(Nazwisko,1,1) +SUBSTRING(Imie,1,1) AS Inicjały,
       YEAR(DataUrodzenia) AS RokUrodzenia,
       IIF(CZYKobieta=1, 'Kobieta', 'Mężczyzna') as Płeć
FROM Osoby
WHERE DataUrodzenia >'1977-01-01'
ORDER BY RokUrodzenia
```

Inicjały	RokUrodzenia	Płeć
MP	1977	Mężczyzna
KM	1978	Mężczyzna
KK	1978	Kobieta
ŁW	1979	Mężczyzna
MZ	1980	Kobieta
AZ	1981	Mężczyzna
KP	1983	Mężczyzna



Rysunek 14. Zapytanie wykorzystujące przekształcanie danych.

Do tej pory, w różnych przykładach wykorzystaliśmy trzy funkcje:

- SUBSTRING – zwraca wskazany fragment ciągu znakowego. Wymaga podania trzech parametrów; pierwszym parametrem jest ciąg znakowy, z którego chcemy pobrać fragment, drugim parametrem jest liczba wskazująca, od którego znaku pobieramy fragment, a trzecim określa, ile znaków chcemy pobrać.
- YEAR – zwraca liczbową wartość roku pobraną z daty przekazanej jako parametr.
- IIF – która jako pierwszy parametr przyjmuje wyrażenie logiczne, drugim parametrem jest wartość przekazywana przez funkcję, gdy wyrażenie logiczne jest prawdziwe, a trzecim wartość zwracana przez funkcję, gdy wyrażenie logiczne jest fałszywe.

Podsumujmy krótko dotychczasową wiedzę z zakresu zapytań skierowanych do jednej tabeli:

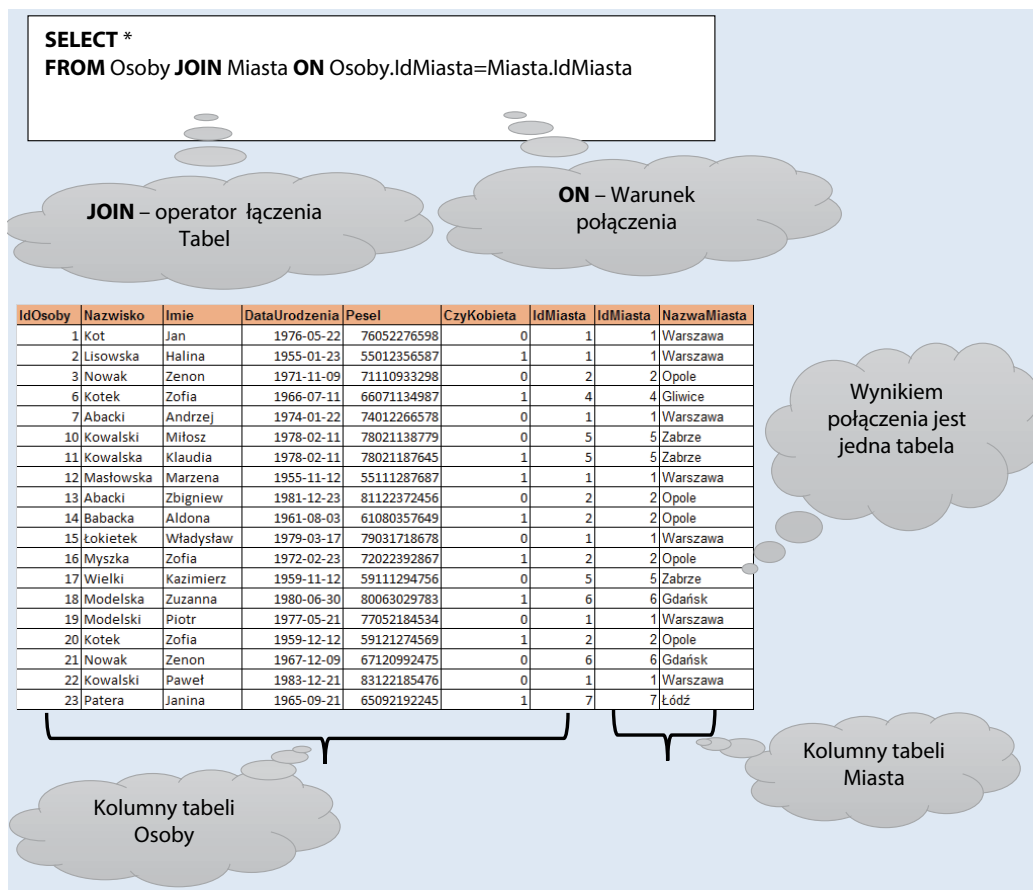
- Możemy wybrać z tabeli tylko te kolumny, które uznajemy za niezbędne w kontekście danego zapytania (**SELECT Nazwisko, Imie, Pesel**) – realizując tym samym operację projekcji
- Musimy podać w klauzuli FROM nazwę tabeli, do której kierujemy zapytanie (**FROM Osoby**)
- Wyrażenia logiczne jako warunki selekcji wpisujemy w klauzuli WHERE (**WHERE IdMiasta=2**)
- Warunki uporządkowania wyniku zapytania podajemy w klauzuli ORDER BY (**ORDER BY Nazwisko DESC**)
- Wartości kolumn wyniku zapytania mogą powstawać w wyniku różnych przekształceń (**IIF (CzyKobieta=1, 'Kobieta', 'Mężczyzna')**)
- Możemy nadawać nazwy kolumnom wynikowym (tak zwane aliasy) (**AS Płeć**)

Z punktu widzenia zapytań prostych, skierowanych do jednej tabeli, pokazaliśmy już podstawowe możliwości polecenia SELECT. Bazy danych składają się z wielu tabel i najczęściej realizowane zapytania pobierają dane, które są rozproszone w wielu tabelach.

Łączenie tabel

Do tej pory, pisząc zapytania, odwoływaliśmy się do jednej tabeli. Przyszedł czas na szersze spojrzenie na całą bazę danych, czyli tworzenie zapytań, w wyniku których tworzymy tabele wynikowe, zawierające dane z wielu tabel. Pozornie mogłoby się wydawać, że używanie wielu tabel jest o wiele trudniejsze i zapytania korzystające z wielu tabel jest trudniej napisać. Naprawdę zmieni się tylko jedno – w klauzuli FROM opisać sposób połączenia tych tabel, które będą brały udział w zapytaniu, ponieważ wynikiem łączenia jest jedna tabela, składająca się ze wszystkich kolumn dostępnych w tabelach, które zostały połączone. Po połączeniu tabel zadanie sprowadza się do działania na jednej tabeli, czyli wszystkie elementy polecenia SELECT omówione wcześniej działają w ten sam sposób.

Na rysunku 15 pokazano przykład zapytania realizującego połączenie tabeli Osoby i tabeli Miasta.



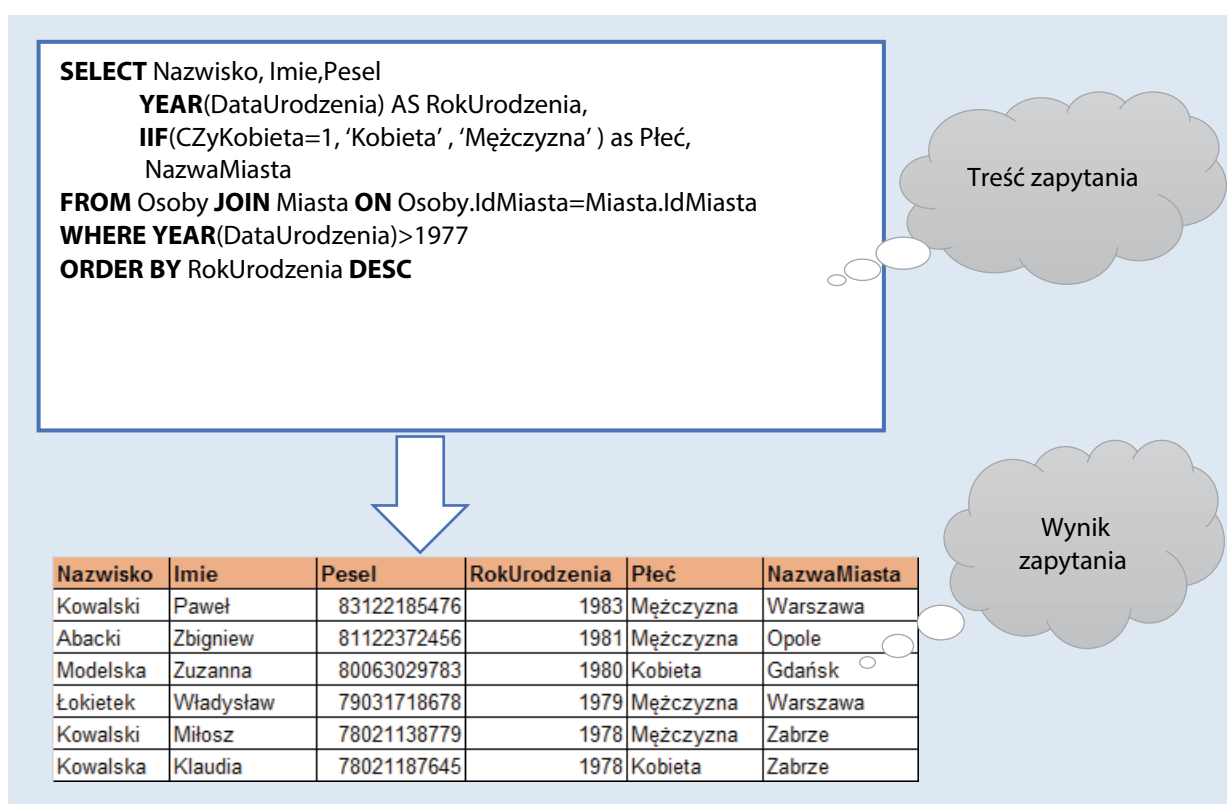
Rysunek 15. Zapytanie wykorzystujące łączenie tabel.

Sens zapytania, które pokazano na rysunku, można opisać w następującym zdaniu:

Proszę wybrać (SELECT) wszystkie kolumny z tabeli Osoby oraz Miasta (znak *), pobieraj dane z tabeli Osoby połączonej z tabelą Miasta (FROM Osoby JOIN Miasta), warunkiem połączenia jest równość wartości IdMiasta w obu tabelach, czyli klucz obcy ma być równy kluczowi podstawowemu (ON Osoby.IdMiasta=Miasta.IdMiasta).

Trzeba zwrócić uwagę na to, że złączenie sprowadza problem do jednej tabeli, w której mamy dostęp do wszystkich kolumn występujących w połączonych tabelach, tym samym po połączeniu możemy wykonywać wszystkie elementy zapytań omówione wcześniej (dla jednej tabeli).

Na rysunku 16 pokazano przykład zapytania wykorzystującego wszystkie poznane do tej pory możliwości polecenia SELECT.



Rysunek 16. Przykładowe zapytanie.

W dotychczasowych rozważaniach, w przykładach łączenia tabel, wykorzystywaliśmy domyślnie złączenie wewnętrzne (ang. **INNER JOIN**), które powoduje, że tylko te wiersze, które spełniają warunek łączenia, znajdują się w tabeli wynikowej. W przypadku stosowania tak zwanego połączenia zewnętrznego (ang. **OUTER JOIN**) będziemy mogli zapewnić występowanie w tabeli wynikowej nawet tych wierszy z wybranej tabeli, dla których nie jest spełniony warunek połączenia. W przykładowych danych tabeli Miasta znajdują się takie, które nie są związane z żadną osobą. Łącząc tabelę Osoby z tabelą Miasta za pomocą operatora złączenia wewnętrznego, nie otrzymamy w wyniku nazw miasta, których klucz nie występuje jako wartość klucza obcego (IdMiasta) w tabeli Osoby. Na rysunku 17 pokazano przykład polecenia wykorzystującego złączenie zewnętrzne.

```

SELECT Pesel, Nazwisko, NazwaMiasta
FROM Osoby OUTER RIGHT JOIN Miasta ON Osoby.IdMiasta=Miasta.IdMiasta
WHERE YEAR(DataUrodzenia)>1977 OR Pesel IS NULL

```



Wartości **NULL**
w kolumnach
pobieranych
z tabeli Osoby

Pesel	Nazwisko	NazwaMiasta
79031718678	Łokietek	Warszawa
83122185476	Kowalski	Warszawa
81122372456	Abacki	Opole
78021138779	Kowalski	Zabrze
78021187645	Kowalska	Zabrze
80063029783	Modelska	Gdańsk
NULL	NULL	Lublin
NULL	NULL	Kraków
NULL	NULL	Rzeszów

Nazwy miast, które
nie są powiązane
z żadną osobą

Rysunek 17. Zapytanie wykorzystujące złączenie zewnętrzne.

Należy zwrócić uwagę na trzy podstawowe różnice zapytania, pokazanego na rysunku, w porównaniu z przykładem wcześniejszym:

- W operacji łączenia wykorzystano opcję **RIGHT OUTER JOIN** (prawostronne łączenie zewnętrzne) – która zapewnia, że do wyniku zapytania, oprócz wierszy spełniających warunek łączenia, zostaną dodane wiersze z tabeli po prawej stronie operatora JOIN (w naszym przypadku tabela Miasta), dla których warunek łączenia jest niespełniony.
- Warunek selekcji (**WHERE YEAR(DataUrodzenia)>1977 OR Pesel IS NULL**) musiał dopuścić do wyniku zapytania wiersze, które w kolumnie Pesel miały wartość NULL.
- W wyniku zapytania dla tych wierszy, które nie spełniają warunku łączenia, w kolumnach Pesel i Nazwisko, występują wartości NULL.

To nie wszystko, co można by powiedzieć o operacji łączenia w zapytaniach. Istnieją także inne operatory połączeń, które można wykorzystywać zamiast operatora **JOIN** (np. **APPLY** lub **PIVOT**), ale omówienie ich wykracza poza ramy tej lekcji.

Podstawa programowa

Etap edukacyjny: IV, przedmiot: informatyka (poziom rozszerzony)

Cele kształcenia – wymagania ogólne

- Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji.

Treści nauczania – wymagania szczegółowe

Uczeń:

- 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych;
- 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL).



Cel

Podstawowym celem lekcji jest zapoznanie uczniów z podstawami polecenia Select języka SQL.

Słowa kluczowe

relacyjny model danych, polecenie SELECT, język SQL, klauzula FROM, klauzula WHERE, klauzula ORDER BY

Co przygotować

- Jeżeli nie zrobiono tego wcześniej, należy, korzystając z załączonych do materiałów scenariusza backupów przykładowych baz danych, odtworzyć je na dostępnym serwerze SQL Server 2012 – materiały pomocnicze 2-5



Przebieg zajęć

Wprowadzenie (30 minut)

W trakcie wprowadzenia należy omówić podstawowe elementy składni polecenia SELECT języka SQL. Najlepszą formą jest wspólne wykonywanie kolejnych postaci poleceń, po wcześniejszym omówieniu elementów składni, które będą wykorzystywane.

Praca w zespołach (10 minut)

Uczniowie, w zespołach, próbują napisać i wykonać polecenie Select dla postawionego zadania.

Podsumowanie (5 minut)

Uczniowie, po zdefiniowaniu ograniczenia, wprowadzają przykładowe dane obserwując działanie zdefiniowanego ograniczenia.

Sprawdzenie wiedzy

Sprawdzenie wiedzy można zrealizować wykorzystując załączony plik test 3 oraz na podstawie wykonanych poprawnie poleceń Select.

Dostępne pliki

1. Test 3
2. Filmy instruktażowe 1 i 2
 - Podstawy zapytań
 - Zapytania do wielu tabel
3. Prezentacja 3 – Podstawy zapytań.pptx



Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



WARSZAWSKA
WYŻSZA SZKOŁA
INFORMATYKI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego