

DO NOWEJ PODSTAWY
PROGRAMOWEJ

Część 2

PODRĘCZNIK dla szkół ponadgimnazjalnych

Informatyka

Europejszczyka

Zakres **rozszerzony**



Zawiera CD

Danuta Korman

 **Helion**
EDUKACJA

Podręcznik dopuszczony do użytku szkolnego przez ministra właściwego do spraw oświaty i wychowania i wpisany do wykazu podręczników przeznaczonych do kształcenia ogólnego do nauczania informatyki, na podstawie opinii rzeczoznawców: mgr. inż. Włodzimierza Kruszwickiego, dr. Leszka Rudaka, dr. Iwony Wandy Grygiel.

Zakres kształcenia: rozszerzony.

Etap edukacyjny: IV.

Typ szkoły: szkoły ponadgimnazjalne.

Rok dopuszczenia 2012.

Numer ewidencyjny w wykazie: 410/2/2012

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Joanna Zaręba

Projekt okładki: ULABUKA

Skład: Ewa Galczak

Ilustracja na okładce została wykorzystana za zgodą Shutterstock.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie?iepp22>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-246-2824-7

Copyright © Helion 2013

Wydanie II

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to!» Nasza społeczność](#)

..... Spis treści

Wstęp	6
Rozdział 1. Bazodanowe operacje arkusza kalkulacyjnego	9
1.1. Jednotabelowa baza danych — kartoteka	9
1.2. Podstawowe pojęcia	11
1.3. Tworzenie nowej bazy i import danych	12
1.4. Wyszukiwanie i organizowanie danych	14
1.4.1. Wyszukiwanie z wykorzystaniem formularza	14
1.4.2. Wyszukiwanie informacji za pomocą autofiltrowania	14
1.4.3. Wyszukiwanie informacji za pomocą filtru zaawansowanego	20
1.4.4. Sortowanie danych	23
1.4.5. Sumy częściowe (pośrednie)	25
1.4.6. Statystyki oparte na raporcie tabeli przestawnej	27
1.4.7. Zadania	30
Rozdział 2. Relacyjna baza danych	35
2.1. Bazy danych i systemy nimi zarządzające	35
2.2. Projektowanie	38
2.2.1. Zasady projektowania bazy danych	38
2.2.2. Projektowanie bazy danych w programie Microsoft Office Access 2007	43
2.3. Tabele	44
2.3.1. Tworzenie tabeli w programie Access 2007	44
2.3.2. Typy danych w tabelach	47
2.3.3. Podstawowe właściwości pól tabeli	50
2.3.4. Pole kluczowe tabeli	55
2.3.5. Definiowanie relacji	57
2.3.6. Operacje na tabelach bazy danych	63
2.3.7. Zmiana sposobu prezentowania danych	65
2.3.8. Import, eksport, załączanie tabeli	70
2.4. Kwerendy — zapytania	74
2.4.1. Kwerendy wybierające	75
2.4.2. Kwerendy funkcjonalne — modyfikujące	88
2.4.3. Język zapytań SQL	95
2.5. Formularze	103
2.5.1. Autoformularze	103
2.5.2. Projektowanie formularzy	104

2.6. Raporty	108
2.7. Makropolecenia	110
2.8. Ochrona i bezpieczeństwo danych	112
2.8.1. Ochrona bazy danych hasłem	113
2.8.2. Kodowanie i dekodowanie bazy danych	114
2.9. Zadania	114
Rozdział 3. Systemy operacyjne i sieci komputerowe	117
3.1. Budowa komputera	117
3.2. BIOS i rozruch komputera	122
3.3. System operacyjny	124
3.3.1. Budowa systemu operacyjnego	124
3.3.2. Rola systemu operacyjnego	126
3.3.3. Maszyna wirtualna	131
3.3.4. Zadania	142
3.4. Sieci komputerowe	143
3.4.1. Podział sieci komputerowych	144
3.4.2. Model OSI	157
3.4.3. Model protokołu TCP/IP	162
3.4.4. Adres IP (ang. <i>Internet Protocol address</i>)	164
3.4.5. Podstawowe rodziny protokołów	177
3.4.6. Obsługa i konfiguracja sieci w Windows	182
3.4.7. Ogólne zasady administrowania siecią komputerową w architekturze klient-serwer	182
3.4.8. Bezpieczeństwo sieci	183
3.4.9. Zadania	191
3.5. Podstawy tworzenia stron WWW	192
3.5.1. Podstawowe elementy usługi WWW	193
3.5.2. Projektowanie witryny WWW	193
3.5.3. Popularne formaty hipertekstowe w sieci WWW	193
3.5.4. Struktura dokumentu	196
3.5.5. Kaskadowe arkusze stylów	202
3.5.6. Publikowanie witryny w sieci WWW	206
3.5.7. Systemy zarządzania treścią	207
3.5.8. Czym jest PHP?	210
3.5.9. Zadania	221

Rozdział 4. Multimedia i grafika komputerowa	225
4.1. Grafika komputerowa	225
4.1.1. Grafika rastrowa	226
4.1.2. Rozdzielczość a wymiary obrazka	226
4.1.3. Reprezentacja obrazu w komputerze	230
4.1.4. Istota obrazu cyfrowego	239
4.1.5. Cyfrowy obraz ze skanera i cyfrowego aparatu fotograficznego	240
4.1.6. Kompresja stratna i bezstratna	240
4.1.7. Formaty plików bitmapowych	247
4.1.8. Grafika na stronach WWW	249
4.1.9. Grafika wektorowa	253
4.1.10. Grafika bitmapowa a wektorowa	257
4.1.11. Grafika trójwymiarowa — 3D	258
4.1.12. Zadania	266
4.2. Dźwięk	269
4.2.1. Teoria dźwięku	269
4.2.2. Zapis dźwięku	271
4.2.3. Formaty plików dźwiękowych	274
4.2.4. Zadania	281
4.3. Wideo	281
4.3.1. Cyfrowy zapis filmu	282
4.3.2. Montaż filmu i jego publikacja	284
4.4. Zadania sprawdzające do rozdziału 4.	285
Rozdział 5. Kierunki rozwoju technologii informatycznych oraz aspekty etyczne, prawne i społeczne w zastosowaniach informatyki	291
5.1. Kierunki rozwoju informatyki i jej zastosowań	291
5.1.1. Krótkie kalendarium wydarzeń ważnych dla rozwoju komputeryzacji	292
5.1.2. Co przyniesie najbliższa przyszłość?	293
5.1.3. Zadania	299
5.2. Aspekty etyczne, prawne i społeczne w zastosowaniach informatyki	300
5.2.1. Prawo autorskie	300
5.2.2. Rodzaje licencji programów komputerowych	300
5.2.3. Wolne licencje	301
5.2.4. Ustawa o prawie autorskim a szkolny referat	302
5.2.5. Zadania	303
Skorowidz	305

..... 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 potrzebnych informacji. Program Access udostępnia dwa języki definiowania kwerend: język *SQL* (ang. *Structured Query Language*) oraz język *QBE* (ang. *Query By Example*).


Jeśli więc chcemy znaleźć i pobrać dane, które spełniają pewne warunki, tworzymy *kwerendę*, czasem nazywaną zapytaniem. Kwerenda może obejmować dane pochodzące z różnych tabel lub z wcześniej utworzonych kwerend (bądź jednych i drugich), wyszukiwać dane, aktualizować lub usuwać wiele rekordów jednocześnie, tworzyć nowe tabele. 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 pochodzących z Olsztyna, urodzonych przed 12 marca 1989 roku. Przejdźmy więc do zakładki z kwerendami.

Po dwukrotnym kliknięciu myszą nazwy kwerendy następuje jej uruchomienie. Klawisz *Enter* lub polecenie *Otwórz* z menu kontekstowego tabeli również uruchamia kwerendy. W przypadku kwerendy *Studenci_z_Olsztyna <1989-03-12* nastąpi otwarcie kwerendy w postaci arkusza (tabelki) z danymi wyszukanyymi 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*, wybierając polecenie *Widok projektu...* z menu kontekstowego kwerendy. Po takim otwarciu zobaczymy na Wstążce

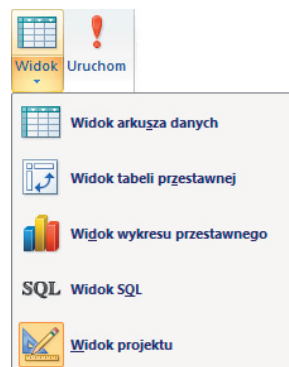
znany zestaw poleceń zaawansowanego filtrowania danych — wybieraliśmy tu wcześniej pola i kryteria do filtrowania. Ta zakładka, *Projektowanie*, jest narzędziem kwerend.

Aby w widoku *projektu* uruchomić kwerendę, należy kliknąć przycisk . Możliwe jest również wybranie polecenia *Uruchom* przy użyciu sekwencji następujących klawiszy: lewy *Alt* (który uaktywnia skróty klawiaturowe), następnie litera *R* (pozwala przejść do zakładki *Projektowanie*) i na koniec litera *H*.

Kwerendy można oglądać jeszcze w innych widokach, pokazanych na rysunku 2.32.

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.



Rysunek 2.32. Dostępne widoki kwerend

Przeanalizujmy podstawowe typy kwerend na przykładach.

2.4.1. Kwerendy wybierające

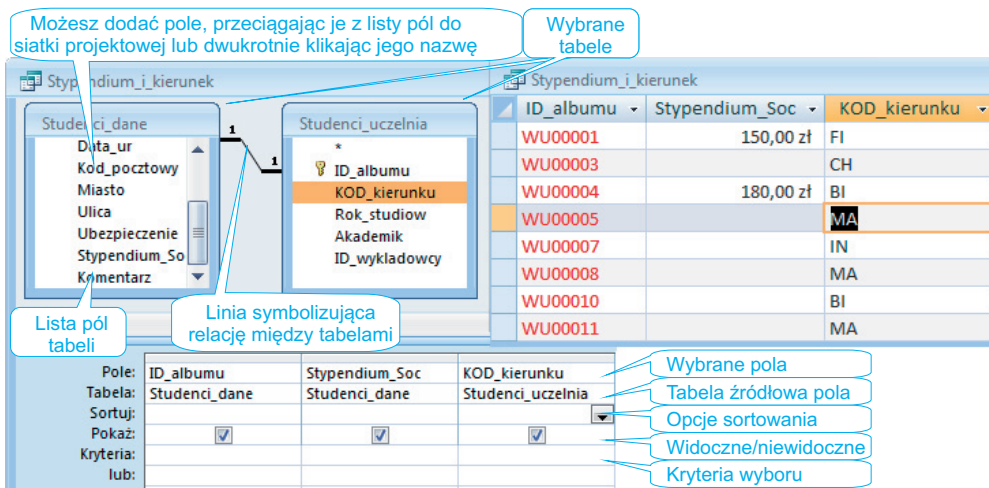
Najczęściej spotykanym rodzajem kwerendy jest kwerenda wybierająca. Pobiera ona dane z jednej tabeli lub z większej liczby tabel bądź z wcześniej zdefiniowanych i zapisanych kwerend, używając podanych kryteriów, a następnie wyświetla wybrane dane w żądanym porządku.

Kwerendę można utworzyć, używając kreatora kwerend lub samodzielnie zaprojektować ją w widoku projektu kwerendy. Oba narzędzia zamieszczono na zakładce *Tworzenie*, w grupie *Inne*.

W widoku projektu kwerendy przez dodanie do siatki projektowej pól zawierających dane z jednej tabeli lub z 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 zilustrowano na rysunku 2.33. Pomoże on nam w wykonaniu ćwiczenia 2.12.

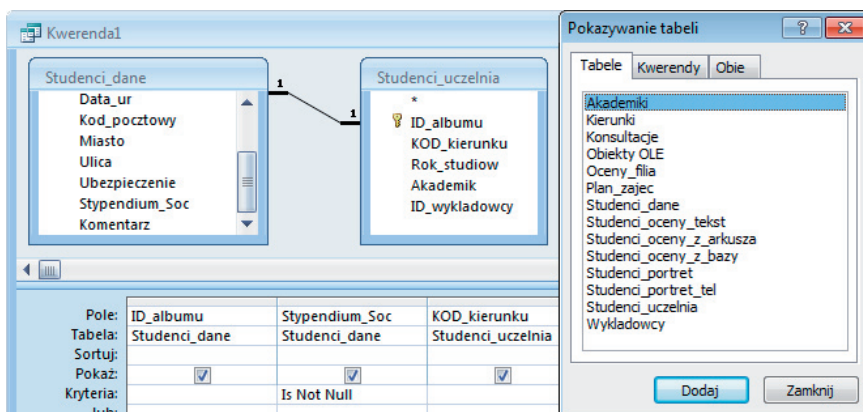
Ćwiczenie 2.12. Zaprojektuj kwerendę wybierającą studentów, którzy otrzymują stypendium. Obok numeru albumu wyświetl kierunek studiów oraz kwotę pobieranych świadczeń. Wybrane dane posortuj rosnąco według numeru indeksu.

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 z grupy *Inne* na karcie *Tworzenie*



Rysunek 2.33. Kwerenda z kryteriami wyboru wybierająca dane z dwóch tabel. Z lewej widok projektu, z prawej — arkusza danych

wybrać polecenie *Projekt kwerendy*. W oknie projektu kwerendy pojawi się, jak na rysunku 2.34, 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 każde z nich. 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 *Uruchom*. 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*.



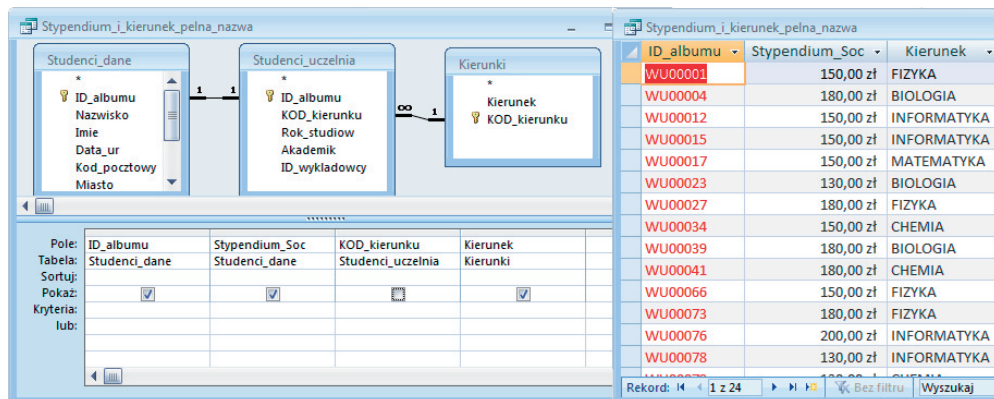
Rysunek 2.34. Okno wyboru tabel do projektu kwerendy

Gdyby wszyscy studenci pobierali stypendium w jednakowej wysokości, w kryteriach wyboru pola *Stypendium_Soc* można by podać tę wartość, na przykład **150**. Jednak po uruchomieniu kwerendy zniknęłyby z zestawu nie tylko studenci bez stypendium, ale również ci, którzy pobierają stypendium inne niż 150 złotych. Ponieważ jest wiele róż-

nych kwot, korzystnym w tym przypadku 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 kwerendę jako *Stypendium_i_kierunek* i uruchom ją. Powinien zostać wyświetlony wybrany zestaw danych tylko tych studentów, którzy pobierają stypendium.

Ćwiczenie 2.13. Zmodyfikuj utworzoną w ćwiczeniu 2.12 kwerendę wybierającą tak, aby zamiast kodu kierunku wypisywała pełną nazwę kierunków studiów każdego studenta pobierającego stypendium socjalne. Zmodyfikowaną kwerendę zapisz jako nową kwerendę i nadaj jej nazwę *Stypendium_i_kierunek_pelna_nazwa*.

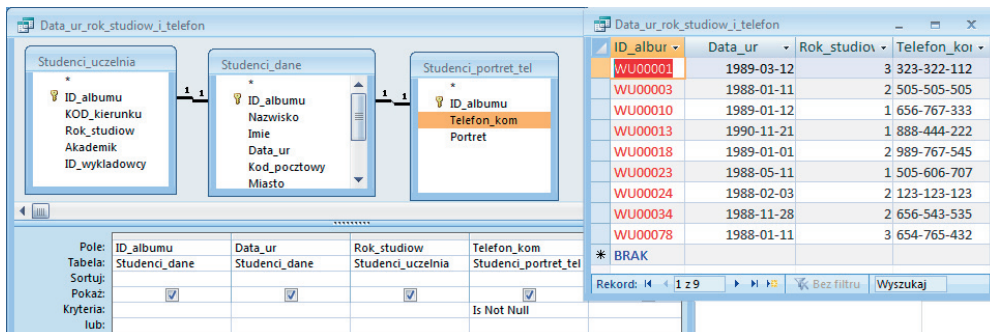
Do wykonania tego ćwiczenia użyj utworzonej wcześniej kwerendy *Stypendium_i_kierunek*. Otwórz ją w widoku *projektu*. Aby wyświetlić pełne nazwy kierunków zamiast ich kodów, potrzebna jest tabela *Kierunki*. Dołącz ją do tabel. W tym celu przyciskiem *Pokaż tabelę* (polecenie dostępne również w menu kontekstowym) otwórz listę dostępnych tabel i dodaj do projektu kwerendy tabelę *Kierunki*. Następnie pole *Kierunek* dodaj do siatki projektowej. Zapisz utworzoną kwerendę (*Przycisk pakietu Office/Zapisz jako...*) pod nazwą *Stypendium_i_kierunek_pelna_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 2.35.



Rysunek 2.35. Okno projektowe korygowanej kwerendy i dynamiczny zestaw danych bez ukrytego pola *KOD_kierunku*

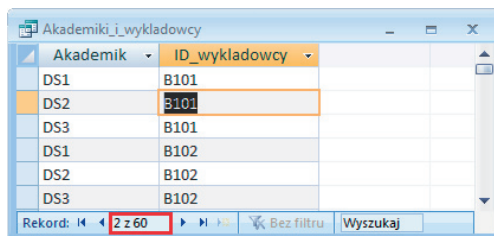
Przyjrzyj się uważnie utworzonej siatce projektowej oraz relacjom między tabelami danych źródłowymi użytymi w kwerendzie. Zauważ, że tabela *Kierunki* jest sprzężona z tabelą *Studenci_uczelnia* przez pole *KOD_kierunku*. Nie ma więc potrzeby zamieszczania pola *KOD_kierunku* w siatce projektowej. Jeśli 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 o studentach posiadających telefon. Dane wybierz z trzech tabel, jak na rysunku 2.36. Wybranych studentów posortuj w kolejności od najmłodszego do najstarszego. Utworzoną kwerendę wybierającą zapisz pod nazwą *Data_ur_rok_studiow_i_telefon*. Który student jest ostatni na liście? Ilu studentów podało swoje numery telefonów?



Rysunek 2.36. Kwerenda wybierająca dane z trzech tabel do ćwiczenia 2.14

Przyjrzyj się zawartości dwóch tabel: *Akademiki* i *Wykladowcy*. Zawierają one, odpowiednio: 3 rekordy i 20 rekordów. Między nimi nie ma ustanowionej relacji. Co się stanie, jeśli utworzysz 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 2.37.



Rysunek 2.37. 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 pola 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". Posortuj wybrane dane według nazwisk. Zapisz utworzoną kwerendę jako *Oceny_informatykw*. Uruchom kwerendę i sprawdź jej działanie. Wyłącz pokazywanie kodu kierunku. Zobacz efekt działania kwerendy. Zauważ, że informatycy nie mają ocen ze wszystkich przedmiotów. Usuń z projektu te przedmioty nauczania. Aby zaznaczyć kolumnę (pole), ustaw wskaźnik myszy nad nazwą pola — przybierze on kształt czarnej strzałki. Zaznacz pole i usuń je z projektu kwerendy (*Delete*).

Dobrze byłoby również ująć w tym wykazie numer ID oraz rok studiów danego studenta, co wiąże się z dodaniem trzeciej tabeli, *Studenci_uczelnia*, w której znajduje się informacja o roku studiów. Uzupełnij to w widoku *projektu* kwerendy. Zapisz wprowadzone zmiany.

Po uruchomieniu kwerendy możesz na wyświetlonym, dynamicznym zestawie danych zastosować filtry. Którzy studenci trzeciego roku otrzymali najniższą ocenę z matematyki? Ustaw kursor w polu, w którym jest ocena 3 z matematyki, i zastosuj filtr dla zaznaczenia *Równa się 3*. Następnie zastosuj filtr dla pola *Rok_studiow*. Fragment zestawu przefiltrowanych w ten sposób danych tej kwerendy znajduje się na rysunku 2.38.

ID_albur	Nazwisko	Imie	Rok_studiow	Matematyka	Informatyka	Jezyk_angi
WU00015	KOŁACZYŃSKA	ROKSANA	3	3	3	5
WU00104	SIEKIERSKI	JAKUB	3	3	3	2
WU00076	ZAKRZEWSKA	HELENA	3	3	3	4
* BRAK						

Rysunek 2.38. Fragment zestawu przefiltrowanych danych kwerendy wybierającej oceny studentów informatyki

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 studenta 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 pod nazwą *Ubezpieczeni*.

W kryteriach tekstowych można stosować *znaki wieloznaczne* (zastępcze), które zastępują wiele znaków lub jeden znak. Należą do nich *, ? i #. Znak * 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 Access 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 kryte-

rium dla pola *Imie* do wyboru płci żeńskiej, zakładając, że zwykle w Polsce imiona kobiet kończą się na literę *a* (choć nie jest to regułą — coraz częściej nadawane są imiona obcego pochodzenia, na przykład imię żeńskie Inez nie spełnia tego warunku, a imię męskie Bonawentura jest błędnie traktowane jako imię żeńskie).

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 1 stycznia w latach 1900-1999. 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, na przykład *LUBICZ1*.

W kryteriach wyboru można również podać zakres *[]*, w jakim musi znaleźć się poszukiwany rekord. Wpis *Like "[A-D]*"* oznacza, że wszystkie wybrane nazwy będą rozpoczynały się jedną z liter z podanego zakresu, więc *A*, *B*, *C* lub *D*. Wpis *Like "[krs]"* oznacza, że wszystkie wybrane nazwy będą kończyły się jedną z liter podanych w zakresie, czyli *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 (na przykład "*[!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: *>*, *>=*, *<*, *<=*, *=*, *<>* (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>**.

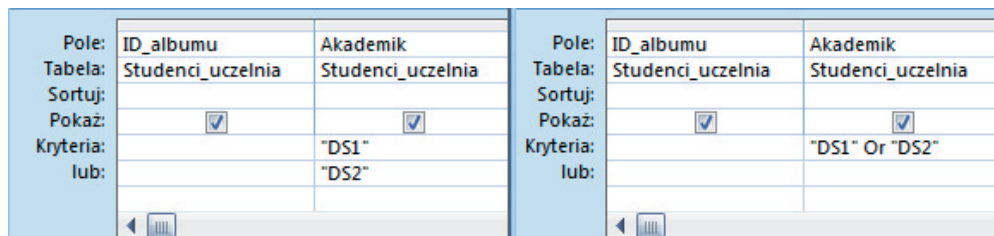
Wielkość liter występujących w kryteriach nie ma znaczenia.

Ćwiczenie 2.17. W naszej przykładowej bazie danych utwórz dwie kwerendy wybierające. Pierwsza ma wybierać z tabeli *Studenci_dane* studentów urodzonych w pierwszym kwartale 1989 roku. Wykaz ma zawierać ID, imiona i nazwiska stu-

dentów. Na początku listy zamieść najstarszego studenta (*Zakrzewska*), a na końcu najmłodszego (*Olek*).

Datę w kryterium można wpisać w sposób następujący: **>=1989-01-01 And <=1989-03-31**. Po uruchomieniu kwerendy data automatycznie zostanie ujęta w znacznik **#data#** i kryterium przyjmie następującą postać: **>=#1989-01-01# And <=#1989-03-31#**. Utworzoną kwerendę zapisz pod nazwą *Urodzeni_w_pierwszym_kwartale_1989_roku*.

W następnej kwerendzie wybierz z tabeli *Studenci_uczelnia* 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 2.39. Kwerendę zapisz pod nazwą *Studenci_z_DS1_i_DS2*.



Rysunek 2.39. Alternatywne 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 zobrazowano na rysunku 2.40;

Pole:	ID_albumu	Rok_studiow	Data_ur	Data_ur
Tabela:	Studenci_dane	Studenci_uczelnia	Studenci_dane	Studenci_dane
Sortuj:			Rosnąco	
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Kryteria:		3	>=#1989-01-01#	<=#1989-03-31#
lub:				

Koniunkcja kryteriów

Wyłączenie widoczności dublowanego pola Data_ur

Rysunek 2.40. Koniunkcja wyboru studentów trzeciego roku urodzonych w pierwszym kwartale 1989 roku

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

- 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**.

Definiowanie wyrażeń

Wyrażenie to dowolna kombinacja operatorów matematycznych i logicznych, stałych, funkcji, nazw pól, formantów i właściwości, której wynikiem jest pojedyncza wartość. Wyrażenia służą do przeprowadzania obliczeń, operowania znakami lub testowania danych.

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 je 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 *Imie* i *Nazwisko* oddzielone pojedynczą spacją. Powstałą kwerendę zapisz pod nazwą *Dane_osobowe_studentow*.

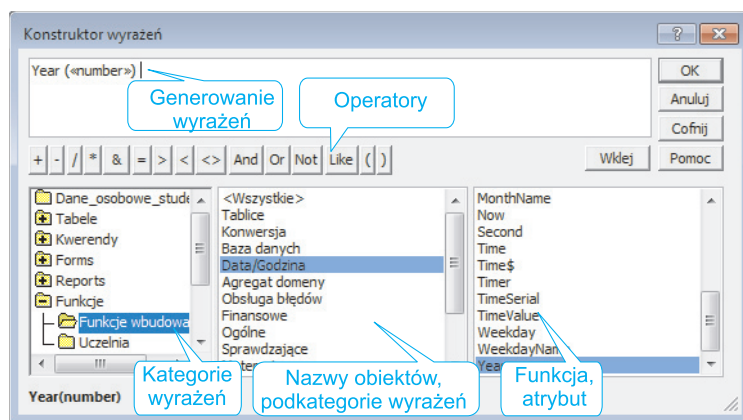
W oknie projektu nowej kwerendy dodaj tabelę *Studenci_dane*. Wybierz z niej trzy pola: *ID_albumu*, *Nazwisko* i *Imie*. W nowym polu siatki projektowej wpisz następujące wyrażenie: `[Imie] & " " & [Nazwisko]` i uruchom powstałą kwerendę. Zauważ, jak jest nazwane pole, które połączyło dwa pola tekstowe. Powróć do projektu. Aby lepiej widzieć powstałe wyrażenie, wybierz polecenie *Powiększenie...* i w otwartym okienku edycji wyrażenia zmień nazwę `Wyr1` na `Imie_i_nazwisko` (nie usuwaj dwukropka). Teraz wyrażenie

ma następującą postać: `Imie_i_nazwisko: [Imie] & " " & [Nazwisko]`. Posortuj listę według nazwisk i wyłącz pokazywanie nazwiska i imienia w osobnych polach. Ponownie uruchom kwerendę. Poszerz kolumnę, aby można było przeczytać dane połączone w jednym polu. Zapisz kwerendę pod nazwą *Dane_osobowe_studentow*.

Aby widzieć całe wyrażenie, ustaw kursor w polu wyrażenia i wybierz polecenie *Powiększenie...* (*Shift+F2*).

W tworzeniu wyrażeń dużą pomoc oferuje *Konstruktor wyrażeń*, uruchamiany poleceniem *Konstruktor* znajdującym się w grupie narzędzi *Konfiguracja kwerendy*. Można go również uruchomić, wybierając opcję *Konstruuuj...* z menu kontekstowego

pustego pola lub pola z już istniejącym wyrażeniem. Polecenie otwiera okno *Konstruktor wyrażeń*, pokazane na rysunku 2.41.



Rysunek 2.41. Okno Konstruktor wyrażeń. Konstruowanie wyrażenia z wykorzystaniem wbudowanej funkcji

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

Uwaga! Pola użyte w kwerendzie będą widziane w oknie *Konstruktor wyrażeń* dopiero po zapisaniu kwerendy.

Ćwiczenie 2.19. Zmodyfikuj utworzoną wcześniej kwerendę *Ubezpieczeni* tak, aby wyświetlić rok urodzenia każdego ubezpieczonego studenta. Wykorzystaj do tego *Konstruktor wyrażeń* oraz funkcję wbudowaną *Year* — zwracającą rok z pola zawierającego datę. Kwerendę zapisz pod nazwą *Rok_urodzenia_studentow*.

Otwórz w widoku *projektu* kwerendę *Ubezpieczeni*. Do siatki projektowej dodaj pole *Data_ur*. Zapisz kwerendę pod nazwą *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 2.41. 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 wyrażeń*, z bieżącej kwerendy, poleceniem *Wklej* lub dwukrotnie klikając nazwę pola.

W wyrażeniu pojawi się nazwa pola w kwadratowym nawiasie, a całe wyrażenie 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()</code> , <code>Now()</code>	Podanie bieżącej daty lub daty i czasu
<code>DateAdd(...)</code> , <code>DateDiff(...)</code>	Obliczanie różnic i sum dat w dniach, latach itp.
<code>Day(...)</code> , <code>Year(...)</code> , <code>Month(...)</code> , <code>Minute(...)</code>	Wyodrębnienie z daty jej części
<code>IsDate(...)</code> , <code>IsNull(...)</code>	Sprawdzenie, czy dany obiekt jest, odpowiednio, datą, wartością <code>Null</code> itp.
<code>Left(...)</code> , <code>Right(...)</code> , <code>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 `Date()`, zwracająca aktualną datę. Zaprojektuj kwerendę, która obliczy wiek studenta według działania: $Wiek = rok(dzisiejsza\ data) - rok(data\ urodzenia)$. Odpowiednie wyrażenie utwórz wykorzystując *Konstruktor wyrażień*. Oprócz wieku wyświetl `ID_studenta` oraz kierunek studiów. Utworzoną kwerendę zapisz pod nazwą *Wiek_studentow*.

W tym przykładzie parametrem funkcji `Year` jest funkcja `Date()`. 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 pod nazwą *Srednia_ocen_informatykov*. Do utworzenia wyrażenia wykorzystaj *Konstruktor wyrażień*, który ułatwi i przyspieszy podanie nazw wszystkich pól zawierających oceny z poszczególnych przedmiotów. Oprócz średniej, obliczonej z dwoma miejscami dziesiętnymi, wyświetl rok studiów, kod kierunku oraz imię i nazwisko studenta zdefiniowane w kwerendzie *Dane_osobowe_studentow*.

Aby obliczyć średnią wartość danych zawartych w kilku polach, napisz wyrażenie zawierające sumę tych pól podzieloną przez liczbę przedmiotów sumowanych. W widoku *projektu* kwerendy dodaj tabelę *Studenci_uczelnia* oraz kwerendy *Dane_osobowe_studentow* i *Oceny_informatykwow*. Dodaj niezbędne pola do siatki projektowej (*ID_albumu*, *Imie_i_nazwisko*, *Rok_studiow*, *Kod_kierunku*, *Matematyka*, *Informatyka*, *Jezyk_angielski*, *Jezyk_niemiecki*, *Przedsiębiorczosc*, *Filozofia*, *WF*). Zapisz kwerendę (ważne, aby pola były widoczne). W nowym polu poleceniem *Konstruuuj...* uruchom *Konstruktor wyrażień*. Utwórz wyrażenie obliczające średnią ocen: $Srednia_ocen: ([Matematyka]+[Informatyka]+[Jezyk_angielski]+[Jezyk_niemiecki]+[Przedsiębiorczosc]+[Filozofia]+[WF])/7$, ustal sposób sortowania i wyłącz widok pól z ocenami. Kwerenda po uruchomieniu wyświetla obliczone średnie. Jeżeli zamiast liczb wyświetliły się krzyżyki #####, oznacza to, że kolumna jest za wąska i należy ją poszerzyć. A co z tak dużą liczbą cyfr po przecinku? Sposób wyświetlania wyniku należy ustalić we właściwościach pola, dostępnych w menu kontekstowym pola, w siatce projektowej, w widoku *projektu* kwerendy. W widoku *projektu*, w grupie *Pokazywanie/ukrywanie*, również jest dostępna opcja wyświetlenia *Arkusz właściwości*. Dopilnuj jednak, żeby aktywnym polem było to z wyrażeniem obliczającym średnią. W arkuszu właściwości ustal *Format* liczby na *Standardowy* i 2 miejsca dziesiętne. Zapisz kwerendę i uruchom ją.

Po zamknięciu i ponownym otwarciu tej kwerendy w widoku *projektu* nie ma nazw przedmiotów, jest tylko wyrażenie obliczające średnią. Wyłączona została widoczność pól, a ponieważ tabele są połączone ze sobą relacjami, nie ma potrzeby zamieszczania nazw przedmiotów w projekcie tej kwerendy. Wstawiając pola do wyrażenia, można było pobierać je bezpośrednio z kwerendy *Oceny_informatykwow*.

Ćwiczenie 2.21. Oblicz średnie ocen wszystkich studentów, biorąc pod uwagę sześć przedmiotów, z których wszyscy studenci mają oceny (matematyka, język angielski i język niemiecki, przedsiębiorczość, filozofia i wychowanie fizyczne). Wykonaj obliczenia na grupach rekordów i wyświetl tylko niezbędne dane, jak na rysunku 2.42. Utworzoną kwerendę zapisz pod nazwą *Srednia_ocen*.

ID_albumu	Imie_i_nazwisko	KOD_kierunku	Rok_studiow	Srednia_ocen
WU00001	ADRIANNA MAKOWSKA	FI	3	3,83
WU00003	OLIVIA RUMIŃSKA	CH	2	3,33
WU00004	DOMINIK SOCHATZKI	BI	1	4,00
WU00005	ALEKSANDRA KWIATKOWSKA	MA	2	4,33
WU00007	MICHAŁ MIKOŁAJEWICZ	IN	2	4,50

Rysunek 2.42. Kwerenda wybierająca dane z polem obliczeniowym

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 razie w polu pojawi się napis *Brak*. Utworzoną kwerendę zapisz pod nazwą *Stypendium_naukowe*.

W widoku *projektu* nowej kwerendy użyj — tym razem zamiast tabeli — utworzonej wcześniej kwerendy *Srednia_ocen*. Wybierz z niej trzy niezbędne pola: *ID_albumu*, *KOD_kierunku* oraz *Srednia_ocen*. Zapisz kwerendę. W nowym polu utwórz wyrażenie, wykorzystując okno *Konstruktor wyrażień*. Wśród funkcji wbudowanych, zgrupowanych w bloku funkcji przepływu sterowania programem, znajduje się funkcja wyboru *Iif*, która w oknie *Konstruktor wyrażień* pojawi się jako: *Iif(«expr»; «truepart»; «falsepart»)*. Należy ją rozumieć następująco: *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.

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 został zilustrowany na rysunku 2.43.

ID_albumu	KOD_kierunku	Srednia_ocen	Stypendium_naukowe	Stypendium_naukowe2
WU00011	MA	4,00	Stypendium	Tak
WU00062	CH	4,00	Stypendium	Tak
WU00013	MA	3,83	Brak	Nie
WU00108	CH	3,83	Brak	Nie
WU00109	MA	3,83	Brak	Nie

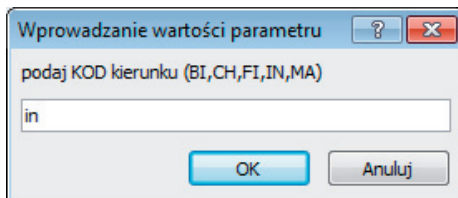
Rysunek 2.43. Wynik działania kwerendy z wyrażeniem zawierającym funkcję warunkową *Iif*

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 miejscowość zamieszkania wszystkich studentów z danego kierunku studiów. Przy każdym uruchomieniu kwerendy będzie można po podaniu kodu kierunku wyświetlić dane jego studentów. Utworzoną kwerendę zapisz pod nazwą *Studenci_wg_kierunku_z_parametrem*.

W widoku *projektu* dodaj table z danymi adresowymi i uczelnianymi studentów. Wybierz niezbędne pola. W kryterium wyboru pola *KOD_kierunku* podaj kryterium zawierające parametr. Wystarczy wpisać w nawiasach kwadratowych 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 2.44. Kryterium przyjmie wówczas postać: [podaj KOD kierunku (BI,CH,FI,IN,MA)]. Nie ma konieczności wpisywania parametrów wielkimi literami — Access radzi sobie zarówno z małymi, jak i z wielkimi literami.



Rysunek 2.44. Okno wprowadzania parametrów kwerendy

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 danym 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 2.45. Uruchom zapytanie i wybierz studentów informatyki mieszkających w Szczecinie lub 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 (BI,CH,FI,IN,MA)]	[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 (BI,CH,FI,IN,MA)]	[podaj 1. miasto zamieszkania]
lub:			[podaj 2. miasto zamieszkania]

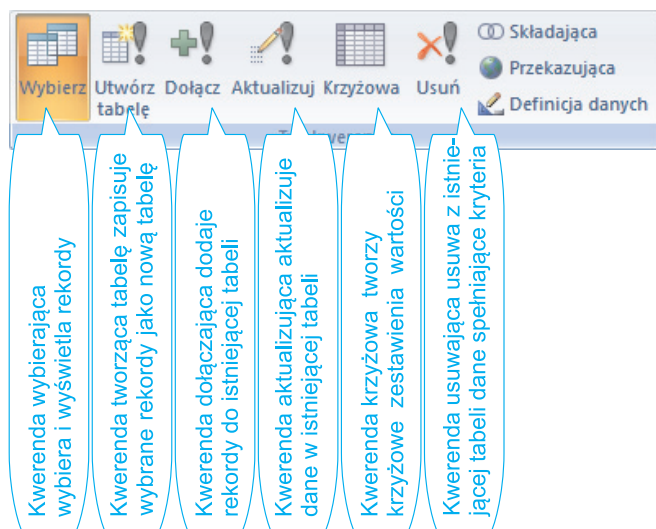
Rysunek 2.45. Kwerenda z wieloma parametrami

Zwróć uwagę, że na rysunku 2.45 użyta została alternatywa dwóch warunków parametrycznych. Zastanów się, dlaczego dolne rozwiązanie jest niepoprawne dla tego przykładu. Jakie rekordy otrzymasz w wyniku działania tej kwerendy, jeśli jako parametry podasz kolejno: *IN*, *Szczecin*, *Toruń*?

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 pod nazwą *Srednia_ocen_z_parametrem*. Wskazówką niech w tym miejscu będzie zapis kryteriów. Podawane przy uruchomieniu oceny muszą zastąpić parametry, a zatem: $\gt;=[\text{Podaj minimalną średnią}]$ And $\le;=[\text{Podaj maksymalną średnią}]$.

2.4.2. Kwerendy funkcjonalne — modyfikujące

Oprócz kwerend wybierających można tworzyć kwerendy funkcjonalne. Lista kwerend, pokazana na rysunku 2.46, jest dostępna w widoku *projektu* kwerendy, w grupie narzędzi *Typ kwerendy*. Typy kwerend są dostępne również w menu kontekstowym obszaru tabel.



Rysunek 2.46. 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 zachować szczególną ostrożność. 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. Dlatego przed przystąpieniem do dalszych ćwiczeń wykonaj kopię bazy *Uczelnia* i zapisz ją pod nazwą *Uczelnia_kopia*.

Kwerenda aktualizująca


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

Aby wykonać zadanie, skorzystaj z tabeli *Studenci_dane*. Wykonaj jej kopię i nazwij ją *Studenci_dane_aktualizacja*. Ponieważ każde uruchomienie kwerendy aktualizującej oznacza przeliczenie wartości, trzeba być bardzo ostrożnym przy jej uruchamianiu. To właśnie dlatego kwerendy tego typu mają ikonę opatrzoną wykrzyknikiem symbolizującym 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.

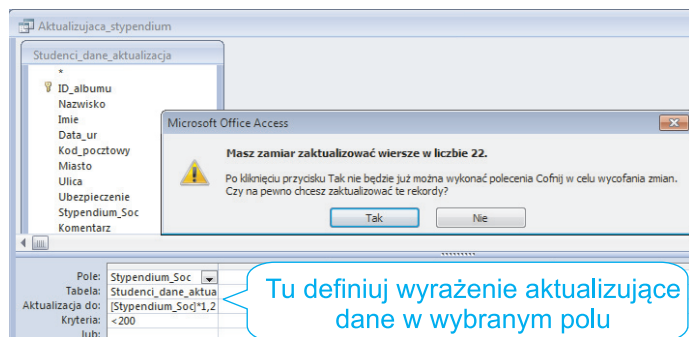
Utwórz nową kwerendę w widoku *projektu*. Wybierz tabelę *Studenci_dane_aktualizacja* i przeciągnij z listy pól tabeli do siatki projektu pole *Stypendium_Soc*, które będziesz

aktualizować lub dla którego chcesz ustawić kryteria. Wpisz <200 w komórce *Kryteria*. Obecnie jest to kwerenda wybierająca, która wyświetli z tabeli *Studenci_dane_aktualizacja* wszystkie stypendia niższe niż 200 złotych.

Aby możliwe było przeliczenie tych kwot, zmień typ kwerendy na aktualizującą. W grupie narzędzi *Typ kwerendy* wybierz przycisk *Aktualizuj*. Typ kwerendy możesz również wybrać z menu kontekstowego obszaru tabel w projekcie kwerendy.

Po zmianie typu kwerendy w siatce projektowej została dodana komórka *Aktualizacja do*. Można do niej wpisać wyrażenie lub wartość, która ma być wykorzystana do zmiany zawartości pola. Wpiszemy tu wyrażenie, które zwiększy wartość obecnego stypendium o 20%. Wpisz `[Stypendium_Soc]*1,2`. Zapisz kwerendę pod nazwą *Aktualizujaca_stypendium*. Zauważ, że kwerenda wyróżnia się inną ikoną:  *Aktualizujaca_stypendium*. Wykrzyknik w ikonie ma nie tylko odróżniać tę kwerendę od kwerend wybierających, ale i ostrzegać przed przypadkowym lub nieprzemyślanym jej użyciem.

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, co ilustruje rysunek 2.47. Po wykonaniu kwerendy sprawdź wartości stypendiów w zaktualizowanej tabeli.



Rysunek 2.47. Kwerenda aktualizująca dane

Tu definiuj wyrażenie aktualizujące dane w wybranym polu

Użycie kwerendy aktualizującej jest niezmiernie wygodne i pomocne w przeliczaniu istniejących lub podawaniu nowych wartości. Jeśli jednak nie zamierzamy w przyszłości użyć takiej kwerendy, po jej zastosowaniu można — a nawet należy — po prostu ją usunąć.

Kwerenda tworząca tabelę

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

Rozwiązanie zadania tego 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. Wykorzystaj kwerendę *Oceny_informatykov* do utworzenia nowej tabeli, która będzie zawierać dane z ocenami studentów informatyki.

W zakładce zawierającej kwerendy odszukaj kwerendę *Oceny_informatykov* i wykonaj jej kopię, a następnie zmień nazwę kopii na *Oceny_informatykov_tworzaca_tabele*. Otwórz ją w widoku *projektu*. Zmień jej typ na kwerendę tworzącą nową tabelę, wybierając ten typ z listy typów kwerend, widocznej na rysunku 2.46. Podaj nazwę nowej tabeli, która powstanie po uruchomieniu kwerendy: *Oceny_informatykov_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.accdb* 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 tabela ta została utworzona zgodnie z projektem tabeli *Wykladowcy*. Aby dołączyć tabelę do istniejącej tabeli, użyj *kwerendy dołączającej*. Zanim przystąpisz do tworzenia kwerendy dołączającej, utwórz kopię tabeli *Wykladowcy* (*Wykladowcy_kopia*). Będzie potrzebna do dalszej analizy.

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, symbolizującą 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*. Utworzoną kwerendę zapisz pod nazwą *Dolaczajaca_tabele_wykladowcy-filia*. Teraz uruchom kwerendę dołączającą. W odpowiedzi na komunikat o próbie dodania rekordów do tabeli potwierdź chęć ich dodania.

Na przygotowanej wcześniej tabeli *Wykladowcy_kopia* sprawdź, jak będzie się zachowywała kwerenda przy kilku uruchomieniach. Jednak wcześniej w widoku projektu kwerendy zmień tabelę *Wykladowcy* na *Wykladowcy_kopia* przez ponowne wybranie typu kwerendy dołączającej przyciskiem *Dołącz*. Uchroni to dane potrzebne do dalszych ćwiczeń. Czy istniejące już w tabeli dane będą nadpisywane, czy podjęta zostanie próba ponownego dodania tych samych wykładowców? Czy podczas wykonywania tego typu funkcjonalnej kwerendy następuje jakaś analiza danych, czy tylko mechaniczne dopi-

sanie 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.accdb* znajduje się tabela *Konsultacje-filia*, w której zamieszczone zostały terminy konsultacji dołączonych wykładowców. Utwórz kwerendę dołączającą te dane do tabeli *Konsultacje*. Kwerendę zapisz pod nazwą *Dolaczajaca_konsultacje_filia*. 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 (*Wykladowcy*), dopiero później można wpisać dane do tabeli podrzędnej (*Konsultacje*).

Kwerenda usuwająca

Jeśli rekordy zostaną usunięte za pomocą kwerendy usuwającej, która usuwa z jednej tabeli lub z kilku tabel wiersze spełniające określone kryteria, czynności tej nie można cofnąć. Dlatego też przed uruchomieniem kwerendy należy obejrzyć dane, które zostały przeznaczone do usunięcia. Aby to uczynić, trzeba kliknąć przycisk *Widok* na pasku narzędzi i obejrzyć kwerendę w widoku *arkusza danych*.

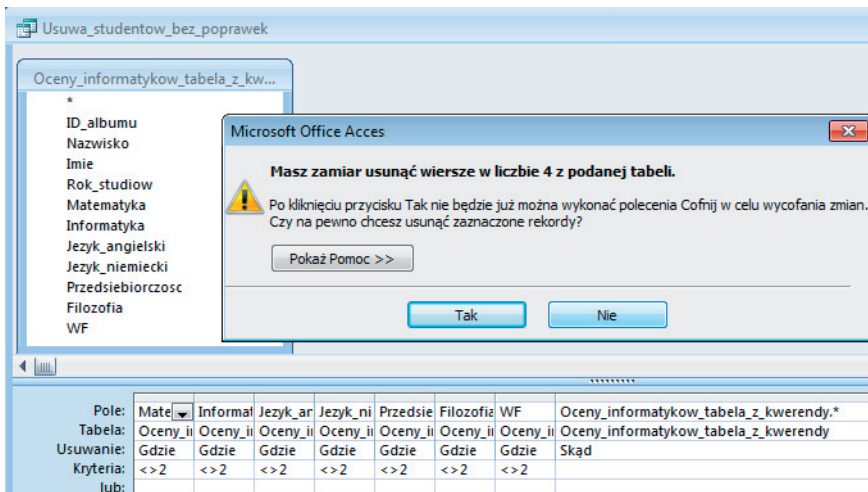
Zawsze należy sporządzać zapasowe kopie danych. Jeśli zostaną usunięte niewłaściwe rekordy, można je odzyskać z kopii zapasowej.

Ćwiczenie 2.27. Z tabeli utworzonej w ćwiczeniu 2.26 usuń dane tych studentów, którzy nie mają żadnej poprawki, czyli otrzymali ze wszystkich przedmiotów ocenę różną od niedostatecznej (2).

Praca ponownie będzie dwuetapowa: najpierw utworzymy kwerendę wybierającą, a następnie przekształcimy ją w funkcjonalną. Utwórz zatem nową kwerendę w widoku *projektu*. Wybierz tabelę *Oceny_informatykov_tabela_z_kwerendy*. Dodaj nazwy wszystkich przedmiotów do wiersza *Pole* w siatce projektu i określ kryteria wyboru (usuwania), wpisując *<>2* dla każdego przedmiotu. Uruchom kwerendę wybierającą wszystkie te rekordy, które za chwilę zostaną usunięte (trwale) z tabeli. Powinny pokazać się tylko oceny pozytywne czterech studentów, którzy za chwilę zostaną usunięci.

Powróć do widoku *projektu* i w grupie *Typ kwerendy* wybierz przycisk *Usuń*. W siatce projektu kwerendy zostanie dodany wiersz *Usuwanie* i automatycznie pojawi się słowo *Gdzie*. Przeciągnij gwiazdkę (*) z listy pól tabeli do siatki projektu kwerendy. W komórce *Usuwanie* pojawi się słowo *Skąd*. Projekt kwerendy pokazano na rysunku 2.48. Kwerendę zapisz pod nazwą *Usuwa_studentow_bez_poprawek*.

Kliknij przycisk *Uruchom*, aby usunąć wybrane rekordy. Potwierdź jeszcze w oknie komunikatu chęć usunięcia czterech rekordów z tabeli, jak na rysunku 2.48. Po usunięciu danych sprawdź zawartość tabeli *Oceny_informatykov_tabela_z_kwerendy*. Wszyscy studenci mają oceny niedostateczne. Zmień nazwę tabeli na *Poprawki_informatykov*.



Rysunek 2.48. Kwerenda usuwająca dane z tabeli

W niektórych przypadkach uruchomienie kwerendy usuwającej może spowodować usunięcie rekordów z tabeli 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. Wówczas, usuwając rekordy z 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 2.49), tworząc krzyżowe zestawienie podsumowanych lub zliczonych danych.

Akademik	Suma ID_albumu	BI	CH	FI	IN	MA
DS1		23	5	4	4	3
DS2		25	3	4	6	5
DS3		24	8	2	6	2

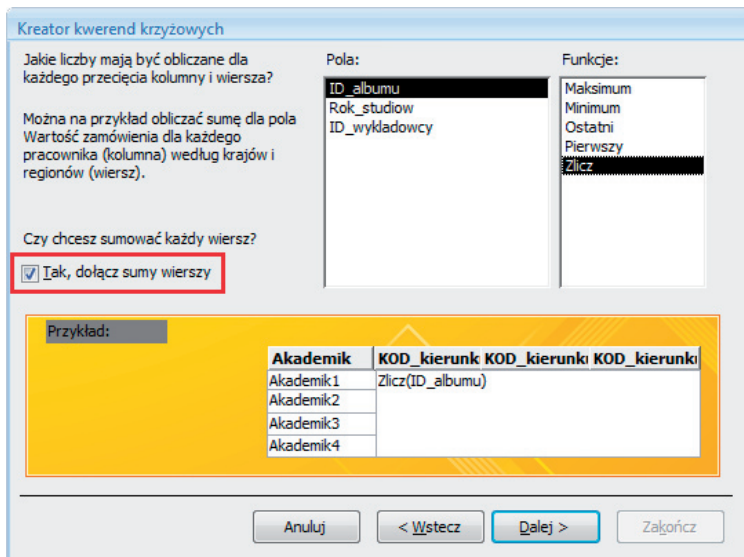
Rysunek 2.49. Kwerenda krzyżowa

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

Aby utworzyć kwerendę krzyżową, posłużymy się narzędziem *Kreator kwerend krzyżowych*, które wybierzemy z zestawu kreatorów po naciśnięciu przycisku *Kreator kwerend* z grupy *Inne*.

- 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, wybierz więc pole *Akademik*.

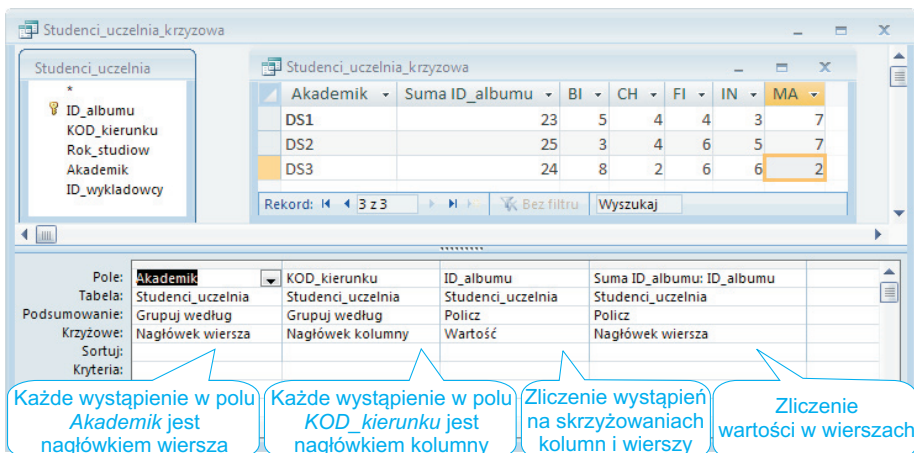
- W trzecim kroku kreatora wskaż, które wartości mają być nagłówkami kolumn. W naszym przykładzie będą to symbole kierunków studiów, wybierz więc pole *KOD_kierunku*.
- W kolejnym kroku ustal, jakie liczby mają być obliczane dla każdego przecięcia kolumny i wiersza. Dokonaj wyboru pola oraz funkcji. Tu możesz zaznaczyć opcję zliczania wszystkich wartości w każdym wierszu. Dla naszego przykładu ustal opcje zgodnie z rysunkiem 2.50.



Rysunek 2.50. Kreator kwerend krzyżowych: wybór pola i funkcji

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

Jak wygląda utworzona kwerenda w widoku *projektu*? Oba widoki, zarówno arkusza danych, jak i projektu, pokazano na rysunku 2.51, gdzie zostały zamieszczone wyjaśnienia.



Rysunek 2.51. 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 jedne dane stanowią nagłówki wierszy, a inne nagłówki kolumn, nie są możliwe w macierzystej tabeli.

Wykonaj ten projekt ponownie, zamieniając miejscami kolumny z wierszami. Powstałą kwerendę zapisz pod nazwą *Studenci_uczelnia_krzyzowa_odwrotnie*.

Ćwiczenie 2.28. Korzystając z kreatora kwerend krzyżowych, zaprojektuj kwerendę, która na podstawie kwerendy *Srednia_ocen* utworzy zestawienie średnich ocen na wszystkich kierunkach i rocznikach studiów. Zwróć też w kolumnie *Suma Srednia_ocen* najwyższą średnią w każdym roczniku, tak jak to pokazano na rysunku 2.52. Kwerendę zapisz pod nazwą *Srednia_ocen_krzyzowa*.

Rok_studiow	Suma Srednia_ocen	BI	CH	FI	IN	MA
1	4,83	4,17	3,83	4,83	4,83	4,33
2	4,50	4,00	4,33	4,50	4,50	4,33
3	4,33	4,33	4,00	4,17	4,17	4,17

Rysunek 2.52. Wynik działania kwerendy z ćwiczenia 2.28

Na rysunku przedstawiono wartości liczbowe (standardowe) z dwoma miejscami dziesiętnymi. Właściwości należy ustawiać w widoku *projektu* zarówno dla zliczania wystąpień na skrzyżowaniach wierszy i kolumn, jak i dla podsumowań wierszy. Zwróć uwagę, że pomimo wyboru funkcji *Maksimum* w tabeli wyświetlana jest kolumna *Suma Srednia_ocen*.

Ćwiczenie 2.29. Korzystając z kreatora kwerend krzyżowych, zaprojektuj kwerendę, która na podstawie tabeli *Studenci_uczelnia* utworzy zestawienie liczby osób studiujących na danym kierunku, z rozbiciem na poszczególne roczniki. Pokaże też sumę studentów na każdym kierunku, tak jak to zaprezentowano na rysunku 2.53. Kwerendę zapisz pod nazwą *Studenci_uczelnia_kierunek_rok_krzyzowa*.

KOD_kierun	Suma ID_albumu	1	2	3
BI	16	7	4	5
CH	10	3	5	2
FI	16	5	6	5
IN	14	3	5	6
MA	16	5	6	5

Rysunek 2.53. 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 2.54. Powstałą kwerendę zapisz pod nazwą *Stypendium_i_kierunek_krzyzowa*.

Stypendium_Soc	Suma	ID_alt	BI	CH	FI	IN	MA
130,00 zł	4	1	1	1	1	1	1
150,00 zł	10	2	2	2	3	3	
180,00 zł	8	2	2	3	1		
200,00 zł	1				1		
230,00 zł	1			1			

Rysunek 2.54. Wynik działania kwerendy z ćwiczenia 2.30

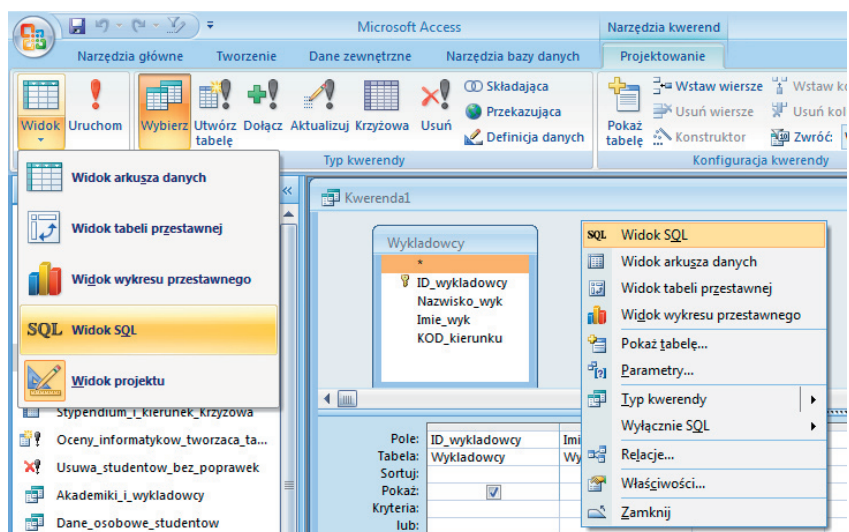
2.4.3. Język zapytań SQL

Zapytania w bazach danych zazwyczaj tworzy się za pomocą języka SQL (jego dialektów). Jednak w Accessie 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.

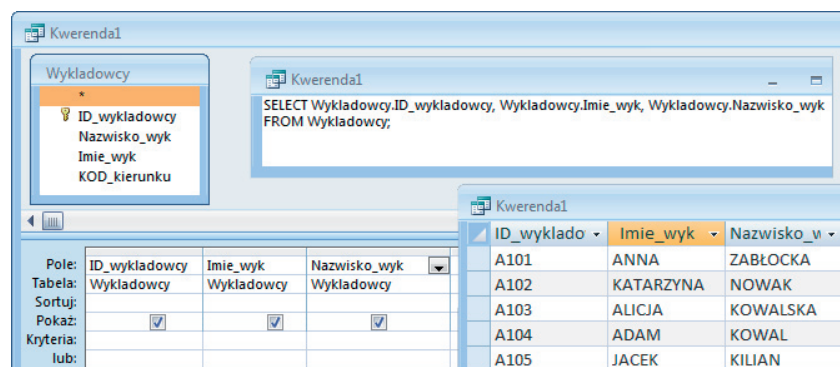
Kwerenda oparta na jednej tabeli

Ponieważ każda kwerenda utworzona przy użyciu graficznego szablonu ma swój zapis w **kodzie SQL**, zobaczmy, jak wygląda taki kod dla kwerendy wybierającej dane z tabeli *Wykladowcy*. W widoku *projektu* wybierz z niej pola z ID oraz imionami i nazwiskami 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 listy rozwijanej przycisku *Widok* zamieszczonego w grupie *Widoki*, co ilustruje rysunek 2.55.

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 kwerendy wybierającej zgromadzone zostały na rysunku 2.56.



Rysunek 2.55. Sposoby przejścia z widoku projektu kwerendy do widoku SQL



Rysunek 2.56. 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 Wykładowcy.ID_wykladowcy, Wykładowcy.Imie_wyk, Wykładowcy.Nazwisko_wyk
FROM Wykładowcy;
```

Język SQL jest językiem strukturalnym. Przeanalizujmy 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* ozna-

cza 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.Imie_wyk, Wykladowcy.Nazwisko_wyk, Wykladowcy.KOD_kierunku
FROM Wykladowcy;
```

Równoważny powyższemu jest zapis:

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

Zamiast wymieniać wszystkie pola z tabeli *Wykladowcy*, użyto znaku ***.

Kwerenda wybierająca dane z kilku tabel

Zobaczmy, jak wygląda kod SQL dla kwerendy wybierającej dane z dwóch tabel, na przykład *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 już dla nas 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.

CROSS JOIN — złączenie krzyżowe; wynikiem jest 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 (*FROM Studenci_dane INNER JOIN Studenci_uczelnia*), których *ID_albumu* są sobie równe (te pola są bowiem połączone relacją). Jednak możliwe jest również następujące polecenie: wybierz wszystkie rekordy występujące w tabeli *Studenci_dane* i tylko te rekordy z tabeli *Studenci_uczelnia* (*FROM Studenci_dane LEFT INNER JOIN Studenci_uczelnia*), których *ID_albumu* są sobie równe. I analogiczne polecenie: wybierz wszystkie rekordy występujące w tabeli *Studenci_uczelnia* i tylko te rekordy z tabeli *Studenci_dane* (*FROM Studenci_dane RIGHT INNER JOIN Studenci_uczelnia*), których *ID_albumu* są sobie równe. Tego typu sprzężenia nazywamy *sprzężeniem lewostronnym* lub *pravostronnym*. W widoku projektu kwerendy zmianę typu sprzężenia można wykonać w oknie *Właściwości sprzężenia*, które jest dostępne po dwukrotnym kliknięciu na linii symbolizującej sprzężenie tabel.

Sortowanie

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

```
SELECT Wykladowcy.ID_wykladowcy, Wykladowcy.Imie_wyk, Wykladowcy.
Nazwisko_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 **"IN"**. Kod SQL ma teraz postać:

```
SELECT Wykladowcy.ID_wykladowcy, Wykladowcy.Imie_wyk, Wykladowcy.
Nazwisko_wyk, Wykladowcy.KOD_kierunku
FROM Wykladowcy
WHERE (((Wykladowcy.KOD_kierunku)="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 koniunkcją 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)="IN" Or (Wykladowcy.KOD_kierunku)="MA"));
```

- Koniunkcja kryteriów: wypisz ID studentów trzeciego roku informatyki.

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

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

```
WHERE (((Studenci_uczelnia.Rok_studiov)=3) AND ((Studenci_uczelnia.KOD_kierunku)="IN" Or (Studenci_uczelnia.KOD_kierunku)="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 pod warunkiem, że student stypendium otrzymuje. Oto zapis kwerendy:

```
SELECT Studenci_dane.ID_albumu, Studenci_dane.Stypendium_Soc, [Stypendium_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_Soc* 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]));
```

Jeśli jednak chcesz użyć kilku parametrów, dla każdego z nich, nawet gdy dotyczy tego samego pola, musisz podać nazwę tabeli, nazwę pola i — opcjonalnie — treść komunikatu. A zatem rozbudowując 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).

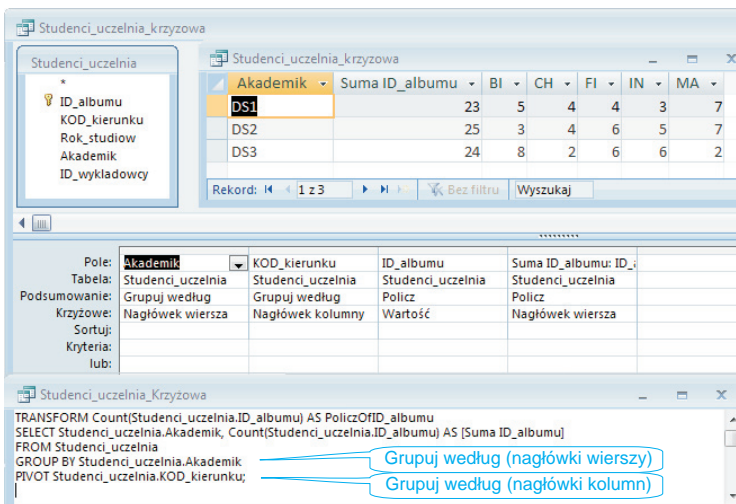
Kwerenda tworząca nową tabelę

Kod SQL kwerendy tworzącej tabelę różni się od kodu kwerendy wybierającej obecnością klauzuli **INTO**, która informuje, jak ma się nazywać nowa tabela, powstała po uruchomieniu kwerendy. Zapis kodu kwerendy tworzącej z tabeli *Studenci_uczelnia* nową tabelę, która zawiera 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 posłuży utworzona wcześniej kwerenda krzyżowa *Studenci_uczelnia_krzyzowa*. Została ona utworzona w oknie kreatora kwerend krzyżowych i samo przełączenie do widoku *projektu* może wywołać lekkie zaskoczenie. Na rysunku 2.57 zestawiono wszystkie trzy widoki tej skomplikowanej kwerendy: widok *projektu*, *arkusza danych* i *SQL*.



Rysunek 2.57. Kwerenda krzyżowa w trzech widokach: projektu, arkusza danych i SQL

Przyjrzyjmy się uważnie kolejnym liniom kodu tej kwerendy krzyżowej.

```
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) AS  
[Suma ID_albumu]
```

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*).

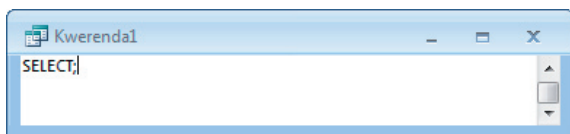
```
PIVOT Studenci_uczelnia.KOD_kierunku
```

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 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 2.58, w którym zdefiniujemy kwerendę za pomocą instrukcji języka SQL. W oknie wpisane jest tylko słowo **SELECT** i średnik, który kończy instrukcję SQL.



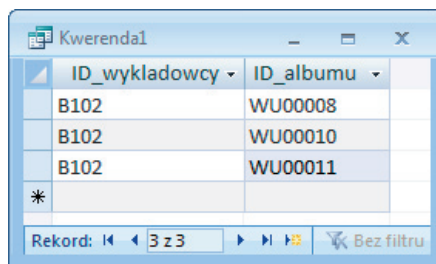
Rysunek 2.58. Okno tworzenia kwerendy w widoku kodu SQL

Po napisaniu kwerendy można 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 2.59, na którym wyświetlona jest lista studentów wykładowcy *B102*.

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

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



ID_wykladowcy	ID_albumu
B102	WU00008
B102	WU00010
B102	WU00011
*	

Rysunek 2.59. Lista studentów wykładowcy *B102*

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]));
```

Kwerendę zapisz pod nazwą *Wykaz_studentow_na_zajecia_wykladowcy_z_parametrem*.

Ćwiczenie 2.32. Utwórz zapytanie wybierające studentów (z podaniem ich imion i nazwisk), którzy uzyskali z matematyki ocenę wyższą od 3. Wynik działania kwerendy *Studenci_z_ocena_z_matematyki>3* zilustrowano na rysunku 2.60.

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));
```

Wydadź polecenie posortowania danych według nazwisk w porządku malejącym. W miejscu *porzadek* wpisz odpowiednie polecenie:

```
ORDER BY tabela.pole porzadek;
```

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));  
ORDER BY Studenci_dane.Nazwisko DESC;
```

Zapisz utworzoną kwerendę.

Warunki zapisywane w podwójnych nawiasach (()) nie są warunkami języka SQL, lecz warunkami języka wyrażeń Office Access (ściślej — języka Visual Basic for Applications).

Imie	Nazwisko	Matematyki
IRENA	ZIELIŃSKA	4
TERESA	ZALEWSKA	4
MONIKA	ZAJĄC	5
KAZIMIERZ	WYSZYŃSKI	4
CECYLIA	WIŚNIEWSKA	4
JANINA	WIECZOREK	5
STANISŁAW	TARKOWSKI	4
JULIANNA	SZCZERBIAK	4

Rysunek 2.60. Wykaz 46 rekordów spełniających kryterium z ćwiczenia 2.32

A

Access, 38, 43

adres

bezklasowy, 168, 170

broadcastowy, 156

domyślny, 168

fizyczny MAC, 155

IP, 156, 163, 164, 180, 189

IPv4, 164, 168

IPv6, 166

multicast, 157

niepubliczny, 168

nierutowalny, *Patrz:* adres niepubliczny

prywatny, *Patrz:* adres niepubliczny

publiczny, 167

rozgłoszeniowy, 167, 170

sieci, 167, 169

specjalny, 167

URL, 193

adware, 189, 301

aliasing, 229

anomalia, 44, 45

antyaliasing, 229

architektura klient-serwer, 182

arkusz danych, 65

arkusz kalkulacyjny, 9, 71

adres, 10

format komórki, 10

formuła, 10

tabela, 10

atak

DDoS, 190

DoS, 190

poprzez słownik, 184

atrybut, 11

autofiltr niestandardowy, 18

autofiltrowanie, 14

autoformularz, 103

B

barwa, 230, 234, 236

achromatyczna, 236

prosta, 232

baza danych, 11, 35, 37, 43

dekodowanie, 114

hierarchiczna, 37

jednotabelowa,

Patrz: baza danych kartotekowa

kartotekowa, 9, 36

kodowanie, 114

multimedialna, 37

obiektowa, 37

obiektowo-relacyjna, 37

plik, 43

relacyjna, 35, 37

sieciowa, 37

typy, 36

złożona, 37

Bell Alexander Graham, 271

Bézier Pierre Étienne, 254

bezklasowy routing międzydomenowy,

Patrz: CIDR

bezpieczeństwo danych,

Patrz: dane ochrona i bezpieczeństwo

BIOS, 122, 123

bitrate, 275

C

CIDR, 167, 169

cloud computing,

Patrz: przetwarzanie w chmurze

CMS, *Patrz:* system zarządzania treścią

cookies, 178

COPS, 184
CPU, *Patrz:* procesor
CSG, *Patrz:* konstruktywna geometria
brył
CSS, *Patrz:* kaskadowy arkusz stylów
CSV, 14
częstość bitowa, *Patrz:* bitrate

D

dane
autonumerowanie, 49
data/godzina, 49
hiperłącze, 50
liczba, 48, 51
nota, 48
obiekt OLE, 49, 64
ochrona i bezpieczeństwo, 112, 113,
183, 186
tak/nie, 49
tekst, 48, 51
typy, 47, 57
waluta, 49
załącznik, 50
Database Management System,
Patrz: system zarządzania bazą danych
datagram, 161, 162, 163
DB2, 38
decybel, 271
dekodowanie,
Patrz: baza danych dekodowanie
diagram obiektowo-związkowy, 41
dokument HTML, 193
domena publiczna, 301
DOS, *Patrz:* system operacyjny DOS
Drupal, 208
drzewo, 244
binarne, 244
korzeń, 244
dynamika dźwięku, 273
dysk twardy
HDD, 294
SSD, 294
dziedziczenie, 205

dźwięk, 269, 271
analogowy, 271
cyfrowy, 271

E

eksport, 70, 73
e-learning, 298
enkoder, 275
Ethernet, 152
exploit, 188

F

filtr zaawansowany, 20
filtrowanie, 15, 65, 66
według formularza, 68
z wykluczeniem wyboru, 16
firewall, *Patrz:* zaporą sieciową
folder, 127
formant, 105, 108, 190
format
AAC, 280
AIFF, 274
AVI, 282
bajtowo-dziesiętny, 164
BMP, 247
EPS, 257
FLAC, 280
GIF, 241, 247, 250
JFIF, 248
JPG (JPEG), 247, 251
MOV, 282
MP3 (MPEG), 274, 275, 276
MPEG, 282
OGG, 279
PNG, 241, 248, 252,
RealMedia, 282
RGB24, 282
SVG, 199, 257
TIFF, 248
WAVE, 274
formularz, 12, 14, 43, 103
dzielony, 104
freeware, 300

G

GIMP, 228
głębina bitowa, 237
GNU GPL, 301
GPU, 265
graficzny interfejs użytkownika,
 Patrz: GUI
grafika, 225
 2D, 225
 3D, 225, 258
 bitmapowa, *Patrz:* grafika rastrowa
 komputerowa, 225
 płaska, *Patrz:* grafika 2D
 przestrzenna, *Patrz:* grafika 3D
 rastrowa, 226, 257
 wektorowa, 253, 256, 257
GUI, 127

H

haker, 183
hasło, 113, 183, 184
hiperlink, 193
hipertekst, 192
host, 164, 169
HSPA, 152

I

identyfikacja danych, 41
IEEE, 152, 157
import, 70, 71
indeksowanie, 54
infradźwięki, 270
inżynieria społeczna, 188
IPtrace, 184
ISO, 157, 158

J

jaskrawość, 234, 237
jasność, 237
jednobarwność,
 Patrz: monochromatyczność

jednostka centralna, 117

język

 HTML, 193, 194
 PHP, 210
 skryptowy, 210
 XHTML, 194, 195
 XML, 194

język zapytań,

 QBE, 74
 SQL, 74, 95, 101

K

kabel

 ekranowany STP, 145, 152
 nieekranowany UTP, 145

kamera, 260, 263

karta rozszerzeń, 121

kaskadowość, 205

kaskadowy arkusz stylów, 202

katalog, 127

klaster, 128

klient, 193

klucz

 główny, 41, 49, 55, 56
 obcy, 42, 56
 podstawowy, *Patrz:* klucz główny
 własny, *Patrz:* klucz główny

kod

 Huffmana, 244
 prefiksowy, 243
 szesnastkowy, 249

kodek, 274, 279-283

kodowanie, *Patrz:* baza danych kodowanie

kolor, 230, 249

 indeksowany, 251

koło chromatyczne, 234

komórka, 161

kompatybilność, 117

kompresja

 bezstratna, 240, 274
 stratna, 240, 246, 251, 274, 278

komunikat, 161

 o błędzie, 53
 sieciowy, 167

koncentrator, 145, 154
konstruktywna geometria brył, 265
kontener multimedialny, 282
konwersja, 272
koń trojański, 184, 188, 189
kreator odnośników, 50
krotka, 44
kwantyzacja, 272
kwerenda, 43, 68, 74
 aktualizująca, 88
 dołączająca, 88, 90
 funkcjonalna, 88, 90
 krzyżowa, 88, 92, 100
 oparta na jednej tabeli, 95
 parametryczna, 86, 99
 tworząca tabelę, 88, 89, 100
 usuwająca, 88, 91
 wybierająca, 75, 88, 97

L

LanWatch, 184
licencja, 300, 302
link, 201
Linux, *Patrz:* system operacyjny Linux
logo, 256
LTE, 153
LZW, 241

M

magistrala, 121, 293
 AGP, 121, 265
 PCI, 121, 265
 PCI Express, 121, 265
 QuickPath, 294
 USB 3.0, 296
 wejścia-wyjścia, 121
makro, 43, 110
makropolecenie, *Patrz:* makro
malware, 187
maska sieci, 168
maska wprowadzania, 51, 52
maszyna wirtualna, 131

memrystor, 295
Microsoft SQL Server, 38
model
 barw, 231
 CIE La*b*, 231
 CIE XYZ, 231
 CMYK, 233
 HLS, 234
 HSB, 234
 HSV, 234
 OSI, 158, 162, 177, 182
 RGB, 232, 234
modeler, 259
moduł, 43
monochromatyczność, 237
most, 155
mostek
 południowy, 121, 122
 północny, 121, 122, 293
MySQL, 38, 208

N

nadmiarowość danych, 38
nasycenie, 234, 235, 236
NAT, 167
notacja dziesiętna z kropkami, 164
null, 48
numer sekwencyjny, 161

O

obrazek, 250
obrót, 264
ochrona danych,
 Patrz: dane ochrona i bezpieczeństwo
odcień, 234
odnośnik hipertekstowy, 201
odsyłacz, 201
okablowanie strukturalne, 152
oprogramowanie złośliwe, *Patrz:* malware
Oracle Universal Server, 38
oryginał analogowy, 240
osadzanie obiektu, 64

P

pakiet, 161
pamięć
 DRAM, 296
 flash, 295
 operacyjna, 118
 RAM, 118, 293
phishing, 191
piksel, 226, 239
plik, 43, 126
płyta główna, 120, 121,
podformularz, 105
podsieć, 174
pole, 11, 57, 64, 68, 99
 kluczowe, 46, 54, 55, 56
 rozmiar, 51
połączenie punkt-punkt, 293
port, 177
PostgreSQL, 38
potomek, 244
prawo autorskie, 300, 302
procesor, 118, 119, 121, 293
procesor graficzny, *Patrz:* GPU
program antywirusowy, 187
protokół, 160
 bezstanowy, 178
 DHCP, 180
 DNS, 180
 FTP, 177, 179
 HTTP, 177, 178, 185, 193
 HTTPS, 178, 185, 188
 IMAP, 179
 IP, 162, 164, 181
 IPv6, 166, 175, 176
 klient-serwer, 179
 nierutowalny, 177
 PGP, 186
 POP3, 177, 179, 188
 rutowalny, 177
 scentralizowany, 186
 SFTP, 179
 SMTP, 177, 179, 188
 SSH, 181

SSL, 178, 185
TCP, 162, 164
TCP/IP, 162, 164, 177, 183
telnet, 181, 184
próbkowanie, 272, 273
przełęczarka internetowa, 193
przełącznik, 145, 155, 189
przepływność, 152, *Patrz też:* bitrate
przepustowość, 152
przetwarzanie w chmurze, 297
przezroczystość, 247, 250, 253
punkt dostępowy Wi-Fi, 186

R

radiosity, 259
ramka, 158, 159, 161, 162
raport, 27, 43, 108
ray tracing, 259
redundancja, 44, 45
reguła sprawdzania poprawności, 53
rekord, 11, 44
relacja, 57
 jeden-do-jednego, 57, 58
 jeden-do-wielu, 57, 60
 wiele-do-wielu, 57, 61
renderowanie, 263
RLE, 240
robak, 188
rodzic, 244
rootkit, 189
router, 156, 163, 186
rozdzielczość, 226

S

segment, 161, 162
sektor, 128
serwer WWW, 182, 193, 206,
shareware, 301
sieć bezprzewodowa, 186
sieć komputerowa, 143, 144
 bezawaryjność, 182
 instalacja, 182
 konfiguracja, 182

- LAN, 144, 155, 156
- MAN, 144
- WAN, 144
- skalowanie, 264
- skaner, 188
- sniffing, 188
- Snoop, 184
- sortowanie danych, 23, 54, 65
- spoofing, 189
- spójnik
 - alternatywy, 18
 - koniunkcji, 18
- sprężenie, 57, 58
- spyware, 189
- SQLite, 38
- strona, 43
- strumień danych, 161
- sumy
 - częściowe, 25, 29
 - pośrednie, *Patrz:* sumy częściowe
- Sybase SQL Server, 38
- sygnet, 256
- system
 - plików, 127, 128
 - zarządzania bazą danych, 35, 37
 - zarządzania treścią, 207
- system operacyjny, 124, 126, 129
 - czasu rzeczywistego, 130
 - dla jednego użytkownika, 130
 - DOS, 125, 133, 137
 - jądro, 125
 - jednozadaniowy, 129
 - Linux, 136
 - powłoka, 125
 - rozproszony, 130
 - równoległy, 130
 - wielodostępny, 130
 - wielozadaniowy, 129
- szum, 277
- szybkie filtrowanie, 66
- szybkie sortowanie, 65
- szyfrowanie, 178, 181, 183, 185, 186, 188, 189,
 - Patrz też:* baza danych kodowanie
- szyna systemowa, *Patrz:* magistrala

Ś

- ścieżki przeszukiwań, 132
- światłowód, 152

T

- tabela, 43, 44, 45, 63, 70
 - kryteriów, 20, 21
 - połączona, 70
 - przestawna, 27
 - załączanie, 70, 72
- token, 148
- Token-Ring, *Patrz:* topologia pierścienia
- tony
 - niskie, 270
 - średnie, 270
 - wysokie, 270
- topologia
 - drzewa, 149
 - fizyczna, 144
 - gwiazdy, 145
 - komórkowa, 150
 - logiczna, 144
 - magistrali, 146
 - mieszana, 151
 - pierścienia, 147
 - podwójnego pierścienia, 149
- transformacja, 264
- translacja, 264
- transmisja
 - grupowa Multicast, 157
 - pojedyncza Unicast, 156
 - rozgłoszeniowa Broadcast, 157
- trasowanie, 163
- trojan, *Patrz:* koń trojański
- tryb
 - konsoli, 136
 - tekstowy, *Patrz:* tryb konsoli

U

- UEFI, 124
- układ współrzędnych, 260
- ultradźwięki, 270

urządzenia
peryferyjne, 118, 119
wejścia-wyjścia,
Patrz: urządzenia peryferyjne
zewnętrzne,
Patrz: urządzenia peryferyjne

W

warstwa
aplikacji, 158, 160, 161, 162, 163, 179, 182
dostępu do sieci, 162, 163
fizyczna, 158, 163
internetu, 162, 163
łącza danych, 158, 161
prezentacji, 158, 159
procesowa, *Patrz:* warstwa aplikacji
programów użytkowych,
Patrz: warstwa aplikacji
protokołu internetowego,
Patrz: warstwa internetu
sesji, 158, 159
warstwa sieciowa, 156, 158, 159, 161
transportowa, 158, 159, 161, 162, 163
zerowa, 158
wartość domyślna, 53
wersja beta, 301
węzeł, 193, 244
wideo, 281

wirus, 188
witryna WWW, 193, 206
WWW, 178, 192, 193
wykres, 26
przestawny, 29
wyrażenie, 82
wzmacniak, 153, 154

Z

zapora sieciowa, 187, 189
zapytanie, *Patrz:* kwerenda
zasoby, 128
zatrucie DNS, 191
zerowa długość dozwolona, 54
zgodność wsteczna, *Patrz:* kompatybilność
złącze
AT-BUS, 122
cylindryczne BNC, 146, 152
LPT, 122
PCI, 122
PS/2, 122
Serial ATA, 122
USB, 122
zombi, 190

Ź

źródło światła, 260, 263

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



- 1. ZAREJESTRUJ SIĘ**
- 2. PREZENTUJ KSIĄŻKI**
- 3. ZBIERAJ PROWIZJĘ**

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA

 **Helion SA**

Komputer łączy nasz mózg.

Steve Jobs, jedna z ikon rewolucji informatycznej, współzałożyciel i prezes firmy Apple Inc.

Na świecie po epoce kamienia, brązu i żelaza bez wątpienia przyszedł czas na epokę informatyczną. Komputery nadają tempo rozwojowi gospodarki, są niezbędne w obszarze nauki, ale służą także rozrywce i komunikacji.

Zestaw **Informatyka Europejczyka** jest całkowicie kompatybilny z wymaganiami, jakie stawia przed każdym uczniem współczesna informatyka. Został stworzony do nauczania informatyki w zakresie rozszerzonym w szkołach ponadgimnazjalnych, a jego treści, struktura, duża liczba przykładów i zadań pozwalają na doskonałe przygotowanie do egzaminu maturalnego.

Z drugą częścią podręcznika zgłębisz budowę komputera, zrozumiesz rolę i zadania systemu operacyjnego oraz usług sieciowych. Nauczysz się korzystać z zaawansowanych funkcji bazodanowych arkusza kalkulacyjnego, poznasz relacyjną bazę danych i język zapytań SQL. Następnie przyswoisz sobie zagadnienia multimedialnych, z uwzględnieniem grafiki bitmapowej, wektorowej, 3D, dźwięku i filmu. Odkryjesz sposób działania algorytmów kompresji danych, a także stworzysz i opublikujesz w internecie stronę WWW.



Na płycie CD zamieszczono materiały niezbędne do wykonania ćwiczeń z podręcznika, kody źródłowe programów opisanych w przykładach oraz oprogramowanie, które możesz legalnie zainstalować na swoim komputerze. Wybrane zadania z egzaminów dojrzałości umożliwią Ci nie tylko zapoznanie się z formą zadań pojawiających się na maturze, ale także pomogą w rozwijaniu Twojej pasji.



Kompletny zestaw Informatyka Europejczyka stanowią **podręcznik cz. 1 + podręcznik cz. 2**

Komplet podręczników oraz płyty z serii Informatyka Europejczyka pozwolą uczniom zdobywać wiedzę poprzez praktykę, a nauczycielom ułatwią przekazywanie nowego materiału w interesujący i niebanalny sposób. Helion, największe wydawnictwo informatyczne w Polsce, teraz pomaga zgłębić tajemnice świata komputerów także pokoleniu przyszłych specjalistów.

Wciśnij Enter i do dzieła!

<http://edukacja.helion.pl>

Nr katalogowy: 5621



Księgarnia internetowa:
<http://helion.pl>



Zamówienia telefoniczne:
0 801 339900



0 601 339900



Helion

Sprawdź najnowsze promocje:
<http://helion.pl/promocje>
Książki najchętniej czytane:
<http://helion.pl/bestsellery>
Zamów informacje o nowościach:
<http://helion.pl/nowosci>

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

helion.pl
księgarnia
internetowa

ISBN 978-83-246-2824-7



Informatyka w najlepszym wydaniu