



Instrukcje grupowania w języku SQL, etap II

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Prezentacja multimedialna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Źródło: Tobias Fischer, domena publiczna.

Systemy bazodanowe implementujące język SQL zawierają narzędzia nie tylko do prostego wybierania danych z określonych tabel. Pozwalają również dokonywać obliczeń na wielu zestawach rekordów jednocześnie, podobnie jak tabele przestawne w arkuszach kalkulacyjnych. To wszystko dzięki mechanizmowi grupowania i funkcjom agregującym.

Więcej informacji na temat instrukcji grupowania w języku SQL znajdziesz w e-materiałach:

- [Instrukcje grupowania w języku SQL, etap I,](#)
- [Instrukcje grupowania w języku SQL, etap III,](#)
- [Instrukcje grupowania w języku SQL, etap IV.](#)

Twoje cele

- Wyjaśnisz, na czym polega grupowanie rekordów.
- Użyjesz w praktyce grupowania rekordów i funkcji agregujących.
- Zestawisz i przeanalizujesz dane z wielu tabel.
- Wykorzystasz w kwerendach podzapytania.

Przeczytaj

Baza danych

Dysponujemy bazą danych SQLite3 zapisaną w pliku `uczniowie2.db`. Ćwiczenia wykonamy w programie SQLiteStudio, ale można do tego celu użyć wiersza poleceń bazy SQLite3. Informacje na temat tych narzędzi zawarte zostały w materiale [Definiowanie schematu bazy danych w języku SQL, etap II](#). Zapytania można również wykonać za pomocą skryptu języka Python, który omówiono w materiale [Instrukcje wyszukiwania w języku SQL, etap III](#).

Baza danych SQLite3 `uczniowie2.db`.
Plik o rozmiarze 832.00 KB w języku polskim

Po uruchomieniu programu SQLiteStudio wybierz polecenie Bazy danych | Dodaj bazę danych i wskaż położenie pliku `uczniowie2.db`. Następnie połącz się z bazą, dwukrotnie klikając nazwę bazy w panelu Bazy danych.

Program SQLiteStudio ułatwia poznanie schematu bazy danych, udostępniając wszystkie informacje potrzebne do poznania tabel i łączących je relacji. Po dwukrotnym kliknięciu wybranej tabeli w panelu po lewej stronie, w zakładce **Struktura** zobaczymy nazwy i typy danych pól zawartych w tabeli.

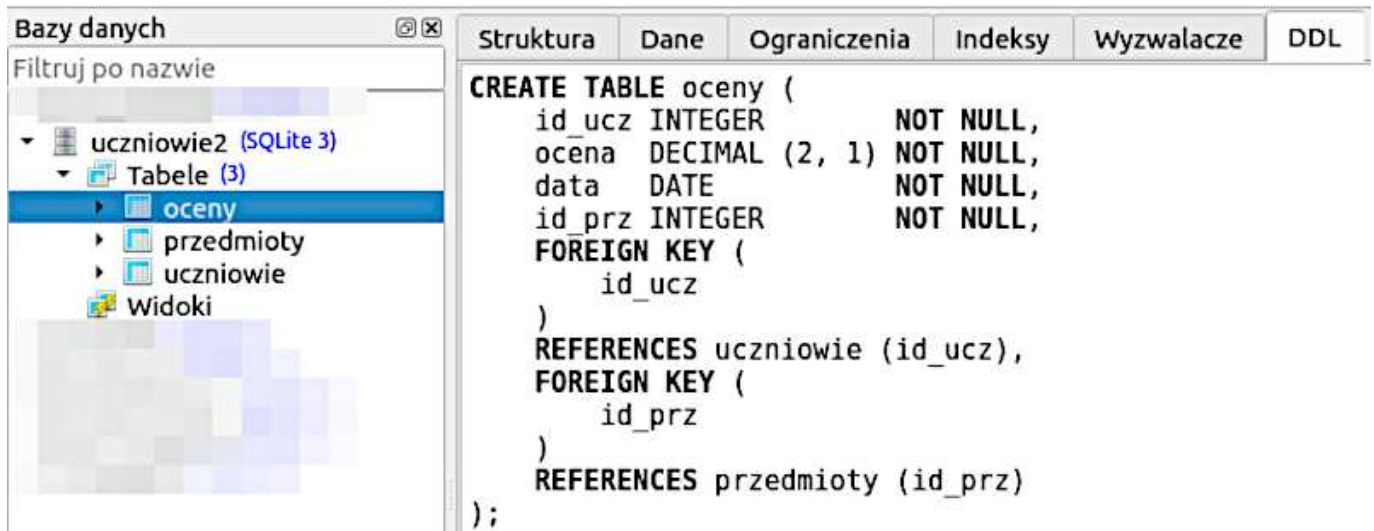
Nazwa	Typ danych	Klucz Główny	Klucz Obcy	Wartości unikalne	Warunek	Niepuste
1 id_ucz	INTEGER					
2 ocena	DECIMAL (2, 1)					
3 data	DATE					
4 id_prz	INTEGER					

Typ	Nazwa	Szczegóły
1 FOREIGN KEY		(id_ucz) REFERENCES uczniowie (id_ucz)
2 FOREIGN KEY		(id_prz) REFERENCES przedmioty (id_prz)

Podgląd struktury tabeli w programie SQLiteStudio.

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Jeżeli klikniemy zakładkę **DDL**, zobaczymy polecenie SQL, które zostało użyte do utworzenia danej tabeli i ewentualnych powiązań.



Podgląd polecenia SQL DDL, które posłużyło do utworzenia tabeli w programie SQLiteStudio.

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Ćwiczenie 1

Przeanalizuj schemat bazy, korzystając z informacji o tabelach widocznych na zakładkach Struktura i/lub DDL.

Grupowanie danych

Spróbujmy przeanalizować niektóre dane dotyczące uczniów zapisane w bazie danych.

Polecenie 1

Napiszmy kwerendę, która wyświetli liczbę uczniów mieszkających na tej samej ulicy.

Zacznijmy od zapytania zliczającego uczniów:

```
1 SELECT COUNT(id_ucz) AS liczba_uczniow
2 FROM uczniowie;
```

Dzięki funkcji COUNT() dowiemy się, że w bazie zapisano dane 414 uczniów.

Jeśli jednak chcemy zobaczyć, ilu uczniów mieszka na tej samej ulicy, powinniśmy pogrupować uczniów według nazw ulic i dopiero później zliczyć poszczególne grupy.

```
1 SELECT ulica, COUNT(id_ucz) liczba_uczniow
2 FROM uczniowie
3 GROUP BY ulica
4 ORDER BY liczba_uczniow ASC;
```

Na przykładzie tego zapytania widzimy, że do grupowania rekordów służy klauzula `GROUP BY`, po której podajemy nazwę pola. Ponadto w klauzuli `SELECT` uwzględniliśmy pole *ulica*, aby widzieć nazwy ulic, dla których zliczamy uczniów. Zapytanie pokazuje również, że klauzula `AS` jest opcjonalna, kiedy tworzymy aliasy pól. Klauzulę sortowania `ORDER BY` dodaliśmy dla czytelności wyników.

Początkowe zwrócone rekordy:

1	Kosmiczna	6
2	Worcella	6
3	Ksiezycowa	7
4	Baczynskiego	8
5	Kopernika	8

Ważne!

Grupowanie poprzedza operacje wykonywane przez funkcje agregujące (np. `COUNT()`), które mogą działać na wszystkich rekordach, na jednej grupie lub na wielu grupach.

Polecenie 2

Napiszmy kwerendę, która poda liczbę uczniów w poszczególnych klasach.

Uzyskanie takiej informacji wymaga sięgnięcia tylko do jednej tabeli, czyli *uczniowie*. W klauzuli `SELECT` wybieramy pole *klasa*, według którego grupujemy rekordy, oraz pole *id_ucz*, które zliczamy. W ten sposób otrzymamy tabelę z liczbą uczniów w poszczególnych klasach.

```
1 SELECT klasa, COUNT(id_ucz) liczba_uczniow
2 FROM uczniowie
3 GROUP BY klasa;
```

Początkowe zwrócone rekordy:

1	1a	30
2	1b	29
3	1c	30
4	1d	27
5	1e	32

Polecenie 3

Napiszmy kwerendę, która wyświetli średnią ocen poszczególnych klas.

Problem sugeruje, że należy pogrupować uczniów według klas, a następnie policzyć średnią ich ocen. Źródłem kwerendy będzie zatem tabela *uczniowie*. Z niej wybieramy pole *klasa*, według którego będą grupowane rekordy. Drugą wymaganą tabelą będą *oceny*, z której dodajemy pole *ocena* i nakładamy na nie funkcję `AVG()` wyliczającą średnią. Zapis w języku SQL:

```
1 SELECT klasa, AVG(ocena) srednia_ocen
2 FROM oceny, uczniowie
3 USING (id_ucz)
4 GROUP BY klasa;
```

Do złączenia tabel użyliśmy klauzuli `USING`, która może zastąpić klauzulę `JOIN ON`, jeżeli pola tworzące relację mają taką samą nazwę. Wyświetlone średnie ocen można zaokrąglić przy użyciu funkcji `ROUND()`, której argumentami są średnia ocena wyliczona przez funkcję `AVG()` oraz precyzja. Początek kwerendy będzie miał wtedy postać:

```
1 SELECT klasa, ROUND(AVG(ocena), 2) srednia_ocen
```

Początkowe zwrócone rekordy:

```
1 1a  2.96
2 1b  3
3 1c  3.03
4 1d  2.99
5 1e  2.96
```

Polecenie 4

Napiszmy zapytanie, które zwróci nazwę przedmiotu z najniższą średnią ocen.

Źródłem danych w zapytaniu będą tabele *oceny* i *przedmioty*. Dane trzeba pogrupować według pola *nazwa* z tabeli *przedmioty*. W klauzuli `SELECT` obok nazwy przedmiotu

umieścimy funkcję `AVG()`, której prześlemy jako argument pole *ocena*. Rekordy uporządkujemy rosnąco według średnich ocen, a ich liczbę ograniczymy do jednego.

Kod SQL prezentuje się następująco:

```
1 SELECT przedmioty.nazwa Przedmiot, AVG(ocena) Średnia
2 FROM oceny, przedmioty
3 WHERE oceny.id_prz = przedmioty.id_prz
4 GROUP BY przedmioty.nazwa
5 ORDER BY Średnia ASC
6 LIMIT 1
```

Operacja grupowania nie wyklucza sortowania w klauzuli `ORDER BY` ani ograniczania liczby zwracanych rekordów w klauzuli `LIMIT`.

Wynik kwerendy:

```
1 informatyka 2.93494228751312
```

Ważne!

Sortowanie następuje po grupowaniu. Tym samym w klauzuli sortowania można podać pola, według których następuje grupowanie, lub pola, na których wykonywana jest funkcja agregująca.

Użycie podzapytań

Polecenie 5

Napiszmy zapytanie, które zwróci średnie ocen z poszczególnych przedmiotów podanej klasy, np. 1a.

Rozwiązanie zaczniemy od skonstruowania zapytania, które pobierze wszystkie oceny uczniów z podanej klasy. W wynikach kwerendy chcemy mieć nazwę klasy, nazwę przedmiotu oraz oceny, dlatego w klauzuli `SELECT` umieszczamy pola *klasa*, *nazwa* i *ocena*. Każde z tych pól jest w innej tabeli, warunki złączenia definiujemy za pomocą pól *id_prz* oraz *id_ucz*, które tworzą relację, w klauzuli `WHERE`. Dodajemy również warunek pozwalający wybrać rekordy klasy 1a. Kod SQL:

```
1 SELECT klasa, nazwa, ocena
```

```

2 FROM oceny, przedmioty, uczniowie
3 WHERE oceny.id_prz = przedmioty.id_prz
4     AND oceny.id_ucz = uczniowie.id_ucz
5     AND klasa = '1a';

```

Wynikiem kwerendy będzie tabela ocen podanej klasy. Początkowe rekordy:

```

1 1a  niemiecki  4
2 1a  angielski  3
3 1a  niemiecki  3
4 1a  niemiecki  4
5 1a  fizyka     4

```

Teraz zaprojektujemy końcową kwerendę, która pogrupuje oceny według nazw przedmiotów, obliczy średnią dla każdej grupy i wyświetli nazwy przedmiotów oraz średnie ocen. Poprzednie zapytanie umieścimy w klauzuli FROM jako źródło końcowej kwerendy.

Kod SQL przyjmie postać:

```

1 SELECT nazwa, AVG(ocena)
2 FROM
3     (SELECT klasa, nazwa, ocena
4       FROM oceny, przedmioty, uczniowie
5       WHERE oceny.id_prz = przedmioty.id_prz
6             AND oceny.id_ucz = uczniowie.id_ucz
7             AND klasa = '1a')
8 GROUP BY nazwa;

```

Ważne!

Kwerendy wykorzystywane wewnątrz innej kwerendy nazywamy podzapytaniem. Po podzapytaniach nie stawiamy średników, są one wykonywane w pierwszej kolejności.

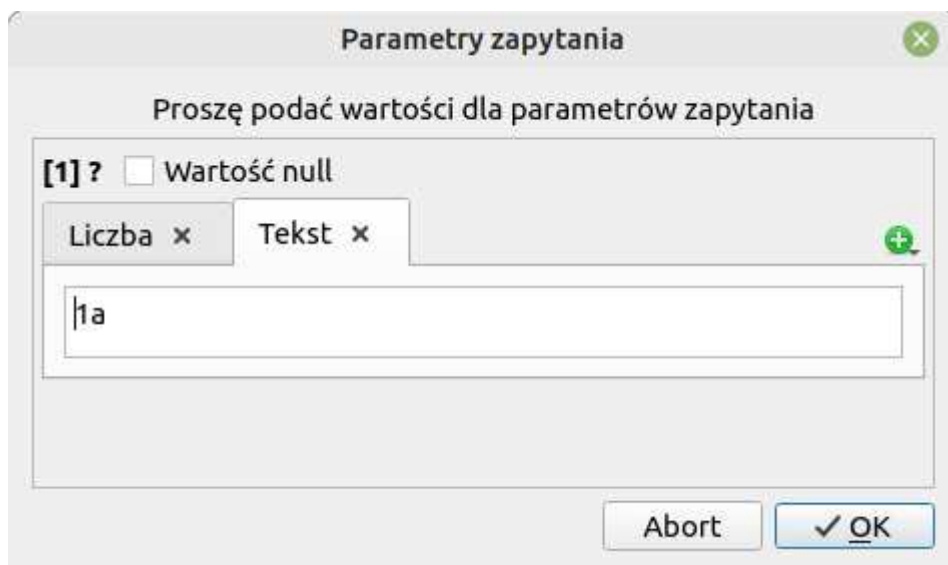
Ćwiczenie 2

Dodaj do ostatniej kwerendy sortowanie malejące wyników według średniej ocen z przedmiotów.

Kwerenda parametryczna

Ostatnia kwerenda byłaby o wiele bardziej użyteczna, gdyby pozwalała na podawanie nazwy klasy przed wykonaniem. Taką możliwość łatwo uzyskać, przekształcając zapytanie w [kwerendę parametryczną](#). Wystarczy zamiast nazwy klasy wpisać znak zapytania. Zapytanie przyjmie następującą postać:

```
1 SELECT nazwa, AVG(ocena)
2 FROM
3     (SELECT klasa, nazwa, ocena
4      FROM oceny, przedmioty, uczniowie
5      WHERE oceny.id_prz = przedmioty.id_prz
6            AND oceny.id_ucz = uczniowie.id_ucz
7            AND klasa = ?)
8 GROUP BY nazwa;
```



Okno Parametry zapytania w programie SQLiteStudio.

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-ND 3.0.

Podane zapytanie wykona się w programie SQLiteStudio, który wyświetli okno dialogowe pozwalające podać odpowiednią wartość. Nazwę klasy należy wprowadzić w zakładce Tekst i kliknąć OK.

W wierszu poleceń bazy SQLite lub w zapytaniach wykonywanych przy użyciu skryptu Pythona kwerenda nie zadziała.

Filtrowanie grup rekordów

Polecenie 6

Napiszmy zapytanie, które zwróci nazwy i średnią ocen klas, jeżeli średnia ocen będzie większa od 3.

Problem postawiony w zadaniu podobny jest do zadania, w którym chcieliśmy otrzymać średnie oceny wszystkich klas. Rozwiązanie opiera się więc na poprzedniej kwerendzie:

```
1 SELECT klasa, AVG(ocena) srednia_ocen
2 FROM oceny, uczniowie
3 USING(id_ucz)
4 GROUP BY klasa
5 HAVING srednia_ocen > 3;
```

Nowością jest klauzula HAVING, która może występować tylko po klauzuli grupującej. Używamy jej wtedy, kiedy nakładamy jakieś warunki na grupy rekordów.

Zwrócone rekordy:

```
1 1c 3.03068783068783
2 2a 3.02972195589645
3 2d 3.04003724394786
4 3b 3.01615074024226
5 3e 3.02485795454545
```

Warunki z klauzuli WHERE nakładane są przed grupowaniem. Tym samym odfiltrowane rekordy nie są brane pod uwagę przy grupowaniu. W powyższym przykładzie moglibyśmy wykluczyć z otrzymanego zestawienia, np. klasy o profilu matematycznym. Na potrzeby przykładu założmy, że są to wszystkie klasy z literą „a” w nazwie:

```
1 SELECT klasa, AVG(ocena) srednia_ocen
2 FROM oceny, uczniowie
3 USING(id_ucz)
4 WHERE klasa NOT LIKE '%a'
5 GROUP BY klasa
6 HAVING srednia_ocen > 3;
```

Zwrócone rekordy:

```
1 1c 3.03068783068783
2 2d 3.04003724394786
3 3b 3.01615074024226
4 3e 3.02485795454545
```

Słownik

grupowanie

tworzenie zbiorów rekordów na podstawie takich samych wartości z podanej kolumny;
grupowanie zazwyczaj poprzedza zastosowanie funkcji agregującej, która przeprowadza obliczenia na grupie rekordów

kwerenda parametryczna

kwerenda, w której wartości w wyrażeniach warunkowych są parametrem podawanym przed wykonaniem zapytania; parametry oznaczane są znakami zapytania

Prezentacja multimedialna

Polecenie 1

Zapoznaj się z prezentacją, która wyjaśnia działanie zapytań grupujących z wykorzystaniem programu SQLiteStudio i operacji wykonywanych w arkuszu kalkulacyjnym.

W pliku `uczniowie2.db` zamieszczono bazę danych SQLite3. Pobierz plik, zapisz i otwórz w programie SQLiteStudio.

Baza danych SQLite3 `uczniowie2.db`.

Plik o rozmiarze 832.00 KB w języku polskim

W pliku tekstowym `uczniowie.csv` zamieszczono dane uczniów zapisane w formacie CSV. Każdy wiersz zawiera dane dla kolumn: *id_ucz*, *nazwisko*, *imie*, *ulica*, *nr_domu* i *klasa*. Pobierz go i zapisz w wybranym katalogu.

Plik CSV `uczniowie.csv`

Plik o rozmiarze 14.67 KB w języku polskim

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1Zht7J1t>

W prezentacji porównamy działanie mechanizmów grupowania w języku SQL (z wykorzystaniem programu SQLiteStudio) oraz w tabeli przestawnej arkusza kalkulacyjnego.

Naszym zadaniem jest otrzymanie zestawienia liczby uczniów mieszkających na poszczególnych ulicach. Wyniki chcemy ograniczyć do ulic, na których mieszka 15 lub więcej uczniów. Dodatkowo chcemy je uporządkować malejąco według liczby uczniów i rosnąco według nazw ulic.

1

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1Zht7J1t>

Pierwszym krokiem, który trzeba wykonać przed policzeniem uczniów mieszkających na tej samej ulicy, jest pogrupowanie rekordów z tabeli *uczniowie* według ulic.

Zadanie to możemy zilustrować, wykonując sortowanie danych w aplikacji SQLiteStudio. Dwa razy klikamy tabelę *uczniowie*, a następnie zakładkę Dane. Nagłówek pola *ulica* klikamy prawym klawiszem i wybieramy Zdefiniuj kolumny, po których sortować. W oknie Sortuj wg. Kolumny zaznaczamy pole *ulica* i klikamy OK.

	id_ucz	nazwisko	imie	ulica	nr_domu	klasa
1	35	Dawidson	Grazyna	Armii Krajowej	17	1b
2	56	Wasowicz	Joanna	Armii Krajowej	71	1b
3	75	Misiak	Urszula	Armii Krajowej	27	1c
4	118	Baranowska	Ewelina	Armii Krajowej	12	1e
5	119	Bator	Ewelina	Armii Krajowej	99	1e
6	121	Berezinski	Janina	Armii Krajowej	89	1e
7	135	Politowska	Sylwia	Armii Krajowej	42	1e
8	184	Hupa	Magdalena	Armii Krajowej	20	2b
9	195	Patryarcha	Ewa	Armii Krajowej	98	2b
10	238	Barucka	Dorota	Armii Krajowej	51	2d
11	290	Zymek	Barbara	Armii Krajowej	51	2e
12	293	Czerwinski	Adam	Armii Krajowej	82	3a
13	312	Liszczyk	Joanna	Armii Krajowej	3	3a
14	319	Wolniak	Teresa	Armii Krajowej	50	3b
15	331	Wieniewska	Joanna	Armii Krajowej	18	3b
16	352	Nawizowska	Jadwiga	Armii Krajowej	47	3c
17	364	Moryn	Mariola	Armii Krajowej	89	3c
18	375	Marciniak	Renata	Armii Krajowej	84	3d
19	379	Klezak	Ewa	Armii Krajowej	15	3d
20	53	Staniec	Zofia	Bacewicz	56	1b
21	86	Wachowicz	Magdalena	Bacewicz	94	1c
22	180	Bak	Krzysztof	Bacewicz	59	2b
23	228	Poznanska	Dorota	Bacewicz	53	2c
24	232	Skalik	Anna	Bacewicz	78	2c
25	254	Pala	Iwona	Bacewicz	22	2d

3

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1Zht7J1t>

Wykonanie operacji sortowania pokazuje, w jaki sposób powstają grupy rekordów, dla których wartość w polu *ulica* jest taka sama. W języku SQL grupowanie rekordów realizuje klauzula GROUP BY, której działanie i użycie pokażemy dalej.

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1Zht7J1t>

Na podobnej zasadzie jak kwerenda grupująca z funkcją agregującą działa tabela przestawna w arkuszu kalkulacyjnym. Zimportujmy zatem plik *uczniowie.csv* do wybranego arkusza kalkulacyjnego.

Tworzymy pusty skoroszyt, następnie w programie LibreOffice Calc wybieramy Plik | Otwórz, natomiast w programie Microsoft Excel klikamy wstążkę Dane i wybieramy polecenie Z tekstu. W otwartych oknach wskazujemy plik *uczniowie.csv*.

5

Importuj tekst - [uczniowie.csv]

Importuj

Zestaw znaków: Unicode (UTF-8)

Język: Domyślny - Polski

Od wiersza: 1

Opcje separatora

☐ Stała szerokość ☒ Rozdzielony

☐ Tabulator ☐ Przecinek ☒ Średnik ☐ Spacja ☐ Inny

☐ Scal separatory ☐ Spacje wiodące Ogranicznik ciągu: "

Inne opcje

☒ Formatuj pola w cudzysłowie jako tekst ☐ Identyfikuj liczby specjalne

☐ Szacuj formuły

Pola

Typ kolumny:

	Standard	Standardowe	Standardowe	Standardowe	Standard	Standard
1	id_ucz	nazwisko	imię	ulica	dom	id_klasy
2	1	Baczynska	Danuta	Kiedrzyńska	42	1a
3	2	Berdysowa	Maria	wyzwolenia	80	1a
4	3	Byczkowska	Ewa	Ludowa	83	1a
5	4	Cygan	Aneta	Sportowa	89	1a
6	5	Cygan	Henryka	Sportowa	88	1a
7	6	Czarnoleska	Aneta	Fiełdorfa	39	1a
8	7	Derda	Beata	Porajska	79	1a

Pomoc Anuluj OK

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1Zht7J1t>

Podczas importu w obydwu programach wybieramy zestaw znaków Unicode (UTF-8), jako separator zaznaczamy Średnik.

Powyżej zamieszczono zrzut okna dialogowego z programu LO Calc.

6

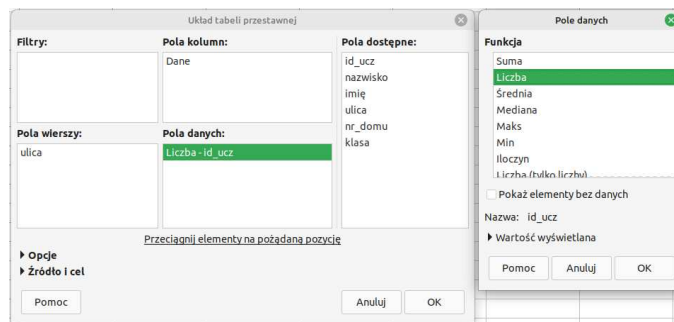
Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1Zht7J1t>

Po imporcie zaznaczamy komórkę A1 i uruchamiamy kreator tabeli przestawnej. W LO Calc wybieramy Dane | Tabela przestawna | Wstaw lub edytuj. W programie MS Excel klikamy zakładkę Wstawianie i wybieramy polecenie Tabela przestawna. Potwierdzamy poprawność zaznaczenia tabeli jako źródła danych.

Wykonana operacja to wybranie źródła danych. W zapytaniach SQL źródło danych podajemy w klauzuli FROM. W omawianym przypadku będzie ona wyglądała tak:

```
1 FROM uczniowie
```



7

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

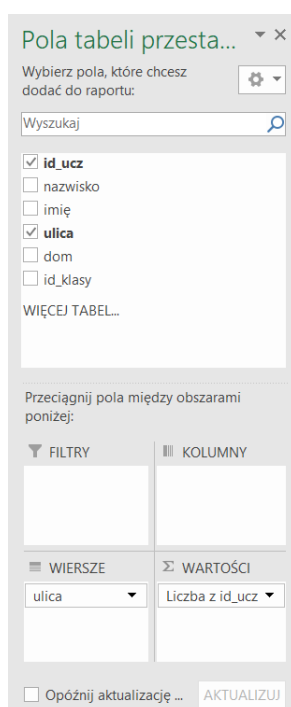
Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1Zht7J1t>

W programie LO Calc pola, które przeciągamy do Pola kolumn lub Pola wierszy zostaną pogrupowane.

Przeciągamy więc pole *ulica* do Pola wierszy. Do Pola danych przeciągamy pole *id_ucz*, które zostanie użyte przez wybraną funkcję agregującą. Domyślną funkcję Suma zmieniamy po dwukrotnym kliknięciu nazwy Suma – *id_ucz* na funkcję Liczba.

8



Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1Zht7J1t>

W programie MS Excel pole *ulica* przeciągamy do komórki WIERKSZE, pole *id_ucz* do komórki WARTOŚCI. Następnie klikamy listę Suma z *id_ucz* i wybieramy Ustawienia pola wartości, zmieniamy typ obliczeń na Licznik.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1Zht7J1t>

9

Projektowanie tabeli przestawnej polega na wybraniu pól, z których chcemy pobierać i grupować dane, oraz wskazaniu funkcji wykonującej obliczenia na pogrupowanych polach.

W kwerendach SQL operacje te wykonywane są przez klauzule SELECT (wybieranie pól) i GROUP BY (grupowanie) oraz wybraną funkcję agregującą. Nasze zapytanie przybierze więc postać:

```
1 SELECT ulica,  
   COUNT(id_ucz)  
2 FROM uczniowie  
3 GROUP BY ulica;  
4
```

Po wykonaniu zapytania w programie SQLiteStudio zobaczymy początkowe rekordy:

1	Armii Krajowej	19
2	Bacewicza	9
3	Baczynskiego	8
4	Broniewskiego	14
5	Fieldorfa	12

Etykiety wierszy	Liczba z id_ucz
Armii Krajowej	19
Gajcego	18
Iwaszkiewicza	30
Kiedrzyńska	34
Kwiatkowskiego	17
Ludowa	23
Mickiewicza	15
PCK	18
Starzyskiego	17
Wyzwolenia	16
Suma końcowa	207

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1Zht7J1t>

Na uzyskane zestawienie nałożymy filtr, który ograniczy zestawienie do ulic, na których mieszka więcej niż 15 uczniów.

W programie LO Calc klikamy komórkę B1 (*Liczba - id_ucz*), wybieramy Dane | Więcej filtrów | Filtr standardowy i ustawiamy *Liczba - id_ucz* ≥ 15 .

W programie MS Excel klikamy przycisk przy polu Etykiety wierszy, wybieramy Filtry wartości | Większe niż lub równe i ustawiamy *Liczba z id_ucz* / jest większe niż lub równe / 15.

Powyżej zamieszczono zrzut przefiltrowanej tabeli z programu MS Excel.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1Zht7J1t>

W kwerendach SQL do nakładania warunków na zgrupowane i przeliczone dane służy klauzula HAVING, którą dodajemy do kwerendy:

```
1 SELECT ulica,  
   COUNT(id_ucz)  
   liczba_uczniow  
2 FROM uczniowie  
3 GROUP BY ulica  
4 HAVING liczba_uczniow >=  
   15;  
5
```

Dla czytelności, aby nie powtarzać klauzuli COUNT(id_ucz), użyliśmy aliasu *liczba_uczniow*.

Początkowe zwrócone rekordy:

1	Armii Krajowej	19
2	Gajcego	18
3	Iwaszkiewicza	30
4	Kiedrzyńska	34
5	Kwiatkowskiego	17

ulica	Liczba - id_ucz
Kiedrzynska	34
Iwaszkiewicza	30
Ludowa	23
Armii Krajowej	19
Gajcego	18
PCK	18
Kwiatkowskiego	17
Starzynskiego	17
Wyzwolenia	16
Mickiewicza	15
Razem Wynik	414

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1Zht7J1t>

Uzyskane zestawienie dodatkowo
uporządkujemy.

W programie LO Calc zaznaczamy zakres
A2 : B32 i za pomocą polecenia Dane | Sortuj
sortujemy malejąco według Kolumna B i
rosnąco według Kolumna A.

W programie MS Excel zaznaczamy etykiety pól
z danymi bez wiersza podsumowania, czyli
zakres A3 : B13, kopiujemy i wklejamy np. do
komórki D3, na zakładce Dane wybieramy
Sortuj, sortujemy malejąco według pola
Liczba z id_ucz, dodajemy poziom i sortujemy
rosnąco według pola *Etykiety wierszy*.

Powyżej zamieszczono zrzut przedstawiający
uporządkowaną tabelę w programie LO Calc.

<https://zpe.gov.pl/b/P1Zht7J1t>

W zapytaniach SQL za sortowanie danych odpowiada klauzula ORDER BY. Dodajemy ją do zapytania, które ostatecznie będzie wyglądało następująco:

```
1 SELECT ulica,  
   COUNT(id_ucz)  
   liczba_uczniow  
2 FROM uczniowie  
3 GROUP BY ulica  
4 HAVING liczba_uczniow >=  
   15  
5 ORDER BY liczba_uczniow  
   DESC, ulica ASC;  
6
```

14

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1Zht7J1t>

Po wykonaniu kwerendy w programie SQLiteStudio zostaną zwrócone następujące rekordy:

1	Kiedrzynska	34
2	Iwaszkiewicza	30
3	Ludowa	23
4	Armii Krajowej	19
5	Gajcego	18
6	PCK	18
7	Kwiatkowskiego	17
8	Starzynskiego	17
9	Wyzwolenia	16
10	Mickiewicza	15

Polecenie 2

Po przeanalizowaniu prezentacji multimedialnej przygotuj notatkę ze swoimi spostrzeżeniami dotyczącymi kwerend grupujących.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Uzupełnij zdanie.

Kwerenda grupująca rekordy .

☐ sortuje je według podanego kryterium

☐ dodaje wyrażenie warunkowe w klauzuli WHERE

☐ wybiera rekordy według powtarzających się wartości wskazanego pola

Ćwiczenie 2



Wskaż dokończenie zdania.

Funkcje agregujące mogą być stosowane...

☐ bez klauzuli GROUP BY, ale tylko w podzapytaniach.

☐ tylko z klauzulą GROUP BY.

☐ z klauzulą GROUP BY lub bez.

Ćwiczenie 3



Wskaż, kiedy wykonywane są funkcje agregujące.

☐ przed grupowaniem rekordów lub po nim – w zależności od wyrażeń warunkowych

☐ przed grupowaniem rekordów

☐ po grupowaniu rekordów

Ćwiczenie 4



Uporządkuj podane klauzule SQL.

FROM



LIMIT



WHERE



GROUP BY



SELECT



HAVING



ORDER BY



Ćwiczenie 5



Wskaż wszystkie właściwe dokończenia zdania.

W kwerendzie grupującej w klauzuli porządkującej rekordy można podać...

☐ pole będące argumentem funkcji agregującej.

☐ dowolne pole z kwerendy.

☐ pole występujące w klauzuli warunkowej.

☐ pole, według którego następuje grupowanie.

Ćwiczenie 6



Wskaż, czego można używać w kwerendzie grupującej.

- ☐ podzapytań
- ☐ złączeń tabel
- ☐ tylko prostych wyrażeń warunkowych
- ☐ złożonych wyrażeń warunkowych

Ćwiczenie 7



Wskaż dokończenie zdania.

Nałożenie dodatkowych warunków na grupy rekordów...

- ☐ jest możliwe w klauzuli WHERE i/lub HAVING.
- ☐ nie jest możliwe.
- ☐ jest możliwe w klauzuli WHERE.
- ☐ jest możliwe tylko w klauzuli HAVING.

Ćwiczenie 8



Dokończ zdanie.

Kwerenda parametryczna to...

- ☐ każde zapytanie, w którym w wyrażeniu warunkowym użyto znaku zapytania.
- ☐ zapytanie z jedną niewiadomą.
- ☐ zapytanie, w którym wartości kryteriów filtrowania trzeba podać dopiero w momencie wykonywania.

Do wykonania ćwiczeń 9 i 10 wykorzystaj bazę danych `uczniowie2.db` zamieszczoną w sekcji „Przeczytaj”.

Ćwiczenie 9



Stwórz kwerendę pozwalającą na dokończenie zdania.

Podaj id ucznia z najwyższą średnią:

Ćwiczenie 10



Stwórz kwerendę pozwalającą na dokończenie zdania.

Podaj nazwisko nauczyciela, który wystawił najmniej ocen:

Dla nauczyciela

Autor: Robert Bednarz

Przedmiot: Informatyka

Temat: Instrukcje grupowania w języku SQL, etap II

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

4) przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:

d) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz sieciową aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie,

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Wyjaśnisz, na czym polega grupowanie rekordów.
- Użyjesz w praktyce grupowania rekordów i funkcji agregujących.
- Zestawisz i przeanalizujesz dane z wielu tabel.
- Wykorzystasz w kwerendach podzapytania.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie SQLiteStudio;
- oprogramowanie Microsoft Excel 2010, LibreOffice Calc 4.1 lub wybrany odpowiednik.

Przebieg lekcji

Przed lekcją:

1. W razie potrzeby nauczyciel prosi uczniów o powtórzenie wiadomości zaprezentowanych w pierwszym e-materiale z serii.
2. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Instrukcje grupowania w języku SQL, etap II”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Nauczyciel wyświetla i odczytuje temat lekcji oraz cele zajęć. Prosi uczniów o sformułowanie kryteriów sukcesu.

2. **Rozpoznanie wiedzy uczniów.** Uczniowie tworzą pytania dotyczące tematu zajęć, na które odpowiedzą w trakcie lekcji.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel ocenia, na podstawie informacji na platformie, stan przygotowania uczniów do zajęć. Jeżeli jest ono niewystarczające, prosi wybraną osobę o przedstawienie najważniejszych informacji z sekcji „Przeczytaj”. W kolejnym kroku uczniowie w parach wykonują ćwiczenia nr 1 i 2. Chętne lub wybrane pary prezentują swoje rozwiązania.
2. **Praca z multimediami.** Uczniowie zapoznają się z multimediami z sekcji „Prezentacja multimedialna”. Nauczyciel wyjaśnia ewentualne niezrozumiałe kwestie.
3. **Ćwiczenie umiejętności.** Uczniowie indywidualnie wykonują ćwiczenia nr 1–8.

Faza podsumowująca:

1. Uczniowie przygotowują notatkę podsumowującą zajęcia, wykonując polecenie 2 z sekcji „Prezentacja multimedialna”.
2. Nauczyciel ponownie wyświetla na tablicy temat lekcji zawarty w sekcji „Wprowadzenie” i inicjuje krótką rozmowę na temat zrealizowanych celów (czego uczniowie się nauczyli).

Praca domowa:

1. Uczniowie wykonują ćwiczenia 9 i 10 z sekcji „Sprawdź się”.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla oprogramowania SQLiteStudio.
- Oficjalna dokumentacja techniczna dla oprogramowania Microsoft Excel 2010 (lub nowszej wersji), LibreOffice Calc 4.1 lub wybranego odpowiednika.

Wskazówki metodyczne:

- Treści w sekcji „Prezentacja multimedialna” można wykorzystać na lekcji jako podsumowanie i utrwalenie wiedzy uczniów.