

» Idź do

- Spis treści
- Przykładowy rozdział

» Katalog książek

- Katalog online
- Zamów drukowany katalog

» Twój koszyk

- Dodaj do koszyka

» Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

» Czytelnia

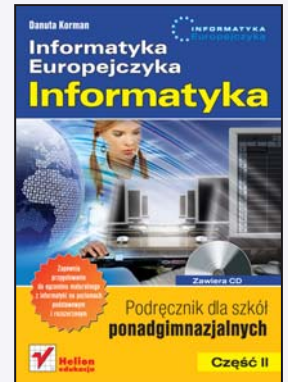
- Fragmenty książek online

» Kontakt

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl
© Helion 1991-2010

Informatyka Europejczyka. Informatyka. Podręcznik dla szkół ponadgimnazjalnych. Część II

Autor: Danuta Korman
ISBN: 978-83-246-2158-3
Format: 168×237, stron: 336



Z komputerami stykamy się dziś niemal każdego dnia. Wykorzystujemy je w pracy, dostarczają nam rozrywki, wyszukujemy za ich pomocą informacje w sieci i komunikujemy się ze znajomymi.

Jednak komputer to nie tylko gry, edytory tekstu, poczta elektroniczna, portale społecznościowe i komunikatory – to także wiele przydatnych narzędzi, niezbędnych do codziennego funkcjonowania we współczesnym świecie.

„Informatyka Europejczyka. Informatyka. Podręcznik dla szkół ponadgimnazjalnych. Część II” przedstawia zasady korzystania z narzędzi i aplikacji komputerowych. Studiując tę książkę, wykonując przykładowe ćwiczenia i rozwiązując zawarte w niej zadania, nauczysz się korzystać z zaawansowanych funkcji bazodanowych arkusza kalkulacyjnego, poznasz relacyjną bazę danych, dowiesz się, jak zbudowany jest komputer i jaką rolę pełni w nim system operacyjny. Dzięki maszynie wirtualnej wykonasz podstawowe operacje na plikach i folderach w różnych systemach operacyjnych.

Z dalszej, bogato ilustrowanej części podręcznika dowiesz się, w jaki sposób w komputerze przechowywane są obrazy i dźwięki oraz jakimi prawami rządzą się algorytmy kompresujące dane. Poznasz również architekturę sieci komputerowych i podstawowe usługi sieciowe. Będziesz mieć także szansę na stworzenie i opublikowanie w internecie swojej strony WWW oraz opanujesz podstawy języka PHP.

Do książki dołączona została płyta CD. Znajdziesz tam materiały niezbędne do wykonania ćwiczeń z podręcznika, kody źródłowe programów opisanych w przykładach i ćwiczeniach, dodatkowe materiały oraz oprogramowanie, które możesz legalnie zainstalować na swoim komputerze.

- Bazodanowe operacje arkusza kalkulacyjnego
- Wprowadzanie i wyszukiwanie danych w bazach danych
- Formularze, raporty, makropolecenia i język zapytań SQL
- Budowa komputera
- Architektura sieci komputerowych
- Protokoły sieciowe
- Rola systemu operacyjnego
- Maszyna wirtualna i różne systemy operacyjne
- Usługi sieciowe
- Grafika komputerowa i multimedia
- Algorytmy kompresji danych
- Strona WWW i elementy PHP
- System zarządzania treścią
- Publikacja opracowań multimedialnych w sieci

Spis treści

Rozdział 1. Bazodanowe operacje arkusza kalkulacyjnego	7
Wstęp	8
1.1. Jednotabelowa baza danych — kartoteka	8
1.2. Podstawowe pojęcia	10
1.3. Tworzenie nowej bazy i import danych	11
1.4. Wyszukiwanie i organizowanie danych	13
1.4.1. Wyszukiwanie z wykorzystaniem formularza	13
1.4.2. Wyszukiwanie informacji za pomocą autofiltru	14
1.4.3. Wyszukiwanie informacji za pomocą filtru zaawansowanego	19
1.4.4. Sortowanie danych	22
1.4.5. Sumy pośrednie (częściowe)	24
1.4.6. Statystyki oparte na raporcie tabeli przestawnej	28
1.5. Zadania	31
1.6. Pytania kontrolne	33
Rozdział 2. Relacyjna baza danych	37
Wstęp	38
2.1. Bazy danych i systemy nimi zarządzające	38
2.2. Projektowanie	41
2.2.1. Zasady projektowania bazy danych	41
2.2.2. Projektowanie bazy danych w programie MS Access	46
2.3. Tabele	48
2.3.1. Tworzenie tabeli w MS Access	49
2.3.2. Typy danych w tabelach	53
2.3.3. Podstawowe właściwości pól tabeli	57
2.3.4. Pole kluczowe tabeli	62
2.3.5. Definiowanie relacji	64
2.3.6. Operacje na tabelach bazy danych	71
2.3.7. Zmiana sposobu prezentowania danych	73
2.3.8. Import, eksport, załączanie tabeli	77

2.4. Kwerendy — zapytania	82
2.4.1. Kwerendy wybierające	83
2.4.2. Kwerendy funkcjonalne — modyfikujące	97
2.4.3. Język zapytań SQL	105
2.5. Formularze	114
2.6. Raporty	118
2.7. Makropolecenia	120
2.8. Ochrona i bezpieczeństwo danych	121
2.9. Pytania kontrolne	128

Rozdział 3. Systemy operacyjne i sieci komputerowe **131**

Wstęp	132
3.1. Budowa komputera	132
3.2. BIOS i rozruch komputera	139
3.3. System operacyjny	140
3.3.1. Budowa systemu operacyjnego	141
3.3.2. Rola systemu operacyjnego	142
3.3.3. Maszyna wirtualna	148
3.3.4. Pytania kontrolne	163
3.4. Sieci komputerowe	164
Wstęp	164
3.4.1. Podział sieci komputerowych	164
3.4.2. Model OSI	182
3.4.3. Podstawowe rodziny protokołów	204
3.4.4. Obsługa i konfiguracja sieci w Windows	207
3.4.5. Bezpieczeństwo sieci	207
3.4.6. Zadania	217
3.5. Podstawy tworzenia stron WWW	219
3.5.1. Podstawowe elementy usługi WWW	219
3.5.2. Projektowanie witryny WWW	220
3.5.3. Popularne formaty hipertekstowe w sieci WWW	220
3.5.4. Struktura dokumentu	223
3.5.5. Kaskadowe arkusze stylów	230
3.5.6. Publikowanie witryny w sieci WWW	233
3.5.7. Systemy zarządzania treścią	235

3.5.8. Czym jest PHP?	237
3.5.9. Zadania	250

Rozdział 4. Multimedia i grafika komputerowa 253

Wstęp	254
4.1. Grafika komputerowa	254
4.1.1. Grafika rastrowa	255
4.1.2. Rozdzielczość a wielkość obrazu	256
4.1.3. Reprezentacja obrazu w komputerze	258
4.1.4. Istota obrazu cyfrowego	268
4.1.5. Cyfrowy obraz ze skanera i cyfrowego aparatu fotograficznego	268
4.1.6. Kompresja stratna i bezstratna	271
4.1.7. Formaty plików bitmapowych	279
4.1.8. Grafika na stronach WWW	281
4.1.9. Grafika wektorowa	287
4.1.10. Grafika bitmapowa a wektorowa	291
4.1.11. Grafika trójwymiarowa — 3D	292
4.2. Dźwięk	302
4.2.1. Teoria dźwięku	302
4.2.2. Zapis dźwięku	304
4.2.3. Zadania sprawdzające do rozdziału 4.	316

Rozdział 5. Tendencje w rozwoju informatyki i jej zastosowań oraz aspekty etyczne, prawne i społeczne w zastosowaniach informatyki 321

Wstęp	322
5.1. Tendencje w rozwoju informatyki i jej zastosowań	322
5.1.1. Krótkie kalendarium wydarzeń ważnych dla rozwoju komputeryzacji	322
5.1.2. Co przyniesie najbliższa przyszłość?	324
5.1.3. Zadania	331
5.2. Aspekty etyczne, prawne i społeczne w zastosowaniach informatyki	332
5.2.1. Prawo autorskie	332
5.2.2. Rodzaje licencji programów komputerowych	333
5.2.3. Zadania	334

2.4. Kwerendy — zapytania


Istotą baz danych jest stworzenie możliwości efektywnego wyszukiwania informacji. Do tego celu służą kwerendy. Są one, obok tabel zawierających dane, podstawowymi **obiektami** definiującymi schematy wyszukiwania potrzebnej informacji. W MS Access istnieją dwa języki definiowania kwerend: język *SQL* (ang. *Structured Query Language* — strukturalizowany język zapytań) oraz język *QBE* (ang. *Query By Example*).

Jeśli więc chcesz znaleźć i pobrać dane, które spełniają pewne warunki, utwórz **kwerendę**, czasem nazywaną zapytaniem. Kwerenda może obejmować dane pochodzące z różnych tabel lub wcześniej utworzonych kwerend (bądź jednych i drugich), może wyszukiwać dane, aktualizować lub usuwać wiele rekordów jednocześnie, tworzyć nowe tabele. Kwerenda pozwala także na wykonywanie obliczeń na danych oraz podsumowań danych.

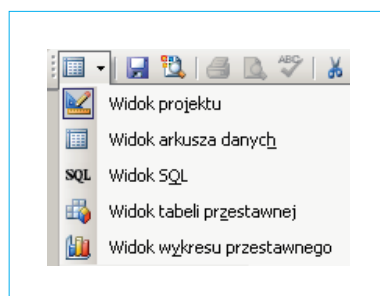
W naszej bazie danych znajduje się już jedna kwerenda, utworzona z zaawansowanego filtrowania danych, wybierająca studentów urodzonych przed 12 marca 1989 roku z Olsztyna. Przejdźmy więc do zakładki z kwerendami.

Po dwukrotnym kliknięciu myszą nazwy kwerendy następuje uruchomienie kwerendy. W wypadku kwerendy **Studenci_z_Olsztyna <1989-03-12** nastąpi otwarcie kwerendy w postaci arkusza (tabelki) z danymi wyszukаныmi zgodnie z kryteriami. Jest to zestaw danych dynamicznych, tworzony na czas uruchomienia kwerendy. Każda zmiana w tabeli, wyświetlonej po uruchomieniu kwerendy, oznacza zmianę w macierzystej tabeli. Sprawdź to, zmieniając datę urodzenia jednego ze studentów. Następnie zamknij kwerendę, przejdź do tabeli **Studenci_dane** i sprawdź, czy zmiana została zapisana w tej tabeli.

Kwerendę można również otworzyć w widoku jej projektu. Po takim otwarciu zobaczymy znane już okno zaawansowanego filtrowania danych, gdzie wybieraliśmy wcześniej pola i kryteria do filtrowania.

Aby w widoku **projektu** uruchomić kwerendę, należy z menu **Rekordy** wybrać polecenie **Uruchom** lub na pasku narzędzi kliknąć przycisk . Możliwe jest również wybranie polecenia **Uruchom** przy użyciu sekwencji następujących klawiszy: lewy **Alt+K**, a następnie **H**.

Kwerendy można oglądać jeszcze w innych widokach, pokazanych na rysunku 2.35.



Rysunek 2.35.

Dostępne widoki kwerend

My będziemy oglądać i projektować je w trzech widokach, wybieranych również z menu **Widok**:

- ▶ **Widok projektu** — wyświetla definicję zapytania w widoku siatki projektowej.
- ▶ **Widok arkusza danych** — wyświetla wynik wykonania zapytania w formie tabelki.
- ▶ **Widok SQL** — podaje definicję zapytania zapisaną w języku SQL.

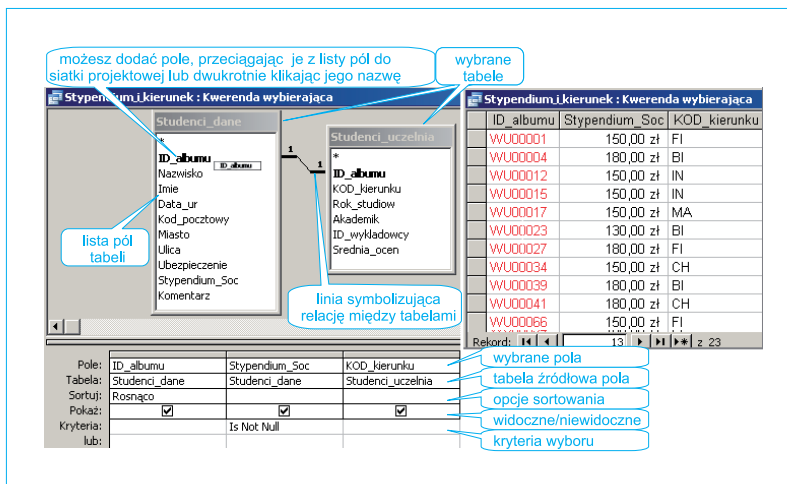
Przeanalizujemy na podstawie przykładów podstawowe typy kwerend.

2.4.1. Kwerendy wybierające

Najczęściej spotykanym rodzajem kwerendy jest kwerenda wybierająca. Pobiera ona dane z jednej lub większej liczby tabel lub wcześniej zdefiniowanych i zapisanych kwerend, używając podanych kryteriów, a następnie wyświetla wybrane dane w żądanym porządku.

Kwerendę można utworzyć, używając **Kreatora kwerend** lub korzystając z widoku **projektu**.

W widoku **projektu**, przez dodanie do siatki projektowej pól zawierających dane z jednej lub kilku tabel, określamy poszczególne parametry wyszukiwania danych. Wypełnianie siatki projektowej przypomina zaawansowane filtrowanie danych. Podstawowa różnica polega na tym, że możemy korzystać z większej liczby tabel. Widok projektu kwerendy wybierającej ilustruje rysunek 2.36. Pomoże on nam w wykonaniu ćwiczenia 2.12.





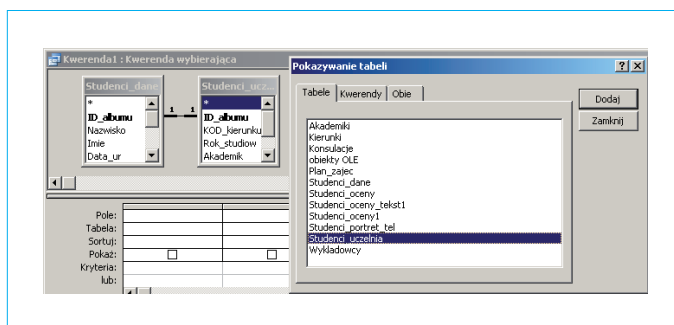
Rysunek 2.36. Kwerenda z kryteriami wyboru wybierająca dane z dwóch tabel. Z lewej widok projektu, z prawej - arkusza danych



Ćwiczenie 2.12.

Zaprojektuj kwerendę wybierającą studentów, którzy otrzymują stypendium. Obok numeru albumu wyświetl kierunek studiów oraz kwotę pobieranych świadczeń. Wybrane dane posortuj rosnąco według numeru indeksu.

Aby zaprojektować kwerendę, najpierw zdecyduj, które z tabel lub już istniejących kwerend będą dla niej źródłem danych. W naszym przykładzie będą to dwie tabele: **Studenci_dane** i **Studenci_uczelnia**. Przejdź do zakładki **Kwerendy** i wybierz polecenie **Utwórz kwerendę w widoku projektu**. W oknie projektu kwerendy pojawi się, jak na rysunku 2.37, okno **Pokazywanie tabeli**. Dodaj niezbędne dwie tabele i zamknij okno. Następnie wybierz pola zawierające potrzebne informacje. Dodaj pola **ID_albumu**, **KOD_kierunku** oraz **Stypendium_Soc** do siatki projektowej, przeciągając je lub dwukrotnie klikając. Możesz też wybrać pole i tabelę z rozwijanej listy bezpośrednio w siatce projektu. Zapisz utworzoną kwerendę pod nazwą **Stypendium_i_kierunek**. Uruchom kwerendę, klikając przycisk . W dynamicznym zestawie danych pojawiły się wybrane pola tabeli. Zwróć uwagę, że w zestawie wybranych danych znaleźli się wszyscy studenci, również ci, którzy nie pobierają stypendium. Powróć do widoku **projektu** .



Rysunek 2.37.


Okno wyboru tabel z projektu kwerendy

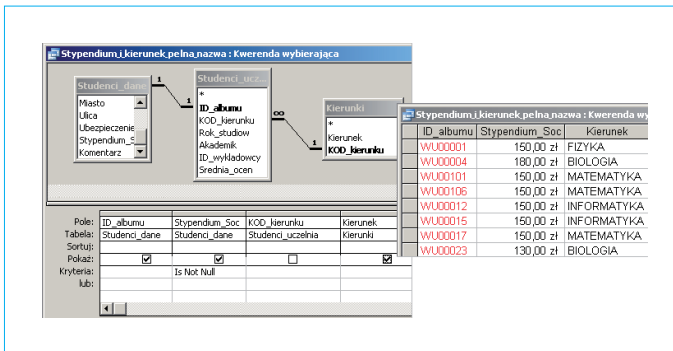
Gdyby wszyscy studenci pobierali stypendium w jednakowej wysokości, w kryteriach wyboru pola **Stypendium_Soc** można by podać tę wartość, np. **150**. Jednak po uruchomieniu kwerendy zniknęłyby z zestawu nie tylko studenci bez stypendium, ale również ci, którzy pobierają stypendium różne niż 150 złotych. Ponieważ jest wiele różnych kwot, korzystnym w tym wypadku kryterium wyboru jest odrzucenie wszystkich rekordów z pustym polem **Stypendium_Soc**. Kryterium **Is Null** spowoduje wypisanie wszystkich studentów, którzy nie pobierają stypendium (mają puste pole **Stypendium_Soc**), dlatego użyj zaprzeczenia i wpisz w kryterium wyboru **Is Not Null**. Zapisz zmiany i uruchom kwerendę. Powinien zostać wyświetlony wybrany zestaw danych, jak na rysunku 2.38.



Ćwiczenie 2.13.

Zmodyfikuj utworzoną w ćwiczeniu 2.12 kwerendę wybierającą, aby zamiast kodu kierunku wypisywała pełną nazwę kierunków studiów każdego studenta pobierającego stypendium socjalne. Zmodyfikowaną kwerendę zapisz jako nową kwerendę.

Do wykonania tego ćwiczenia użyjemy utworzonej wcześniej kwerendy **Stypendium_i_kierunek**. Otwórz ją w widoku **projektu**. Aby wyświetlić pełne nazwy kierunków, zamiast ich kodów potrzebna jest nam tabela **Kierunki**. Dołącz ją do tabel. W tym celu z menu **Kwerenda** wybierz **Pokaż tabelę...** (polecenie dostępne również w menu kontekstowym przy tabelach oraz na pasku narzędzi ). Dodaj tabelę **Kierunki** i pole **Kierunek** do siatki projektowej. Zapisz utworzoną kwerendę (**Plik/Zapisz jako...**) pod nazwą **Stypendium_i_kierunek_pełna_nazwa**. Uruchom ją. Zauważ, że niepotrzebnie wyświetlane są teraz dwie informacje o kierunku studiów: w postaci kodu i w postaci pełnej nazwy. W widoku **projektu** usuń zaznaczenie **Pokaż** dla pola **KOD_Kierunku** i ponownie uruchom kwerendę. Teraz efekt jest taki jak na rysunku 2.38.



Rysunek 2.38.

Okno projektowe korygowanej kwerendy

Przyjrzyj się uważnie utworzonej siatce projektowej oraz relacjom między użytymi w kwerendzie tabelami danych źródłowych. Zauważ, że tabela **Kierunki** jest sprzężona z tabelą **Studenci_uczniowie** przez pole **KOD_kierunku**. Nie ma więc potrzeby zamieszczenia pola **KOD_kierunku** w siatce projektowej. Jeśli usuniemy to pole, efekt wyboru rekordów będzie dokładnie taki sam. W widoku **projektu** zaznacz więc w siatce projektowej pole **KOD_kierunku** i usuń je przyciskiem **Delete** z klawiatury lub używając menu podręcznego. Ponownie uruchom kwerendę. Nic się nie zmieniło. Zapisz zmiany i zamknij okno kwerendy.



Ćwiczenie 2.14.

Zbierz dane z kilku tabel, jak na rysunku 2.39. Wybranych studentów posortuj w kolejności od najmłodszego do najstarszego. Utworzoną kwerendę wybierającą zapisz pod nazwą **Data_ur_rok_studiow_i_telefon**.

Studenci_dane : Tabela				Studenci_uczelnia : Tabela			
ID_albumu	Nazwisko	Imie	Data_ur	ID_albumu	KOD_kierunku	Rok_studiov	Rok_studiov
WU00001	MAKOWSKA	ADRIANNA	1989-03-12	WU00001	FI		3
WU00003	RUMINSKA	OLIWIA	1988-01-11	WU00003	CH		2

Studenci_portret_tel : Tabela			Data_ur_rok_studiov_telefon : Kwerenda wybierająca			
ID_albumu	Telefon_kom	portret_100x100_pik	ID_albumu	Data_ur	Rok_studiov	Telefon_kom
WU00001	323-322-112	Obraz - mapa bit	WU00001	1989-03-12		3 323-322-112
WU00003	555-555-555	Obraz - mapa bit	WU00003	1988-01-11		2 555-555-555
WU00004	505-505-505	Obraz - mapa bit	WU00004	1990-07-29		1 505-505-505
WU00005	606-555-222	Obraz - mapa bit	WU00005	1988-05-27		2 606-555-222
WU00007	601-222-333	Obraz - mapa bit	WU00007	1989-04-17		2 601-222-333
WU00008			WU00008	1989-02-21		2 633-633-633
WU00010			WU00010	1989-03-12		1 888-888-888

Rysunek 2.39.

Kwerenda wybierająca dane z trzech tabel do ćwiczenia 2.14

Przjrzyj się zawartościom dwóch tabel: **Akademiki** i **Wykładowcy**. Zawierają one, odpowiednio: 3 i 20 rekordów. Między nimi nie ma ustanowionej relacji. Co się stanie, jeśli utworzymy kwerendę, której wybranymi polami będą: **Akademik** i **ID_wykladowcy**? Zaprojektuj taką kwerendę, zapisz ją pod nazwą **Akademiki_i_wykladowcy**. Nie stosuj żadnych kryteriów wyboru. Uruchom kwerendę i zobacz, ile powstało par danych. Stało się tak właśnie ze względu na brak relacji między tymi dwiema tabelami. Powstały wszystkie możliwe kombinacje 3 akademików i 20 wykładowców, czyli 60 zestawów danych, co ilustruje rysunek 2.40.

Akademiki_i_wykladowcy : Kwerenda wybierająca		Kwerenda1 : Kwerenda wybierająca	
Akademik	Wykładowcy	Akademik	ID_wykladowcy
DS1	C101	DS1	C101
DS2	C101	DS2	C101
DS3	C101	DS3	C101
DS1	C102	DS1	C102
DS2	C102	DS2	C102
DS3	C102	DS3	C102
DS1	C103	DS1	C103
DS2	C103	DS2	C103
DS3	C103	DS3	C103
DS1	F101	DS1	F101
DS2	F101	DS2	F101
DS3	F101	DS3	F101

Rysunek 2.40.

Kwerenda wybierająca rekordy z tabel niepowiązanych relacjami

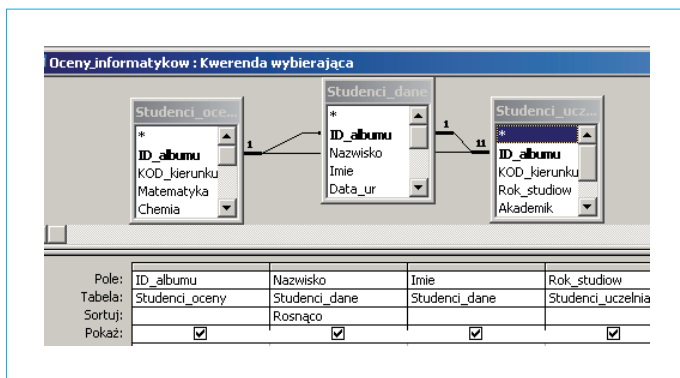


Ćwiczenie 2.15.

Zaprojektuj kwerendę, która będzie wybierała imiona i nazwiska studentów informatyki oraz ich oceny. Wybrane dane posortuj według nazwisk.


Do wykonania tego ćwiczenia użyj dwóch tabel: **Studenci_dane** i **Studenci_oceny**. Dodaj pola z nazwiskiem i imieniem, kierunkiem studiów oraz siedem pól z ocenami. Ustal kryterium wyboru. W tej kwerendzie będzie to kryterium tekstowe, które sprowadza się do podania w komórce **Kryteria** pola **KOD_kierunku** odpowiedniego ciągu znaków, czyli **IN**. Zapisz utworzoną kwerendę jako **Oceny_informatykw**. Uruchom kwerendę i sprawdź jej działanie. Wyłącz pokazywanie kodu kierunku. Zobacz efekt działania kwerendy.

Czy tak utworzona kwerenda pozwoli na sprawdzenie ocen Anny Jaworskiej? Czego zabrakło w kwerendzie? Dodaj więc brakujące, niezbędne pole **ID_albumu**. Dobrze byłoby również ująć w tym wykazie rok studiów danego studenta, co wiąże się z dodaniem trzeciej tabeli, **Studenci_uczelnia**, w której jest ta informacja. Uzupełnij to w widoku **projektu** kwerendy. Zapisz wprowadzone zmiany. Fragment projektu tej kwerendy znajduje się na rysunku 2.41.



Rysunek 2.41.

Fragment projektu kwerendy wybierającej oceny informatyków

Po uruchomieniu kwerendy możesz poddać dane działaniu filtrów, ustawiając kursor w polu tabeli na trzecim roku studiów. Po uruchomieniu filtrowania według wyboru  zobaczysz wszystkich studentów trzeciego roku i ich oceny.

Definiowanie złożonych kryteriów wyboru

W przypadku wszystkich typów pól najprostszym kryterium jest podanie konkretnej wartości:

"WARSZAWA" — dla pola typu **Tekst**,

50 — dla pola liczbowego (**Autonumerowanie, Liczba, Walutowy**),

#1995-12-25# lub #1995/12/25# — dla pola typu **Data/Godzina**,

-1 lub 0 (albo **tak** lub **nie**) — dla pola logicznego **Tak/Nie** (**Tak** odpowiada -1, **Nie** odpowiada 0).



Ćwiczenie 2.16.

Zaprojektuj kwerendę, która wybierze wszystkich ubezpieczonych studentów. Jako wynik wyszukiwania podaj **ID_albumu** oraz kierunek i rok studiów.

W widoku **projektu** wybierz odpowiednie tabele i cztery pola. W polu **Ubezpieczenie** wpisz kryterium, które pozwoli wybrać wszystkich ubezpieczonych studentów. Sprawdź działanie kwerendy. Jeśli wybór jest prawidłowy, wyłącz wyświetlanie wartości pola **Ubezpieczenie**, ponieważ nie ma potrzeby wyświetlania całej kolumny z napisem **tak**. Zapisz kwerendę jako **Ubezpieczeni**.

W kryteriach tekstowych można stosować **znaki wieloznaczne** (zastępcze), które zastępują wiele znaków lub jeden znak. Należą do nich *, ? i #. * zastępuje ciąg znaków dowolnej długości. Aby zrealizować kryterium wyboru wszystkich studentów, których nazwisko rozpoczyna się na literę J, należy w polu **Kryteria** wpisać "J*" lub **Like** "J*". Wystarczy wpisać "J*", a MS Access sam automatycznie uzupełni to kryterium do pełnej postaci, co zauważysz po ponownym otwarciu kwerendy w widoku **projektu**. Natomiast zapis **Like** "*a" spowoduje wybór tych rekordów, które w danym polu tekstowym na ostatniej pozycji ciągu znaków mają literę "a". Tego zapisu często używa się jako kryterium dla pola **Imię** do wyboru płci żeńskiej, gdyż imiona kobiet kończą się na literę a (choć w ostatnich czasach nie jest to już regułą, np. imię Inez).

Oprócz *, w kryteriach można użyć znaku ?, zastępującego pojedynczy znak, oraz znaku # wpisywanego na miejscu cyfry. Przykładowo kryterium "?????" wymusza poszukiwanie wszystkich osób, których nazwisko składa się z 5 znaków, "KO?KIEWICZ" wyszuka takie nazwiska jak: KOLKIEWICZ, KORKIEWICZ, KOTKIEWICZ, ale nie wybierze nazwiska KOKIEWICZ. W roli kryterium dla pola liczbowego można używać zarówno "19##-01-01", jak i "19??-01-01", w obu przypadkach otrzymamy wszystkich urodzonych w dniu 1 stycznia przed 2000 rokiem. Aby wziąć pod uwagę również osoby z młodszych roczników, należałoby wpisać kryterium "####-01-01" lub "????-01-01". Natomiast kryterium dla daty "1989-??-??" lub "1989-##-##" pozwala wybrać wszystkich urodzonych w 1989 roku. Można tych dwóch znaków, ? i #, używać tutaj wymiennie, ponieważ dowolnymi znakami w polu typu **Data** na miejscu zastąpionym znakami ? lub # są zawsze cyfry, więc nie popełnimy błędu i nastąpi poprawne wybranie rekordów zgodnie z kryteriami. Jednak jeśli wybór miałby dotyczyć cyfry w polu typu **Tekst**, to bezwzględnie użyjemy znaku #, dla odróżnienia jej od dowolnego znaku. Na przykład wpis "*"# w polu **Miasto** oznacza kryterium wyboru miast, które mają cyfrę na końcu nazwy, np. LUBICZ1.

W kryteriach wyboru można również podać zakres [], w jakim musi się znaleźć poszukiwany rekord. Wpis **Like** "[A-D]*" oznacza, że wszystkie wybrane nazwy będą rozpoczynały się na jedną z liter z podanego zakresu, więc na A, B, C lub D. Wpis **Like** "*[krs]" oznacza, że wszystkie wybrane nazwy będą kończyły się na jedną z liter podanych w zakresie, czyli na k, r lub s. Korzystanie z zakresów znacznie przyspiesza definiowanie złożonych kryteriów, które można zastąpić złożoną alternatywą.

Oto znaczenie poszczególnych znaków:

- * — zastępuje dowolny ciąg znaków (również ciąg pusty);
- ? — zastępuje jeden znak, który musi wystąpić w polu;
- # — zastępuje cyfrę, która musi wystąpić w polu;

[] — zakres, do jakiego musi należeć dany znak;

[!] — zakres, do jakiego nie może należeć dany znak. Wykrzyknik oznacza negację (np. "[!act]*" oznacza, że ciąg może zaczynać się na dowolną literę oprócz a, c lub t).

Określenie złożonego kryterium liczbowego wymaga użycia jednego z sześciu operatorów relacji:

- > większe,
- >= większe lub równe,
- < mniejsze,
- <= mniejsze lub równe,
- = równe,
- <> różne.

Obok nieskomplikowanych warunków wyboru można tworzyć rozbudowane kryteria, łączące co najmniej dwa warunki. Są to **koniunkcja kryteriów**, czyli wymóg spełnienia wszystkich warunków, oraz **alternatywa kryteriów**, czyli wymóg spełnienia co najmniej jednego warunku. Kryteria można łączyć operatorami logicznymi:

And — koniunkcja, **X And Y** zwraca prawdę, gdy **X** i **Y** są prawdziwe;

Or — alternatywa, **X Or Y** zwraca prawdę, gdy przynajmniej jedno z nich (**X** lub **Y**) jest prawdziwe;

Not — zaprzeczenie, **Not X** zwraca prawdę, gdy **X** jest fałszywe.

In ("tekst1"; "tekst2"; "tekst3"; "tekst4") — wyświetlenie rekordów, które w danym polu mają wartości należące do listy. To kryterium dotyczy danych tekstowych i zastępuje złożoną alternatywę;

Between wartość_początkowa And wartość_końcowa — wyświetlenie wszystkich rekordów, które mają wartości z przedziału od wartości początkowej do wartości końcowej włącznie. Na przykład **Between 10 And 100** oznacza wybór rekordów mających we wskazanym polu wartość z przedziału <10;100>.



Wielkość liter występujących w kryteriach nie ma znaczenia.



Ćwiczenie 2.17.

W naszej przykładowej bazie danych utwórz kwerendę wybierającą studentów urodzonych w pierwszym kwartale 1989 roku. Datę w kryterium można wpi-

ścić w sposób następujący: $\geq 1989-01-01$ And $\leq 1989-03-31$. Po uruchomieniu kwerendy data automatycznie zostanie ujęta w znacznik #data# i kryterium przyjmie następującą postać: $\geq \#1989-01-01\#$ And $\leq \#1989-03-31\#$. Utworzoną kwerendę zapisz jako **Urodzeni_w_pierwszym_kwartale_1989_roku**.

W następnej kwerendzie wybierz studentów mieszkających w akademiku DS1 lub DS2. To zadanie można zrealizować na dwa sposoby. Jeden to podanie alternatywy kryterium wyboru "**DS1**" Or "**DS2**", drugi to wpisanie dwóch warunków do kolejnych pól kryteriów wyboru w siatce projektowej. Obie możliwości pokazano na rysunku 2.42.

Pole:	ID_albumu	Akademik	Akademik
Tabela:	Studenci_uczelnia	Studenci_uczelnia	Studenci_uczelnia
Sortuj:			
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:		"DS1"	"DS1" Or "DS2"
lub:		"DS2"	

Rysunek 2.42.

Dwa warianty kryteriów wyboru

Przy definiowaniu kryteriów należy pamiętać o następujących regułach:

- ▶ kryteria wprowadzone w polach wiersza **Kryteria** połączone są operatorem logicznym **And**, czyli **koniunkcją**, której przykład obrazuje rysunek 2.43;

Pole:	ID_albumu	Data_ur	Data_ur
Tabela:	Studenci_dane	Studenci_dane	Studenci_dane
Sortuj:			
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Kryteria:		$\geq \#1989-01-01\#$	$\leq \#1989-03-31\#$
lub:			

Koniunkcja kryteriów równoważna zapisowi z "And"

Wyłączenie widoczności dublowanego pola Data_ur

Rysunek 2.43. Koniunkcja wyboru studentów urodzonych w pierwszym kwartale 1989 roku

- ▶ kryteria dla jednego pola, wprowadzone w wierszach **Kryteria** i **lub** oraz w każdym następnym wierszu (kryteriów może być więcej niż dwa), połączone są operatorem logicznym **Or**, czyli **alternatywą**.



Wszystkie wymienione powyżej kryteria mogą być stosowane jako reguły poprawności w definicjach pól (ich właściwości) tabeli i formularzy.

Definiowanie wyrażeń



Wyrażenie to dowolna kombinacja operatorów matematycznych i logicznych, stałych, funkcji, nazw pól, formantów i właściwości, której wynikiem jest pojedyncza wartość. Wyrażenia mogą przeprowadzać obliczenia, operować znakami lub testować dane.

Kwerendy mogą wyświetlać pola tabel lub innych utworzonych wcześniej kwerend wybierających oraz **wyrażenia** arytmetyczne, tekstowe i inne. Wyrażenia takie definiowane są w siatce projektowej, w wierszu **Pole**.

Wyrażenie składa się z dwóch części: **tytułu** (nazwa pola) i **formuły** (reguła, według której obliczana jest wartość w tym polu), które zapisujemy, rozdzielając dwukropkiem. Na formułę może składać się kombinacja pól tabeli źródłowej, funkcji Accessa, operatorów arytmetycznych (+, -, /, *) i innych. Ogólna postać wyrażenia jest następująca:

`tytuł_wyrażenia: formuła`



W wyrażeniach nazwy pól z tabeli źródłowej podajemy w nawiasach kwadratowych.





Ćwiczenie 2.18.

Zaprojektuj kwerendę, która połączy zawartości pól tekstowych **Nazwisko** i **Imie** oddzielone pojedynczą spacją. Powstałą kwerendę zapisz jako **Dane_osobowe_studentow**.

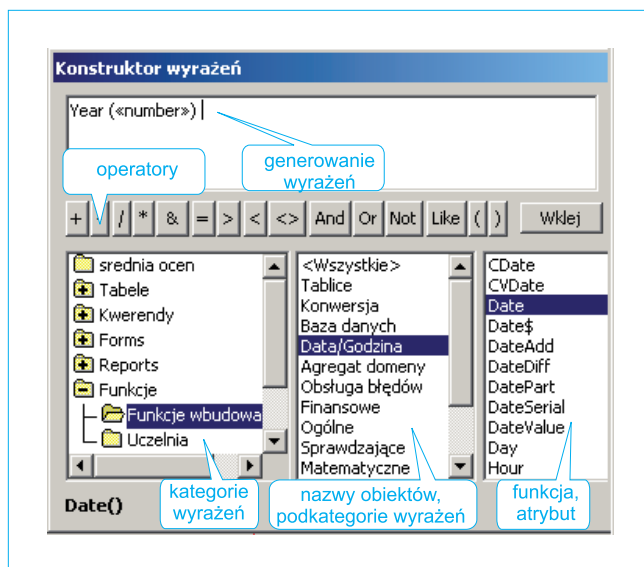
W oknie projektu nowej kwerendy dodaj tabelę **Studenci_dane**. Wybierz z niej trzy pola: **ID_albumu**, **Nazwisko** i **Imie**. W nowym polu siatki projektowej wpisz następujące wyrażenie: `[Nazwisko] & " " & [Imie]` i uruchom powstałą kwerendę. Zauważ, jak jest nazwane pole, które połączyło dwa pola tekstowe. Powróć do projektu. Aby lepiej widzieć powstałe wyrażenie, wybierz polecenie **Powiększenie...** i w otwartym okienku edycji wyrażenia zmień nazwę **Wyr1** na **Nazwisko i Imie** (nie usuwaj dwukropka). Wyłącz pokazywanie nazwiska i imienia w osobnych polach. Ponownie uruchom kwerendę. Poszerz kolumnę, aby można było przeczytać połączone w jednym polu dane. Zapisz kwerendę jako **Dane_osobowe_studentow**.



Aby widzieć całe wyrażenie, wybierz polecenie **Powiększenie...** (**Shift+F2**).

W tworzeniu wyrażeń dużą pomoc oferuje **Konstruktor wyrażeń**, uruchamiany poleceniem **Konstruuj...**  (można go również uruchomić, wybierając z menu kontekstowego opcję  **Konstruuj...**) w pustym polu lub w polu z już ist-

niejącym wyrażeniem w projekcie kwerendy. Polecenie otwiera okno **Konstruktor wyrażen**, pokazane na rysunku 2.44.



Rysunek 2.44.

Okno Konstruktor wyrażen.
Konstruowanie wyrażenia z wykorzystaniem wbudowanej funkcji

Konstruktor wyrażen jest narzędziem, dzięki któremu można generować skomplikowane wyrażenia bez ryzyka popełnienia pomyłki literowej w nazwach pól lub błędu w składni funkcji.




Pola użyte w kwerendzie będą widziane w oknie *Konstruktor wyrażen* dopiero po zapisaniu kwerendy.



Ćwiczenie 2.19.

Zmodyfikuj utworzoną wcześniej kwerendę **Ubezpieczeni**, tak aby wyświetlić rok urodzenia każdego ubezpieczonego studenta. Wykorzystaj do tego **Konstruktor wyrażen** oraz funkcję wbudowaną **Year** — zwracającą rok z pola zawierającego datę. Kwerendę zapisz jako **Rok_urodzenia_studentow**.

Otwórz w widoku **Projektu** kwerendę **Ubezpieczeni**. Do siatki projektowej dodaj pole **Data_ur**. Zapisz kwerendę jako **Rok_urodzenia_studentow**. Zapis jest konieczny, aby było widać dodane przed chwilą pole. W następnym, pustym polu wybierz polecenie **Konstruuuj...**  i wstaw funkcję **Year**, zgodnie z rysunkiem 2.44. W oknie generowania wyrażenia pojawi się treść: **Year («number»)**. Teraz zamiast «number», czyli w miejscu parametru funkcji, wstaw pole **Data_ur**, wybierając je w dolnej części okna **Konstruktor wyrażen**, z bieżącej kwerendy, poleceniem **Wstaw** lub dwukrotnie klikając nazwę pola. W wyrażeniu pojawi się nazwa pola w kwadratowym nawiasie, a całe wyraże-

nie powinno wyglądać następująco: `Year([Data_ur])`. Oczywiście, można wpisywać nazwy samodzielnie, bez wybierania z listy, jednak niesie to ze sobą niebezpieczeństwo niedokładnego podania nazwy pól lub funkcji, co w konsekwencji doprowadzi do błędu w obliczeniach. Uruchom utworzoną kwerendę i sprawdź jej działanie. Kwerenda działa prawidłowo, jednak utworzone pole otrzymało nazwę `Wyr1`. Powróć do widoku **projektu** i zmień nazwę pola `Wyr1` na `Rok`: (nie zapomnij o dwukropku). Wyrażenie powinno wyglądać następująco: `Rok: Year([Data_ur])`. Zapisz zmiany.

Kilka przydatnych funkcji zamieszczonych w tabeli 2.7 przyspieszy i ułatwi pracę. Pamiętaj, że pomoc otrzymasz po wpisaniu w indeksie haseł odpowiedniej nazwy funkcji.

Tabela 2.7. Przydatne wbudowane funkcje MS Access

<code>Date (), Now()</code>	Podanie bieżącej daty lub daty i czasu
<code>DateAdd (...), DateDiff (...)</code>	Obliczanie różnic, sum dat w dniach, latach itp.
<code>Day (...), Year (...), Month (...), Minute (...)</code>	Wyodrębnienie z daty jej części
<code>IsDate (...), IsNull (...)</code>	Sprawdzenie, czy dany obiekt jest odpowiednio datą, wartością <code>Null</code> itp.
<code>Left (...), Right(...), Mid (...)</code>	Wybieranie fragmentu napisu z tekstu. <code>(Left([ID_albumu], 2)</code> zwróci <code>WU</code>
<code>LTrim (...)</code>	Usuwanie spacji
<code>Nz (...)</code>	Konwersja wartości <code>Null</code> na <code>0</code> lub <code>" "</code>
<code>Choose (...)</code>	Wybieranie z listy wartości elementu o podanym numerze
<code>IIf (...)</code>	Wybranie spośród dwu wartości, zależnie od warunku logicznego
<code>Format (...)</code>	Przekształcenie podanej wartości w napis według zadanego formatu

Wśród funkcji, których parametrem są data i czas, znajduje się funkcja `Data()`, zwracająca aktualną datę. Zaprojektuj kwerendę, która obliczy wiek studenta według działania: `wiek=rok(dzisiejsza data)-rok(data urodzenia)`. Odpowiednie wyrażenie utwórz w **Konstruktorze wyrażeń**. Oprócz wieku wyświetl `ID_studenta` oraz kierunek studiów. Utworzoną kwerendę zapisz jako `Wiek_studentow`.

W tym przykładzie parametrem funkcji `Year` jest funkcja `Data()`. Wyrażenie będzie więc miało następującą postać: `wiek: Year(Date())-Year([Data_ur])`.



Ćwiczenie 2.20.

Zaprojektuj kwerendę, która obliczy i wyświetli średnią ocen studentów informatyki. Otrzymany zestaw danych wyświetl w kolejności malejącej średniej. Kwerendę zapisz jako `Srednia_ocen_informatykwow`. Do utworzenia wyrażenia użyj **Konstruktora wyrażeń**, który ułatwi i przyspieszy podanie nazw

wszystkich pól zawierających oceny z poszczególnych przedmiotów. Oprócz średniej wyświetl rok studiów, kod kierunku oraz imię i nazwisko studenta zdefiniowane w kwerendzie **Dane_osobowe_studentow**.

Aby obliczyć wartość średnią z danych zawartych w kilku polach, napisz wyrażenie zawierające sumę tych pól podzieloną przez liczbę przedmiotów sumowanych. W widoku **projektu** nowej kwerendy dodaj table **Studenci_uczelnia** i **Studenci_oceny** oraz kwerendę **Dane_osobowe_studentow**. Dodaj niezbędne pola do siatki projektowej. W nowym polu poleceniem **Konstruuuj...** uruchom **Konstruktor wyrażień**. Utwórz wyrażenie `Srednia_ocen: ([Matematyka]+[Informatyka]+[Jezyk_angielski]+[Jezyk_niemiecki]+[Przedsiębiorczosc]+[Filozofia]+[WF])/7`, ustal sposób sortowania i wyłącz widok pól z ocenami. Zapisz kwerendę.



Ćwiczenie 2.21.

Oblicz średnie ocen wszystkich studentów, biorąc pod uwagę siedem przedmiotów, z których wszyscy studenci mają oceny. Wykonaj obliczenia na grupach rekordów i wyświetl tylko niezbędne dane, jak na rysunku 2.45. Utworzoną kwerendę zapisz jako **Srednia_ocen**.

ID_albumu	KOD_kierunku	Srednia_ocen	Rok_studiow
WU00027	FI	4,14	1
WU00012	IN	4,14	1
WU00053	FI	3,86	2
WU00007	IN	3,86	2
WU00034	CH	3,71	2
WU00071	BI	3,71	3
WU00106	MA	3,71	1

Rysunek 2.45.

Kwerenda wybierająca dane z polem obliczeniowym (ćwiczenie 2.21)



Ćwiczenie 2.22.

Zaprojektuj kwerendę, która na podstawie średnich ocen obliczonych w ćwiczeniu 2.21 wypisze obok ID studenta i kierunku studiów średnią ocen wraz z informacją, czy student otrzyma stypendium naukowe, czy też nie. Kryterium przyznania stypendium to średnia ocen, która wynosi **minimum 4,0**. Jeśli studentowi przysługuje stypendium, w polu wpisz słowo **Stypendium**, w przeciwnym wypadku w polu pojawi się napis **Brak**. Utworzoną kwerendę zapisz jako **Stypendium_naukowe**.

W widoku **projektu** nowej kwerendy tym razem zamiast tabeli użyj utworzonej wcześniej kwerendy **Srednia_ocen**. Wybierz z niej trzy niezbędne pola: **ID_albumu**, **KOD_kierunku** oraz **Srednia_ocen**. W nowym polu utwórz wyrażenie, wykorzystując okno **Konstruktor wyrażień**. Wśród funkcji wbudowanych, zgrupowanych w bloku funkcji przepływu sterowania programu,

znajduje się funkcja wyboru **Iif**, która w oknie **Konstruktor wyrażen** pojawi się jako: **Iif** («expr»; «truepart»; «falsepart»). Należy ją rozumieć jako: **Iif**(warunek;wartość_gdy_prawda,wartość_gdy_fałsz). W oknie wyrażenia wpisz brakującą nazwę pola oraz parametry funkcji: **Stypendium_naukowe**: **Iif**([Srednia_ocen]>=4;"Stypendium";"Brak"). Sprawdź poprawność działania kwerendy. Zapisz ją.

W kolejnym polu, które nazwij **Stypendium_naukowe2**, ponownie wykorzystaj funkcję warunkową. Podaj w niej wartości liczbowe **1** i **0**, odpowiednio: dla spełnionego i niespełnionego warunku przydzielenia stypendium. W oknie wyrażenia wpisz nazwę pola oraz parametry funkcji: **Stypendium_naukowe2**: **Iif**([Srednia_ocen]>=4;1;0) i zobacz wynik działania kwerendy. Powróć do widoku **projektu** i zmień właściwości dla tego pola, ustalając format pola najpierw jako typ **Prawda/Fałsz**, później jako typ **Tak/Nie**. Zobacz, jakie wartości liczbowe odpowiadają wartościom **Tak/Nie**. Ustal sposób sortowania według średniej ocen. Wynik działania tej kwerendy ilustruje rysunek 2.46.

Stypendium_naukowe : Kwerenda wybierająca					
	ID_albumu	KOD_kierunku	Srednia_ocen	Stypendium_naukowe	Stypendium_naukowe2
	WU00027	FI	4,14	Stypendium	Tak
	WU00012	IN	4,14	Stypendium	Tak
	WU00053	FI	3,86	Brak	Nie
	WU00007	IN	3,86	Brak	Nie
	WU00034	CH	3,71	Brak	Nie

Rysunek 2.46. Wynik działania kwerendy z wyrażeniem zawierającym funkcję warunkową

Kwerendy parametryczne

Zapytania mogą być tak konstruowane, aby wartości określające kryterium wyboru podawane były dopiero w momencie uruchamiania kwerendy, bez każdorazowej zmiany kryteriów w siatce projektowej. Takie rozwiązanie wymaga, aby w polu z kryteriami lub w wyrażeniu użyć parametru, podając jego nazwę w nawiasach kwadratowych []. Parametrów może być kilka.

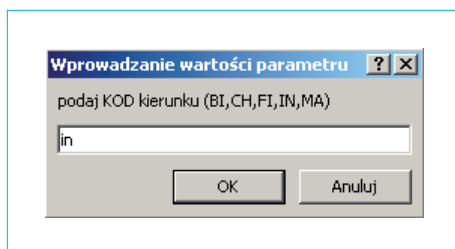


Ćwiczenie 2.23.

Zaprojektuj kwerendę, która będzie wyświetlała ID i miasto zamieszkania wszystkich studentów z danego kierunku studiów. Przy każdym uruchomieniu kwerendy będzie można podać kod tego kierunku, którego dane studentów mają być wyświetlone. Utworzoną kwerendę zapisz jako **Studenci_wg_kierunku_z_parametrem**.

W widoku **projektu** dodaj tabele z danymi adresowymi i uczelnianymi studentów. Wybierz niezbędne pola. W kryterium wyboru pola **KOD_kierunku** podaj kryterium zawierające parametr. Wystarczy wpisać w nawiasach kwa-

dratowych treść komunikatu [podaj KOD kierunku], który pojawi się na ekranie w oknie wprowadzania parametru. Zapisz kwerendę i sprawdź jej działanie. Można ułatwić sobie wypełnienie pola przez podanie podpowiedzi z wypisanymi kodami kierunków, jak na rysunku 2.47. Kryterium przyjmie wówczas postać: [podaj KOD kierunku (BI,CH,FI,IN,MA)]. Nie ma konieczności podania parametrów dużymi literami, Access radzi sobie zarówno z małymi, jak i dużymi literami.



Rysunek 2.47.

Okno podawania parametrów kwerendy parametrycznej



Ćwiczenie 2.24.

W kwerendach można stosować wiele parametrów. Przeprojektuj powstałą w ćwiczeniu 2.23 kwerendę, tak aby wybierała studentów z danego kierunku oraz mieszkających w mieście (określonym parametrem). Zauważ, że kolejność podawania parametrów jest zgodna z kolejnością pól w siatce projektowej. Dodaj drugie miasto do kryterium wyboru, jak na górze rysunku 2.48. Uruchom zapytanie i wybierz studentów informatyki mieszkających w Olsztynie i Toruniu. Utworzoną kwerendę zapisz pod nazwą **Studenci_wg_kierunku_z_wieloma_parametrami**.

Pole:	ID_albumu	KOD_kierunku	Miasto
Tabela:	Studenci_dane	Studenci_uczelnia	Studenci_dane
Sortuj:			
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:		[podaj KOD kierunku]	[podaj 1 miasto zamieszkania] Or [podaj 2 miasto zamieszkania]
lub:			

Pole:	ID_albumu	KOD_kierunku	Miasto
Tabela:	Studenci_dane	Studenci_uczelnia	Studenci_dane
Sortuj:			
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:		[podaj KOD kierunku]	[podaj 1 miasto zamieszkania]
lub:			[podaj 2 miasto zamieszkania]

A jaki będzie efekt działania tych kryteriów?

Rysunek 2.48. Kwerenda z wieloma parametrami

Zwróć uwagę, że na rysunku 2.48 użyta została alternatywa dwóch warunków parametrycznych. Zastanów się, dlaczego dolne rozwiązanie jest niepoprawne dla tego przykładu. Jakie rekordy otrzymamy w wyniku jej działania?

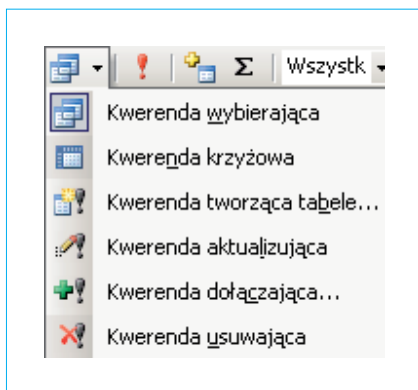


Ćwiczenie 2.25.

Na podstawie kwerendy **Srednia_ocen** utwórz nową kwerendę parametryczną, która będzie wybierała studentów ze średnią ocen między dwiema wartościami podawanymi przy uruchamianiu zapytania w oknie z komunikatem **Podaj minimalną średnią** oraz **Podaj maksymalną średnią**. Utworzoną kwerendę zapisz jako **Srednie ocen z parametrem**.

2.4.2. Kwerendy funkcjonalne — modyfikujące

Oprócz kwerend wybierających, możemy tworzyć kwerendy funkcjonalne. Lista kwerend jest dostępna w widoku **projektu** kwerendy, po kliknięciu myszą strzałki obok przycisku typów kwerend, co ilustruje rysunek 2.49. Typy kwerend są dostępne również w menu kontekstowym obszaru tabel.



Rysunek 2.49.

Typy kwerend

Przy ikonach czterech typów kwerend widoczny jest wykrzyknik, co oznacza, że wykonanie tych kwerend skutkuje trwałymi zmianami w bazie danych. Każdorazowe ich uruchomienie może powodować przeliczanie wartości lub usuwanie danych, jeśli spełnione zostaną określone kryteria. Trzeba więc być szczególnie ostrożnym. Należy również czytać komunikaty pojawiające się na ekranie przy próbie uruchomienia kwerend funkcjonalnych. Warto też wykonać kopię bazy lub przynajmniej tabel, aby nie stracić cennych danych.

Kwerenda aktualizująca

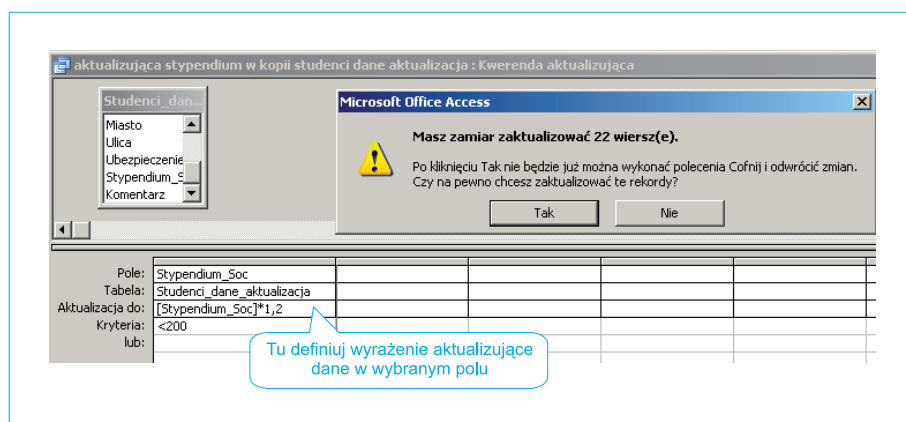
Utwórz kwerendę opartą na tabelach lub kwerendach zawierających rekordy, które mają zostać zaktualizowane.

Zadanie zrealizujemy, korzystając z tabeli **Studenci_dane**. Wykonaj jej kopię i nazwij ją **Studenci_dane_aktualizacja** (z menu kontekstowego wybierz **zapisz jako...**). Ponieważ każde uruchomienie takiej kwerendy oznacza przeliczenie wartości, trzeba być bardzo ostrożnym przy jej uruchamianiu. To właśnie dlatego kwerendy tego typu mają ikonę opatrzoną wykrzyknikiem, sym-

bolizującym niebezpieczeństwo. Zaktualizujemy dane w kopii tabeli, zwiększając stawki obecnych stypendiów socjalnych o 20%. Podwyżka dotyczy jedynie tych osób, których stypendium jest mniejsze niż 200 złotych.

Utwórz nową kwerendę w widoku **projektu**. Wybierz tabelę **Studenci_dane_aktualizacja**. Kliknij strzałkę obok przycisku **Typ kwerendy** i wybierz **Kwerenda aktualizująca**, zmieniając w ten sposób typ kwerendy. Typ kwerendy możesz również wybrać z menu kontekstowego.

Przeciwnij z listy pól do siatki projektu pole **Stypendium_Soc**, które będziesz aktualizować lub dla którego chcesz ustawić kryteria. Wpisz **<200** w komórce **Kryteria**. W komórce **Aktualizacja do:** wpisz wyrażenie lub wartość, która ma być wykorzystana do zmiany pól, tak jak to pokazano na rysunku 2.50.



Rysunek 2.50. Kwerenda aktualizująca dane

Zapisz kwerendę pod nazwą **Aktualizujaca_stypendium_w_kopii Studenci_dane_aktualizacja**.

UWAGA! Każdorazowe uruchomienie tej kwerendy spowoduje kolejne przeliczenie kwoty stypendiów spełniających kryterium. Dlatego uruchomienie kwerendy aktualizującej jest poprzedzone ostrzeżeniem i zapytaniem, czy na pewno chcesz zaktualizować dane.

Po wykonaniu kwerendy sprawdź wartości stypendiów w kopii tabeli.

Zwróć uwagę na ikonę tej aktualizującej kwerendy. Wykrzykownik w ikonie ma nie tylko odróżniać tę kwerendę od kwerend wybierających, ale i ostrzegać przed przypadkowym lub nieprzemyślanym użyciem.



Użycie kwerendy aktualizującej jest niezmiernie wygodne i pomocne w przeliczaniu istniejących lub podawaniu nowych wartości. Jednak jeśli nie zamierzamy w przyszłości użyć takiej kwerendy, po jej zastosowaniu można ją po prostu usunąć.

Kwerenda tworząca tabelę

Utwórz kwerendę, wybierając table lub kwerendy zawierające te rekordy, które mają się znaleźć w nowej tabeli.

Rozwiązanie zadania takiego typu przebiega w dwóch etapach. Etap pierwszy to utworzenie kwerendy wybierającej, a następnie przekształcenie jej w **kwerendę tworzącą nową tabelę**. Po podaniu nazwy powstałej tabeli w zakładce z tabelami pojawi się nowy zestaw danych — nowa tabela. Do ćwiczenia użyjemy więc już zdefiniowanej wcześniej kwerendy.



Ćwiczenie 2.26.

Kwerendy **Oceny informatyków** użyj do utworzenia nowej tabeli, która będzie zawierać dane z ocenami studentów informatyki.

W zakładce zawierającej kwerendy odszukaj kwerendę **Oceny informatyków** i wykonaj jej kopię, a następnie zmień nazwę kopii na **Oceny_informatykw_tworzaca_tabela**. Otwórz ją w widoku **projektu**. Zmień jej typ na kwerendę tworzącą nową tabelę, wybierając ten typ z listy typów kwerend, jak na rysunku 2.49. Podaj nazwę nowej tabeli, która powstanie po uruchomieniu kwerendy: **Oceny_informatykw_tabela_z_kwerendy**. Uruchom przekształconą kwerendę. Przeczytaj komunikat i potwierdź chęć utworzenia tabeli z 14 rekordami. Zamknij okno kwerendy, zapisując zmiany, i sprawdź w zakładce z tabelami, czy powstała nowa tabela i jakie zawiera rekordy.

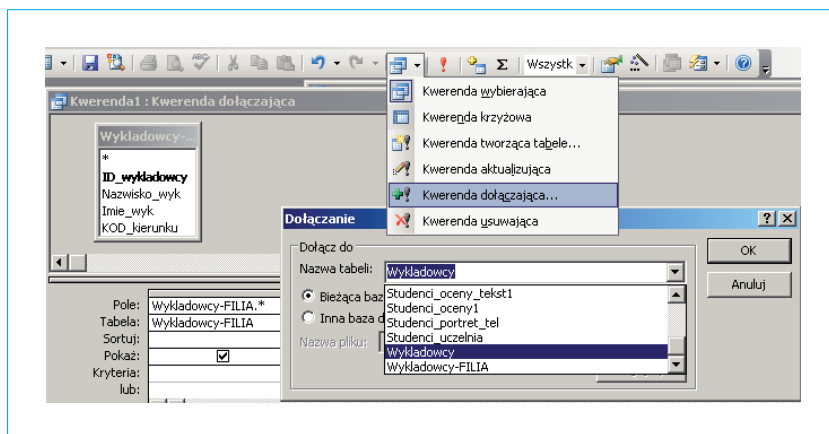
Kwerenda dołączająca

W bazie danych *Uczelnia_filia.mdb* znajduje się tabela **wykladowcy-FILIA**. Zamieszczono tam dane wykładowców, którzy uczą języków obcych. Utworzymy kwerendę, która dołączy dane tych wykładowców do tabeli **Wykladowcy**. W tym celu należy zaimportować tabelę **wykladowcy-FILIA**. Po zaimportowaniu obejrzyj jej zawartość. Zwróć uwagę, że utworzona została zgodnie z projektem tabeli **Wykladowcy**. Aby dołączyć tabelę do tabeli, użyjemy **kwerendy dołączającej**.

Kwerenda dołączająca dodaje na koniec tabeli rekordy pochodzące z innej tabeli (lub kilku tabel). Jest to kwerenda z grupy kwerend funkcjonalnych. Jej ikona również jest wyróżniona znakiem wykrzyknika. Każdorazowe uruchomienie kwerendy dołączającej będzie próbą ponownego dodania rekordów do tabeli.

Utwórz nową kwerendę w widoku **projektu**. Wybierz tabelę, którą chcesz dołączyć, czyli zaimportowaną tabelę **wykladowcy-FILIA**. Wybierz gwiazdkę z listy pól tabeli, która symbolizuje dołączanie wszystkich pól. Teraz zmień typ kwerendy z wybierającej na dołączającą. Wybierz z listy lub wpisz nazwę tabeli, do której mają być dołączone rekordy po uruchomieniu kwerendy, czyli **Wykladowcy**, jak na rysunku 2.51. Utworzoną kwerendę zapisz jako **Dolaczajaca_tabela_Wykladowcy-FILIA**. Zanim uruchomisz tę kwerendę, utwórz ko-

pię tabeli **Wykładowcy**. Będzie potrzebna do dalszej analizy. Teraz uruchom kwerendę dołączającą, która wyświetla komunikat o próbie dodania rekordów do tabeli. Potwierdź chęć ich dodania.



Rysunek 2.51. Projektowanie kwerendy dołączającej dane z innej tabeli



Na kopii tabeli *Wykładowcy* sprawdź, jak będzie się zachowywała kwerenda przy kilku uruchomieniach. W widoku *projektu* zmień tabelę *Wykładowcy* na *kopia Wykładowcy* przez ponowne wybranie typu kwerendy dołączającej. Uchroni to dane potrzebne do dalszych ćwiczeń. Czy istniejące już w tabeli dane będą nadpisywane, czy podjęta zostanie próba ponownego dodania tych samych wykładowców? Innymi słowy, czy następuje jakaś analiza danych, czy tylko mechaniczne dopisanie danych na końcu tabeli? Czy reguła poprawności i właściwość indeksowania bez powtórzeń pozwolą na zdublowanie danych? Wyciągnij wnioski.



Wykonaj drugą część tego ćwiczenia. W bazie *Uczelnia_filia.mdb* znajduje się tabela **Konsultacje_FILIA**, w której zamieszczone zostały terminy konsultacji dołączonych wykładowców. Utwórz kwerendę dołączającą te dane do tabeli **Konsultacje**. Kwerendę zapisz jako **Dołączająca_konsultacje_FILII**. Dodaj konsultacje pracowników z filii. Jeśli wcześniej nie zostały dodane dane wykładowców, nastąpi odmowa dodania konsultacji tych pracowników. Dlaczego? Wyjaśnienie ma związek z projektowaniem relacji między tabelami. W przypadku tabel połączonych relacją **jeden-do-jednego** najpierw należy wpisać dane do tabeli nadrzędnej (**Wykładowcy**), dopiero później możemy wpisać dane do tabeli podrzędnej (**Konsultacje**).

Kwerenda usuwająca

Jeśli rekordy zostaną usunięte za pomocą kwerendy usuwającej, która usuwa z jednej lub kilku tabel wiersze spełniające określone kryteria, czynności tej nie można cofnąć. Dlatego też przed uruchomieniem kwerendy należy obej-

rzeć dane, które zostały przeznaczone do usunięcia. Aby to uczynić, należy kliknąć przycisk **Widok** na pasku narzędzi i obejrzeć kwerendę w widoku **arkusza danych**.



Zawsze należy sporządzać zapasowe kopie danych. Jeśli zostaną usunięte niewłaściwe rekordy, można je odzyskać z kopii zapasowej.



Ćwiczenie 2.27.

Z utworzonej w ćwiczeniu 2.26 tabeli usuń dane tych studentów, którzy nie mają żadnej poprawki, czyli otrzymali ze wszystkich przedmiotów nauczania ocenę różną od niedostatecznej (2).

W tym celu utwórz nową kwerendę w widoku **projektu**. Wybierz tabelę **Oceny_informatykw_tabela_z_kwerendy**. Dodaj nazwy przedmiotów do wiersza **Pole** w siatce projektu i określ kryteria usuwania, wpisując **<>2** dla każdego przedmiotu. Uruchom kwerendę wybierając wszystkie te rekordy, które za chwilę zostaną usunięte (trwale) z tabeli. Powinny pokazać się tylko oceny pozytywne studentów, którzy za chwilę zostaną usunięci.

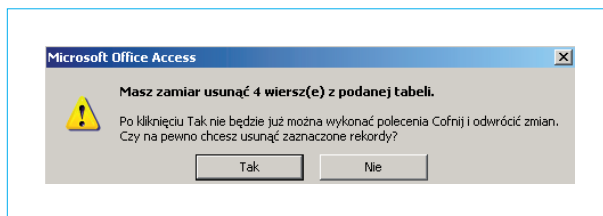
W widoku **projektu** na pasku narzędzi kliknij strzałkę obok przycisku **Typ kwerendy**, a następnie wybierz typ **Kwerenda usuwająca**. W wierszu **Usuwanie** pojawi się słowo **Gdzie**, jak to zostało pokazane na rysunku 2.52.

Pole:	Matematyka	Informatyk	Język_angielski	Język_niemiecki	Przedsiębiorczość	Filozofia	WF	Oceny_informatykw_tabela_z_kwerendy.*
Tabela:	Oceny_infor	Oceny_infor	Oceny_informa	Oceny_informal	Oceny_informal	Oceny_infori	Oceny_infor	Oceny_informatykw z kwerendy.*
Usuwanie:	Gdzie	Gdzie	Gdzie	Gdzie	Gdzie	Gdzie	Gdzie	Skąd
Kryteria:	<>2	<>2	<>2	<>2	<>2	<>2	<>2	
lub:								

Rysunek 2.52. Kwerenda usuwająca dane z tabeli

Przeciagnij gwiazdkę (*) z listy pól tabeli do siatki projektu kwerendy. W komórce **Usuwanie** pojawi się słowo **Skąd**.

Na pasku narzędzi kliknij przycisk **Uruchom**, aby usunąć wybrane rekordy. Potwierdź jeszcze w oknie komunikatu chęć usunięcia 4 rekordów z tabeli, jak na rysunku 2.53. Po usunięciu danych sprawdź zawartość tabeli **Oceny_informatykw_tabela_z_kwerendy**. Wszyscy studenci mają oceny niedostateczne. Zmień nazwę tabeli na **Poprawki_informatykw**.

**Rysunek 2.53.**

Komunikat o usunięciu danych z tabeli kwerendą usuwającą

W niektórych przypadkach uruchomienie kwerendy usuwającej może spowodować usunięcie rekordów z tabel związanych relacją, nawet jeśli nie zostały one uwzględnione w kwerendzie. Może się tak stać, jeśli tabela znajdująca się w kwerendzie jest stroną „jeden” relacji **jeden-do-wielu** i dla tej relacji została włączona opcja usuwania kaskadowego. Usuwając rekordy w tabeli po stronie „jeden” relacji, usuwa się również rekordy z tabeli po stronie „wiele”.

Kwerenda krzyżowa

Oblicz sumę, zlicz lub wykonaj inne działania, a następnie pogrupuj wyniki według dwóch typów informacji zawartych w tabeli, jak na rysunku 2.54, tworząc krzyżowe zestawienie podsumowanych lub zliczonych danych.

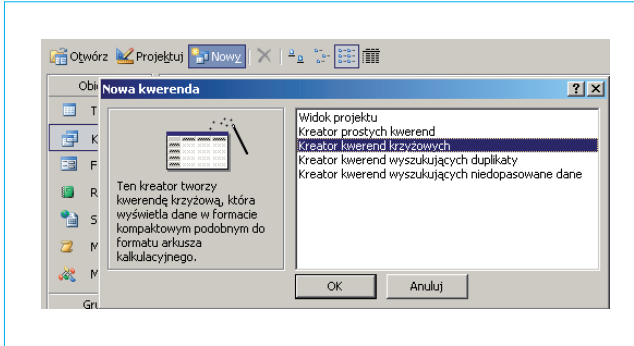
Akademik	Suma ID_albumu	BI	CH	FI	IN	MA
DS1	23	5	4	4	3	7
DS2	24	3	3	6	5	7
DS3	24	8	2	6	6	2

Rysunek 2.54. Kwerenda krzyżowa

W przykładzie tym utworzona została kwerenda zliczająca studentów z danych kierunków mieszkających w poszczególnych akademikach — domach studentów.

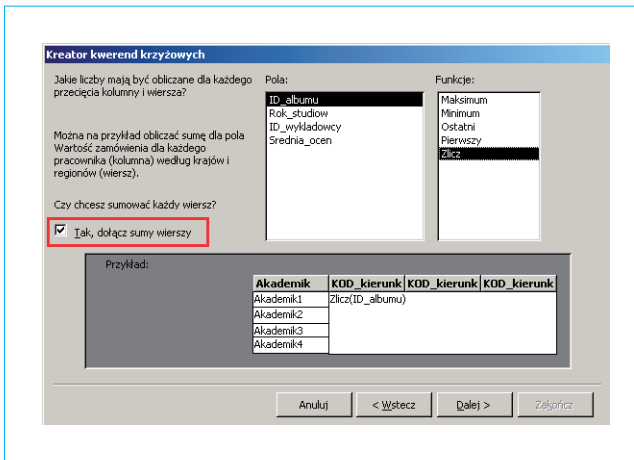
Aby utworzyć kwerendę krzyżową, posłużymy się **Kreatorem kwerend krzyżowych**, który wybierzemy z zestawu kreatorów, po wybraniu przycisku **Nowy** w zakładce z kwerendami, jak na rysunku 2.55.

- ▶ **W pierwszym kroku** kreatora wybierz tabelę, na podstawie której wykonasz zestawienie krzyżowe danych. W naszym przykładzie jest to tabela **Studenci_uczelnia**.
- ▶ **W drugim kroku** kreatora wskaż, które wartości mają być nagłówkami wierszy. W naszym przykładzie będą to symbole akademików.
- ▶ **W trzecim kroku** kreatora ustal, jakie liczby mają być obliczane dla każdego przecięcia kolumny i wiersza. Dokonaj wyboru pola oraz funkcji. Tu możesz zaznaczyć opcję zliczania wszystkich wartości w każdym wierszu. Dla naszego przykładu ustal opcje zgodnie z rysunkiem 2.56.



Rysunek 2.55.

Wybór Kreatora kwerend krzyżowych



Rysunek 2.56.

Krok trzeci kreatora kwerend krzyżowych: wybór pola i funkcji

- ▶ W ostatnim kroku zatwierdź lub podaj nazwę utworzonej kwerendy i zakończ pracę kreatora.

Jak wygląda utworzona kwerenda w widoku **projektu**? Oba widoki, zarówno arkusza danych, jak i projektu, pokazano na rysunku 2.57, gdzie zamieszczone zostały wyjaśnienia.

Studenci_uczelnia_Krzyzowa2 : Kwerenda krzyżowa							
Akademik	Suma ID_albumu	BI	CH	FI	IN	MA	
DS1	23	5	4	4	3	7	
DS2	24	3	3	6	5	7	
▶ DS3	24	8	2	6	6	2	

Pole:	Akademik	KOD_kierunku	ID_albumu	Suma ID_albumu: ID_albumu
Tabela:	Studenci_uczelnia	Studenci_uczelnia	Studenci_uczelnia	Studenci_uczelnia
Podsumowanie:	Grupuj według	Grupuj według	Policz	Policz
Krzyżowe:	Nagłówek wiersza	Nagłówek kolumny	Wartość	Nagłówek wiersza

każde wystąpienie w polu Akademik jest nagłówkiem wiersza	każde wystąpienie w polu KOD_kierunku jest nagłówkiem kolumny	zliczenie wystąpień na skrzyżowaniach kolumn i wierszy	zliczenie wartości w wierszach
---	---	--	--------------------------------

Rysunek 2.57. Kwerenda krzyżowa w widoku arkusza danych i projektu

Z kwerendy krzyżowej korzysta się między innymi przy obliczaniu podsumowań, wartości średnich, zliczaniu wystąpień danych, wybieraniu wartości najmniejszych, największych, pierwszych lub ostatnich. Tak tworzone kombinacje danych, w których jedno dane stanowią nagłówki wierszy, a inne stanowią nagłówki kolumn, nie są możliwe w macierzystej tabeli.

Wykonaj ten projekt ponownie, zamieniając miejscami kolumny z wierszami. Zapisz ją jako **Studenci_uczelnia_krzyzowa_odwrotnie**.



Ćwiczenie 2.28.

Korzystając z kreatora kwerend krzyżowych, zaprojektuj kwerendę, która na podstawie tabeli **Srednia_ocen** utworzy zestawienie średnich ocen na wszystkich kierunkach i rocznikach studiów. Zwróci też najwyższą średnią w każdym roczniku, tak jak to pokazano na rysunku 2.58. Kwerendę zapisz jako **Srednia_ocen_Krzyzowa**.

Srednia_ocen_Krzyzowa : Kwerenda krzyżowa					
	KOD_kierunku	Suma Srednia_ocen	1	2	3
	BI	3,71428571428571	3,5714285714	3,4285714286	3,7142857143
	CH	3,71428571428571	3,2857142857	3,7142857143	3,4285714286
	FI	4,14285714285714	4,1428571429	3,8571428571	3,5714285714
	IN	4,14285714285714	4,1428571429	3,8571428571	3,5714285714
	▶ ma	3,71428571428571	3,7142857143	3,7142857143	3,5714285714

Rysunek 2.58. Wynik działania kwerendy z ćwiczenia 2.28



Ćwiczenie 2.29.

Korzystając z kreatora kwerend krzyżowych, zaprojektuj kwerendę, która na podstawie tabeli **Studenci_uczelnia** utworzy zestawienie liczby studentów studiujących na danym kierunku, z rozbiciem na poszczególne roczniki. Pokaże też sumę studentów na każdym kierunku, tak jak to zaprezentowano na rysunku 2.59. Kwerendę zapisz pod nazwą **Studenci_uczelnia_kierunek_rok_Krzyzowa**.

Studenci_uczelnia_kierunek_rok_Krzyzowa : Kwerenda krzyżow					
	KOD_kierunku	Suma ID_albumu	1	2	3
	BI	16	7	4	5
	CH	9	2	5	2
	FI	16	5	6	5
	IN	14	3	5	6
	▶ MA	16	5	6	5

Rysunek 2.59.

Wynik działania kwerendy z ćwiczenia 2.29



Ćwiczenie 2.30.

Korzystając z kreatora kwerend krzyżowych, zaprojektuj kwerendę, która na podstawie kwerendy **Stypendium_i_kierunek** utworzy zestawienie ilościowe studentów każdego kierunku, rozróżniając wszystkie możliwe kwoty wypłacanego stypendium. Zlicz liczbę studentów z wszystkich kierunków razem dla każdej kwoty, tak jak to pokazano na rysunku 2.60.

Stypendium_Soc	Suma ID_albumu	BI	CH	FI	IN	MA
130,00 zł	4	1	1		1	1
150,00 zł	9		1	2	3	3
180,00 zł	8	2	2	3	1	
200,00 zł	1				1	
230,00 zł	1			1		

Rysunek 2.60.

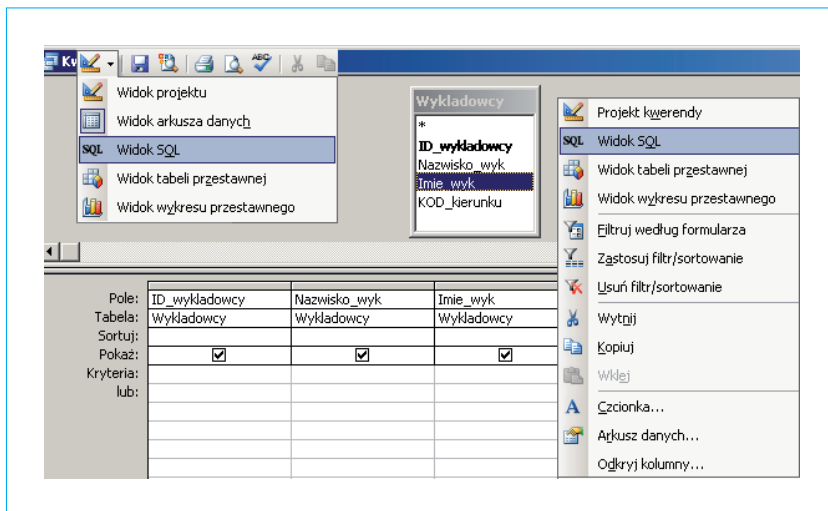
Wynik działania kwerendy z ćwiczenia 2.30

2.4.3. Język zapytań SQL

Zapytania w bazach danych zazwyczaj tworzy się za pomocą języka SQL (jego dialektów). Jednak w MS Access dominującą metodą, ułatwiającą tworzenie kwerend, jest korzystanie z szablonu graficznego, w którym zapytanie tworzone jest za pomocą graficznych elementów przy użyciu myszki, jak to robiliśmy we wszystkich dotychczasowych ćwiczeniach. Należy jednak pamiętać, że jest to tylko forma prezentacji. Zapytanie istnieje w postaci kodu SQL, a niektóre zapytania można stworzyć tylko przy użyciu języka SQL — szablon graficzny jest wtedy nieprzydatny. Każde zapytanie można zapisać bezpośrednio przy użyciu tego języka.

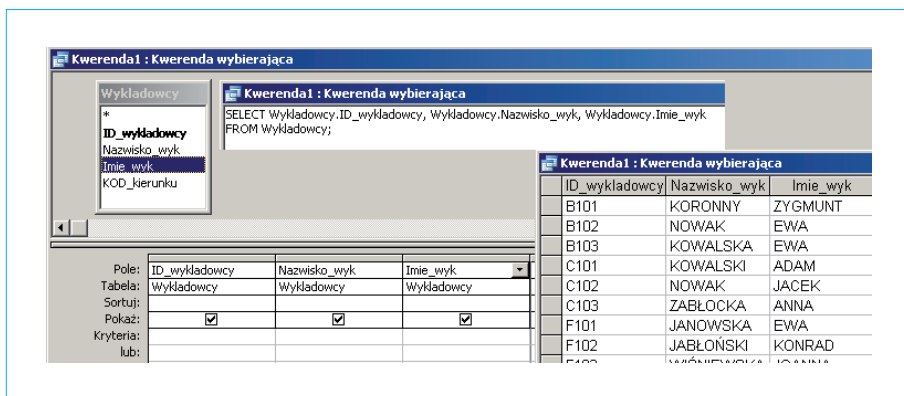
Kwerenda oparta na jednej tabeli

Ponieważ każda kwerenda utworzona przy użyciu graficznego szablonu ma swój zapis w kodzie SQL, zobaczymy, jak wygląda taki kod dla kwerendy wybierającej dane z tabeli **Wykladowcy**. W widoku **projektu** wybierzmy z niej pola z ID oraz imieniem i nazwiskiem wszystkich wykładowców. Teraz zobacz jej kod w języku zapytań. W tym celu kliknij prawym przyciskiem myszy w wolnym obszarze tabel kwerendy. W oknie dialogowym wybierz opcję **Widok SQL**. Ten sam efekt otrzymasz, wybierając **Widok SQL** z przycisku **Widok** na pasku narzędzi, co ilustruje rysunek 2.61.



Rysunek 2.61. Sposoby przejścia z widoku projektu kwerendy do widoku SQL

Po wywołaniu **widoku SQL** pojawi się okno z zapisem kwerendy w języku SQL. Treść kodu w tym oknie możesz czytać i modyfikować. Wszystkie trzy widoki tej wybierającej kwerendy zgromadzone zostały na rysunku 2.62.



Rysunek 2.62. Kwerenda wybierająca w trzech widokach: projektu, arkusza danych i kodu SQL

SQL to język deklaratywny, bardzo wysokiego poziomu, jednak zrozumienie podstaw składni SQL nie będzie stanowić dla Ciebie wielkiego problemu. Omawiana prosta kwerenda wybierająca ma następującą postać:

```
SELECT Studenci_dane.ID_albumu, Studenci_dane.Nazwisko, Studenci_
↳dane.Imie
FROM Studenci_dane;
```

Język SQL jest językiem strukturalnym. Przeanalizujemy treść powyższego kodu.

`SELECT` jest instrukcją służącą do wybierania rekordów i pól z tabeli lub wielu tabel. Klauzula `FROM` wyznacza źródłowe tabele. Podstawowa składnia instrukcji `SELECT` wygląda następująco:

```
SELECT pola FROM tabela;
```

Nazwę pola poprzedzamy nazwą tabeli i kropką. Listę pól oddzielamy przecinkami. Pola zostaną wyświetlone w kwerendzie w takiej kolejności, w jakiej podamy je w zapisie SQL. Jeśli chcesz wyświetlić wszystkie pola, w miejscu `pola` wpisz `*`. Zapis `tabela` oznacza nazwę pojedynczej tabeli lub określa zbiór połączonych tabel (traktowanych dzięki relacjom jak jedna tabela).

W oknie **widoku SQL** dokonaj zmiany, dopisując jeszcze pole z kodem kierunku. Kwerenda będzie więc miała postać:

```
SELECT Wykladowcy.ID_wykladowcy, Wykladowcy.Nazwisko_wyk, Wykladowcy.Imie_wyk, Wykladowcy.KOD_kierunku
FROM Wykladowcy;
```

Równoważny powyższemu jest zapis:

```
SELECT Wykladowcy.*
FROM Wykladowcy;
```

Zamiast wymieniania wszystkich pól z tabeli **Wykladowcy**, użyty został znak `*`.

Kwerenda wybierająca dane z kilku tabel

Zobaczmy, jak wygląda kod SQL dla kwerendy wybierającej dane z dwóch tabel, np. **Studenci_dane** i **Studenci_uczelnia**. W widoku **projektu** wybierzmy z nich pola z ID, imieniem i nazwiskiem oraz kierunkiem studiów. Interesuje nas zapis połączenia tabel relacją. W języku SQL wygląda on następująco:

```
SELECT Studenci_dane.ID_albumu, Studenci_dane.Nazwisko, Studenci_dane.Imie, Studenci_uczelnia.KOD_kierunku
FROM Studenci_dane INNER JOIN Studenci_uczelnia ON Studenci_dane.ID_albumu = Studenci_uczelnia.ID_albumu;
```

Instrukcja `SELECT` jest dla nas już czytelna i jasna. W klauzuli `FROM` zastosowano operator złączenia w celu pobrania danych z dwóch tabel. Do najczęściej stosowanych operatorów należą:

INNER JOIN — złączenie wewnętrzne, wynikające z ustanowionych relacji; w wyniku znajdują się tylko odpowiadające sobie rekordy z obu tabel.

UNION JOIN — złączenie przeciwne do **INNER JOIN**; w wyniku otrzymujemy te rekordy, które do siebie nie pasują.

CROSS JOIN — złączenie krzyżowe; wynik stanowi zestawienie kombinacji wszystkich rekordów z obu tabel (tak jak to miało miejsce w przypadku wyboru danych z tabel, między którymi nie ustanowiono relacji).

Klauzulę `FROM` z naszego przykładu można zatem tłumaczyć następująco: wybierz rekordy występujące w obu tabelach (**Studenci_dane** **INNER JOIN** **Stu-**

denci_uczelnia), których ID_albumu są sobie równe (te pola są bowiem połączone relacją).

Sortowanie

Powróćmy do widoku **projektu** i dodajmy opcję sortowania danych według nazwiska. Kod SQL jest teraz następujący:

```
SELECT Wykladowcy.ID_wykladowcy, Wykladowcy.Nazwisko_wyk, Wykladowcy.  
↳Imie_wyk, Wykladowcy.KOD_kierunku  
FROM Wykladowcy  
ORDER BY Wykladowcy.ID_wykladowcy;
```

Gdy za klauzulą **ORDER BY** (sortuj według) umieścimy zapis **DESC** (ang. *descending* — malejąco), posortujemy nazwiska malejąco:

```
ORDER BY Wykladowcy.ID_wykladowcy DESC;
```

Analogicznie **ASC** (ang. *ascending* — rosnąco) posortuje dane w porządku rosnącym. Brak na końcu klauzuli **ORDER BY** jednej z opcji **ASC/DESC** spowoduje domyślne posortowanie w porządku rosnącym, jak to miało miejsce w naszym przykładzie.

Kryterium wyboru

Kolejnym etapem będzie wprowadzenie kryterium wyboru. Wybierzmy więc z naszej bazy tylko wykładowców informatyki. W oknie projektu dodamy do kwerendy kryterium wyboru **Like "IN"**. Kod SQL ma teraz postać:

```
SELECT Wykladowcy.ID_wykladowcy, Wykladowcy.Nazwisko_wyk, Wykladowcy.  
↳Imie_wyk, Wykladowcy.KOD_kierunku  
FROM Wykladowcy  
WHERE (((Wykladowcy.KOD_kierunku) Like "IN"))  
ORDER BY Wykladowcy.ID_wykladowcy DESC;
```

W celu wprowadzenia ograniczeń kryteriów dla wyświetlanych rekordów zastosowana została klauzula **WHERE**. Za pomocą klauzuli **WHERE** można filtrować rekordy, czyli decydować o tym, które rekordy zostaną włączone do wyniku zapytania. Po słowie **WHERE** podaje się wyrażenie, które musi być prawdziwe, aby rekord pojawił się w wyniku zapytania (został wybrany). Istotne jest zachowanie kolejności, ponieważ klauzula **WHERE** wybiera rekordy z tabel wymienionych wcześniej w klauzuli **FROM**, które spełniają podane kryteria. Bez użycia klauzuli **WHERE** zwracane byłyby wszystkie rekordy tabeli (wszyscy wykładowcy, wszyscy studenci), tak jak w poprzednich przykładach. Klauzula **WHERE** jest więc konieczna, gdy chcemy wprowadzić kryteria wyboru. Składnia kwerendy z kryterium wyboru jest następująca:

```
SELECT pola FROM tabela WHERE wyrażenie;
```

Wyrażenie (predykat) zastosowane po słowie **WHERE** musi być wyrażeniem logicznym. Predykat może być pojedynczym kryterium, alternatywą lub ko-

niunkcją kilku kryteriów wyboru. Najczęściej predykatem jest po prostu kryterium wyboru pól rekordów. Gdy jest ono prawdziwe (spełnione), powoduje wyświetlenie rekordów spełniających to kryterium. Można więc przedstawić składnię kwerendy z określonym kryterium w taki sposób:

```
SELECT pola FROM tabela WHERE predykat;
```

Oto kilka przykładów zastosowania kryteriów w widoku SQL:

Alternatywa kryteriów: wypisz wykładowców, którzy w polu **KOD_kierunku** mają wpis równy *IN* lub *MA*.

```
WHERE (((Wykladowcy.KOD_kierunku) Like "IN" Or (Wykladowcy.KOD_
↳kierunku) Like "MA"));
```

Koniunkcja kryteriów: wypisz ID studentów trzeciego roku informatyki.

```
SELECT Studenci_uczelnia.ID_albumu
FROM Studenci_uczelnia
WHERE (((Studenci_uczelnia.Rok_studiow)=3) AND ((Studenci_uczelnia.
↳KOD_kierunku) Like "IN"));
```

Alternatywa i koniunkcja: wypisz ID studentów trzeciego roku informatyki i matematyki.

```
WHERE (((Studenci_uczelnia.Rok_studiow)=3) AND ((Studenci_uczelnia.
↳KOD_kierunku) Like "IN" Or (Studenci_uczelnia.KOD_kierunku) Like
↳"MA"));
```

Kryterium ograniczone podanymi wartościami:

```
WHERE (((Studenci_uczelnia.Srednia_ocen)>=3 And (Studenci_uczelnia.
↳Srednia_ocen)<=4));
```

Krótszy zapis powyższej koniunkcji z podaniem zakresów:

```
WHERE (((Studenci_uczelnia.Srednia_ocen) Between 3 And 4));
```

Pola obliczeniowe

Na przykładzie kwerendy obliczającej nową stawkę stypendium socjalnego przeanalizujmy zasadę tworzenia nowego pola, którego wartości są wynikiem obliczeń. Do kwoty stypendium dodajemy każdemu 10 złotych. Oczywiście dodajemy tylko wtedy, gdy student stypendium otrzymuje. Oto zapis kwerendy:

```
SELECT Studenci_dane.ID_albumu, Studenci_dane.Stypendium_Soc, [Sty
↳pendium_Soc]+10 AS Nowa_stawka
FROM Studenci_dane
WHERE (((Studenci_dane.Stypendium_Soc) Is Not Null));
```

W wykazie pól pojawiło się nowe pole, powstałe z wyrażenia, które do pola **Stypendium_stawka** dodaje wartość 10. *AS* oznacza, że w widoku **arkusza danych** wyświetlona zostanie kolumna (pole) o podanej nazwie **Nowa_stawka**. Zwróć również uwagę na wprowadzone kryterium **WHERE ... Is Not Null**.

Kwerendy parametryczne

Parametry podajemy w kryteriach, więc nie będzie dla nikogo zaskoczeniem, że w kodzie SQL parametr z treścią komunikatu **podaj kod kierunku** pojawi się w nawiasach kwadratowych w klauzuli **WHERE**.

```
SELECT Wykladowcy.ID_wykladowcy, Wykladowcy.Nazwisko_wyk, Wykladowcy.
↳KOD_kierunku
FROM Wykladowcy
WHERE (((Wykladowcy.KOD_kierunku)=[podaj kod kierunku]));
```

Jednak jeśli chcesz użyć kilku parametrów, do każdego z nich, nawet jeśli dotyczy tego samego pola, musisz podać nazwę tabeli, nazwę pola i — opcjonalnie — treść komunikatu. Rozbudowując więc poprzedni przykład, tak aby podać dwie wartości parametrów przy uruchamianiu kwerendy, musisz napisać:

```
WHERE (((Wykladowcy.KOD_kierunku)=[podaj kod kierunku] Or (Wykladowcy.
↳KOD_kierunku)=[podaj kod drugiego kierunku]));
```

Kwerenda ta wypisze dane wykładowców z dwóch kierunków, przy czym konkretne kierunki podaje użytkownik (przy uruchomieniu kwerendy).

Kwerenda tworząca nową tabelę

Kod SQL kwerendy tworzącej tabelę różni się od kodu kwerendy wybierającej obecnością klauzuli **INTO** informującej, jak ma się nazywać nowa, powstała po uruchomieniu kwerendy tabela. Zapis kodu kwerendy tworzącej z tabeli **Studenci_uczelnia** nową tabelę, zawierającą dwie wybrane kolumny danych, wygląda zatem następująco:

```
SELECT Studenci_uczelnia.ID_albumu, Studenci_uczelnia.KOD_kierunku
↳INTO [Studenci ID+KOD]
FROM Studenci_uczelnia;
```

Kwerenda krzyżowa

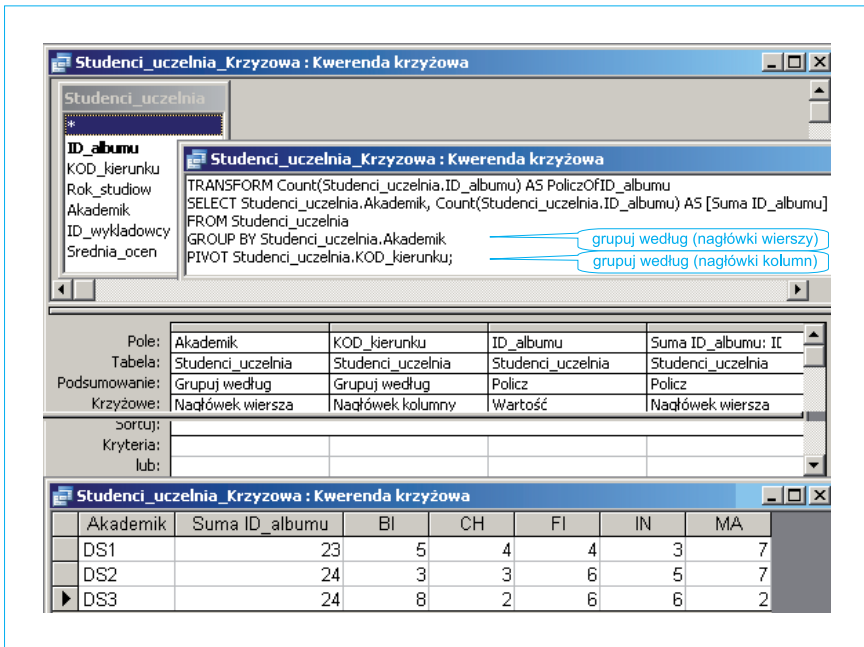
Jako przykład do analizy niech nam posłuży utworzona wcześniej kwerenda krzyżowa **Studenci_uczelnia_Krzyzowa**. Kwerendę tę tworzyliśmy w oknie kreatora kwerend krzyżowych i samo przełączenie do widoku **projektu** może wywołać lekkie zaskoczenie. Na rysunku 2.63 zestawiono wszystkie trzy widoki tej skomplikowanej kwerendy: widok **projektu**, **arkusza danych** i **SQL**.

Każdą linię kodu połączono z odpowiednim miejscem w projekcie i w tabeli wynikowych danych. Powstała w ten sposób kolorowa pajęczyna, której przyjrzymy się uważniej.

```
TRANSFORM Count(Studenci_uczelnia.ID_albumu) AS PoliczOfID_albumu
```

Operacje zliczenia (**policzOf**) będą wykonywane na podstawie pola **ID_albumu** z tabeli **Studenci_uczelnia**.

```
SELECT Studenci_uczelnia.Akademik, Count(Studenci_uczelnia.ID_albumu)
↳mu) AS [Suma ID_albumu]
```



Rysunek 2.63. Kwerenda krzyżowa w trzech widokach: projektu, arkusza danych i SQL

W kolumnie, która pojawi się jako **Suma ID_albumu**, nastąpi zliczenie wszystkich wartości dla każdego wiersza **Akademik**. Dla każdego wystąpienia w polu **Akademik** nastąpi zliczenie identyfikatorów w polu **ID_albumu**.

FROM Studenci_uczelnia

To jest chyba jasne.

GROUP BY Studenci_uczelnia.Akademik

Grupuj według wystąpień w polu **Akademik**. Każde wystąpienie będzie nowym wierszem (**DS1**, **DS2**, **DS3**).

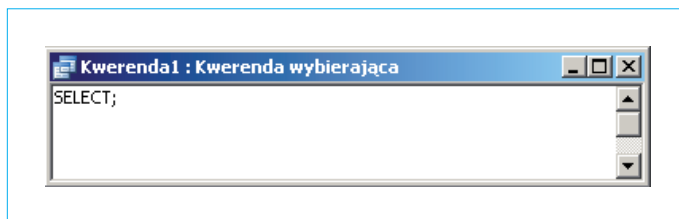
PIVOT Studenci_uczelnia.KOD_kierunku;

Grupuj według wystąpień w polu **KOD_kierunku**. Każde wystąpienie będzie nową kolumną (**BI**, **CH**, **FI**, **IN**, **MA**).

Tworzenie kwerend za pomocą instrukcji języka SQL

Przejdźmy do tworzenia kwerend za pomocą instrukcji języka SQL.

W celu utworzenia od podstaw kwerendy za pomocą instrukcji języka SQL należy w widoku **projektu** nowej kwerendy zamknąć pojawiające się standardowo okno **Pokazywanie tabel**, a następnie przejść do widoku SQL kwerendy. Zostanie otwarte okno, jak na rysunku 2.64, w którym zdefiniujemy kwerendę za pomocą instrukcji języka SQL. W oknie wpisane jest tylko słowo **SELECT** i średnik.

**Rysunek 2.64.**

Okno tworzenia kwerendy w widoku kodu SQL

Po napisaniu kwerendy możesz ją uruchomić lub otworzyć w widoku **projektu**. Zmienimy więc kolejność tworzenia zapytania. Najpierw będziemy pisać kwerendy w języku SQL, a potem obserwować efekty w widoku **projektu** kwerendy.



Ćwiczenie 2.31.

Utwórz zapytanie prezentujące na podstawie tabeli **Plan_zajec** wykaz studentów (ich ID) wybranego wykładowcy. Efekt działania kwerendy powinien być taki jak na rysunku 2.65, na którym wyświetlona jest lista studentów wykładowcy **B102**.

ID_wykladowcy	ID_albumu
B102	WU00008
B102	WU00011
B102	WU00010

Rysunek 2.65.

Lista studentów wykładowcy B102

W pierwszej linii wpisz nazwy pól, oddzielone przecinkiem. Pamiętaj o składni:

```
SELECT tabela.pole1, tabela.pole2
```

Określ, skąd mają być pobierane dane:

```
FROM tabela
```

Ustal kryterium wyboru. W miejscu kryterium wpisz parametr:

```
WHERE (((tabela.pole)=[komunikat dla parametru]));
```

Oto gotowa treść kwerendy zapisana w języku SQL. Porównaj ją ze swoim rozwiązaniem:

```
SELECT Plan_zajec.ID_wykladowcy, Plan_zajec.ID_albumu
FROM Plan_zajec
WHERE (((Plan_zajec.ID_wykladowcy)=[podaj ID wykładowcy]));
```



Ćwiczenie 2.32.

Utwórz zapytanie wybierające studentów (z podaniem ich imienia i nazwiska), którzy uzyskali z matematyki ocenę większą od 3. Wynik działania kwerendy ilustruje rysunek 2.66.

Imie	Nazwisko	Matematyka
ADRIANNA	MAKOWSKA	4
ALEKSANDRA	KWIATKOWSKA	5
MICHAŁ	MIKOŁAJEWICZ	5
JAROSŁAW	KOKIEWICZ	4
SANDRA	SURMA	4

Rysunek 2.66.

46 rekordów spełniających kryterium z ćwiczenia 2.32

Jakie pola musisz wybrać do działania tej kwerendy? Podaj nazwy pól poprzedzone kropką i nazwą tabeli.

Określ, skąd mają być pobierane dane. Pamiętaj, że te dwie tabele są połączone relacją. Ogólna składnia jest następująca:

```
FROM tabela1 INNER JOIN tabela2 ON tabela1.pole = tabela2.pole
```

Ustal kryterium wyboru. W miejscu kryterium wpisz parametr:

```
WHERE (((tabela.pole)kryterium));
```

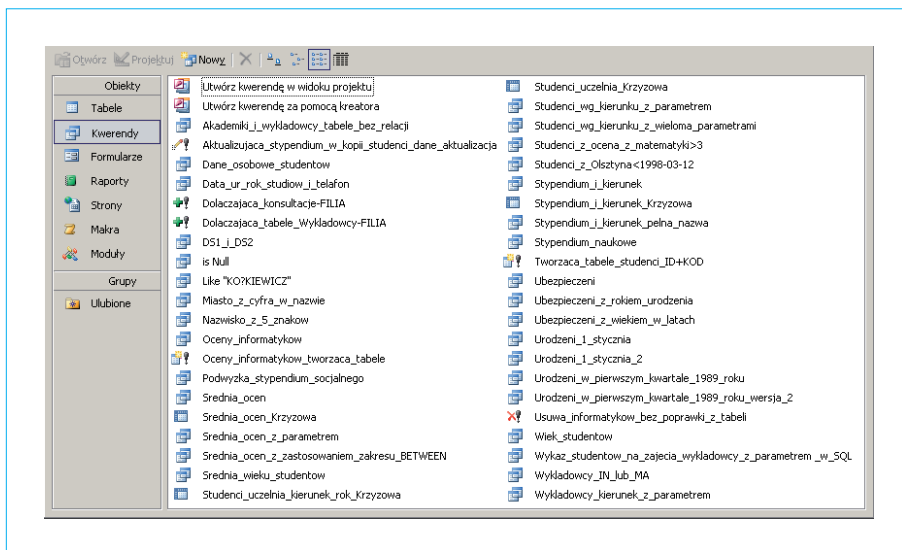
Oto gotowa treść kwerendy zapisana w języku SQL. Porównaj ją ze swoim rozwiązaniem:

```
SELECT Studenci_dane.Imie, Studenci_dane.Nazwisko, Studenci_oceny.
↳Matematyka
FROM Studenci_dane INNER JOIN Studenci_oceny ON Studenci_dane.
↳ID_albumu = Studenci_oceny.ID_albumu
WHERE (((Studenci_oceny.Matematyka)>3));
```



Warunki zapisywane w podwójnych nawiasach (()) nie są warunkami języka SQL, lecz warunkami języka wyrażeń Office Access (ściślej — języka Visual Basic for Applications). Może to mieć znaczenie, gdybyś kiedyś korzystał z Accessa w bardziej zaawansowany sposób, np. w połączeniu z bazą SQL Server, gdzie Access służy do wprowadzania danych przez wielu użytkowników i wykorzystywany jest jedynie jako tzw. „front” do bazy. Wtedy tego typu wyrażeń nie można stosować.

Po wykonaniu wszystkich ćwiczeń z punktu „Kwerendy” w zakładce z kwerendami powinny pojawić się kwerendy pokazane na rysunku 2.67.



Rysunek 2.67. Lista kwerend utworzonych w ramach ćwiczeń z punktu „Kwerendy”