

Ben Forta

SQL Server i T-SQL

W MGNIE NIU OKA

OPANUJ JĘZYK ZAPYTAŃ W  10 MINUT DZIENNIE



wydanie II

Helion 

Tytuł oryginału: Sams Teach Yourself Microsoft® SQL Server T-SQL in 10 Minutes, Second Edition

Tłumaczenie: Patryk Wierchoń

Projekt okładki: Studio Gravite / Olsztyn; Obarek, Pokoński, Pazdrijowski, Zaprucki

ISBN: 978-83-283-3565-3

Authorized translation from the English language edition: SAMS TEACH YOURSELF MICROSOFT SQL SERVER T-SQL IN 10 MINUTES, SAMS TEACH YOURSELF, Second Edition; ISBN 0672337924; by Ben Forta; published by Pearson Education, Inc, publishing as SAMS Publishing.

Copyright © 2017 by Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education Inc.

Polish language edition published by HELION SA. Copyright © 2017.

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.

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/sqlsts>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

Spis treści

	O autorze	9
	Wprowadzenie	11
Lekcja 1.	Zrozumieć SQL	15
	Podstawy baz danych	15
	Język SQL	20
	Ćwicz	21
	Podsumowanie	21
Lekcja 2.	Początek pracy z SQL Serverem	23
	Czym jest SQL Server?	23
	Rozpoczynamy pracę z SQL Serverem i T-SQL	25
	Przygotowanie do lekcji	27
	Podsumowanie	27
Lekcja 3.	Praca z SQL Serverem	29
	Nawiązywanie połączenia	29
	Szybkie wprowadzenie do SQL Server Management Studio	30
	Tworzenie i wypełnianie danymi przykładowych tabel	31
	Wybór bazy danych	32
	Pobieranie informacji o bazie i tabelach	33
	Podsumowanie	36
Lekcja 4.	Pobieranie danych	37
	Klauzula SELECT	37
	Pobieranie pojedynczych kolumn	37
	Pobieranie wielu kolumn	39
	Pobieranie wszystkich kolumn	40
	Pobieranie unikatowych wierszy	41
	Ograniczanie wyników	42
	Używanie kwalifikowanych nazw tabel	44
	Podsumowanie	44

Lekcja 5.	Sortowanie otrzymywanych danych	45
	Sortowanie danych	45
	Sortowanie względem wielu kolumn	47
	Określenie kierunku sortowania	48
	Podsumowanie	50
Lekcja 6.	Filtrowanie danych	51
	Stosowanie frazy WHERE	51
	Operatory frazy WHERE	52
	Podsumowanie	57
Lekcja 7.	Zaawansowane filtrowanie danych	59
	Łączenie fraz WHERE	59
	Operator IN	63
	Operator NOT	64
	Podsumowanie	65
Lekcja 8.	Filtrowanie za pomocą znaków wieloznacznych	67
	Korzystanie z operatora LIKE	67
	Wskazówki dotyczące używania znaków wieloznacznych	72
	Podsumowanie	72
Lekcja 9.	Tworzenie pól obliczanych	73
	Pojęcie pól obliczanych	73
	Konkatenacja pól	74
	Przeprowadzanie obliczeń matematycznych	77
	Podsumowanie	79
Lekcja 10.	Modyfikacja danych za pomocą funkcji	81
	Czym są funkcje?	81
	Stosowanie funkcji	82
	Podsumowanie	91
Lekcja 11.	Funkcje agregujące	93
	Funkcje agregujące	93
	Agregacja tylko różnorodnych wartości	99
	Łączenie funkcji agregujących	100
	Podsumowanie	101

Lekcja 12.	Grupowanie danych	103
	Omówienie grupowania danych	103
	Tworzenie grup	103
	Filtrowanie grup	105
	Grupowanie i sortowanie	107
	Kolejność fraz instrukcji SELECT	109
	Podsumowanie	109
Lekcja 13.	Zapytania zagnieżdżone	111
	Zagnieżdżanie zapytań	111
	Filtrowanie na podstawie zapytań zagnieżdżonych	111
	Zapytania zagnieżdżone jako pola obliczane	115
	Sprawdzanie istnienia rekordów za pomocą zapytań zagnieżdżonych	117
	Podsumowanie	119
Lekcja 14.	Łączenie tabel	121
	Czym są złączenia?	121
	Tworzenie złączeń	123
	Podsumowanie	131
Lekcja 15.	Tworzenie rozbudowanych złączeń	133
	Stosowanie aliasów tabel	133
	Typy złączeń	134
	Złączenia i funkcje agregujące	139
	Złączenia i ich warunki	140
	Podsumowanie	141
Lekcja 16.	Łączenie zapytań	143
	Łączenie zapytań	143
	Tworzenie unii	143
	Podsumowanie	148
Lekcja 17.	Wyszukiwanie pełnotekstowe	149
	Zrozumieć wyszukiwanie pełnotekstowe	149
	Przygotowanie wyszukiwania pełnotekstowego	150
	Wyszukiwanie pełnotekstowe	153
	Podsumowanie	159

Lekcja 18.	Wstawianie danych	161
	Wstawianie danych	161
	Wstawianie całych wierszy	161
	Wstawianie wielu wierszy	165
	Wstawianie pobranych danych	166
	Raportowanie wprowadzanych danych	168
	Podsumowanie	169
Lekcja 19.	Aktualizacja i usuwanie danych	171
	Aktualizacja danych	171
	Usuwanie danych	173
	Raportowanie usuwanych i aktualizowanych danych	174
	Wskazówki związane z aktualizacją lub usuwaniem danych	175
	Podsumowanie	175
Lekcja 20.	Tworzenie i modyfikacja tabel	177
	Tworzenie tabel	177
	Aktualizacja tabel	183
	Usuwanie tabel	185
	Zmiana nazwy tabeli	185
	Podsumowanie	186
Lekcja 21.	Stosowanie perspektyw	187
	Perspektywy	187
	Tworzenie perspektyw	189
	Podsumowanie	194
Lekcja 22.	Programowanie w T-SQL	195
	Programowanie w T-SQL	195
	Używanie zmiennych	195
	Przetwarzanie warunkowe	201
	Grupowanie wyrażeń	203
	Pętle	205
	Podsumowanie	206
Lekcja 23.	Korzystanie z procedur składowanych	207
	Procedury składowane	207
	Dlaczego warto używać procedur składowanych	208
	Używanie procedur składowanych	209
	Podsumowanie	215

Lekcja 24.	Kursory	217
	Działanie kursorów	217
	Praca z kursorami	217
	Podsumowanie	223
Lekcja 25.	Wyzwalacze	225
	Zrozumieć wyzwalacze	225
	Korzystanie z wyzwalaczy	228
	Podsumowanie	230
Lekcja 26.	Zarządzanie transakcjami	231
	Działanie transakcji	231
	Sterowanie transakcjami	233
	Podsumowanie	236
Lekcja 27.	Zastosowanie XML i JSON	237
	Korzystanie z obsługi XML w SQL Serverze	237
	Korzystanie z obsługi JSON w SQL Serverze	244
	Pobieranie danych jako JSON	245
	Funkcje JSON	247
	Podsumowanie	247
Lekcja 28.	Globalizacja i lokalizacja	249
	Zestawy i sekwencje sortowania znaków	249
	Praca z sekwencjami sortowania znaków	249
	Obsługa wielkości znaków	251
	Obsługa Unicode	253
	Podsumowanie	255
Lekcja 29.	Zarządzanie bezpieczeństwem	257
	Kontrola dostępu	257
	Zarządzanie użytkownikami	258
	Zarządzanie dostępami	260
	Podsumowanie	262
Lekcja 30.	Zwiększanie wydajności	263
	Poprawianie wydajności	263
	Podsumowanie	265

Dodatek A	Przykładowe tabele	267
	Omówienie przykładowych tabel	267
Dodatek B	Składnia zapytania T-SQL	273
	BEGIN TRANSACTION	273
	ALTER TABLE	273
	COMMIT TRANSACTION	274
	CREATE INDEX	274
	CREATE LOGIN	274
	CREATE PROCEDURE	274
	CREATE TABLE	274
	CREATE VIEW	275
	DELETE	275
	DROP	275
	INSERT	275
	INSERT SELECT	276
	ROLLBACK TRANSACTION	276
	SAVE TRANSACTION	276
	SELECT	276
	UPDATE	277
Dodatek C	Typy danych T-SQL	279
	Tekstowe typy danych	279
	Numeryczne typy danych	281
	Typy danych daty i czasu	281
	Binarne typy danych	282
Dodatek D	Słowa kluczowe i zarezerwowane języka T-SQL	283
	Skorowidz	287

Lekcja 17.

Wyszukiwanie pełnotekstowe

W tej lekcji dowiesz się, jak wykorzystać możliwości wyszukiwania pełnotekstowego SQL Servera do tworzenia złożonych zapytań

Zrozumieć wyszukiwanie pełnotekstowe

W lekcji 8., „Filtrowanie za pomocą znaków wieloznacznych”, poznałeś słowo kluczowe LIKE, wykorzystywane do porównywania tekstu (oraz części tekstu) przez zastosowanie znaków wieloznacznych. W ten sposób możliwe jest wyszukiwanie wierszy zawierających określony tekst lub jego część, niezależnie od tego, w której kolumnie znajduje się szukana wartość.

Ten mechanizm jest bardzo pomocny, lecz ma istotne ograniczenia:

- ▶ **Wydajność** — Wyszukiwanie za pomocą znaków wieloznacznych zazwyczaj wymaga sprawdzenia każdego wiersza w tabeli (indeksy są rzadko wykorzystywane). W związku z tym takie wyszukiwanie może być bardzo czasochłonne dla dużej liczby rekordów.
- ▶ **Jawna kontrola** — Korzystając ze znaków wieloznacznych, ciężko dokładnie skontrolować, co jest porównywane, a co nie jest (czasem w ogóle nie ma takiej możliwości). Przykładem jest wyszukiwanie słowa, które musi znaleźć się w tekście, słowa, które nie może znaleźć się w tekście, i słowa, które może się pojawić w zależności od tego, czy pierwsze słowo zostało znalezione, czy nie.
- ▶ **Inteligentne wyniki** — Chociaż wyszukiwanie za pomocą znaków wieloznacznych umożliwia bardzo elastyczne wyszukiwanie, nie pozwala na inteligentne pobieranie wyników. Na przykład wyszukiwanie danego słowa zwróci wszystkie wiersze zawierające słowo, ale nie można określić, czy wyszukiwane słowo pojawia się w danym rekordzie raz, czy też więcej razy (i tym samym stanowi potencjalnie lepsze dopasowanie). Z kolei wyszukiwanie określonego wyrazu nie odnajdzie wierszy zawierających podobne wyrazy.

Wszystkie te ograniczenia i nie tylko rozwiązywane są przez wyszukiwanie pełnotekstowe. W wyszukiwaniu pełnotekstowym SQL Server nie przegląda każdego rekordu, analizując słowo po słowie. SZBD tworzy indeks słów (w określonych kolumnach) i w nim przeprowadza wyszukiwanie. Dzięki temu SQL Server może określić, które słowa spełniają kryteria wyszukiwania (i w jakich wierszach się znajdują), które ich nie spełniają, jaka jest liczba wystąpień i tak dalej.

Przygotowanie wyszukiwania pełnotekstowego

Oto wymagania, jakie należy spełnić, aby przeprowadzić wyszukiwanie pełnotekstowe:

- ▶ Baza danych musi mieć włączoną możliwość wyszukiwania pełnotekstowego.
- ▶ Należy określić katalog, w którym przechowywane będą dane do wyszukiwania.
- ▶ Należy utworzyć indeks pełnotekstowy dla tabel i kolumn.

Po utworzeniu indeksu można wykonywać zapytania SELECT, korzystając ze słów kluczowych FREETEXT i CONTAINS.

Włączenie wyszukiwania pełnotekstowego

Po utworzeniu bazy danych należy włączyć wyszukiwanie pełnotekstowe przed rozpoczęciem jakiegokolwiek wyszukiwania z użyciem tej funkcjonalności. Służy do tego procedura składowana `sp_fulltext_database`. Procedura ta aktualizuje wybraną bazę danych, upewnij się więc, że wykonujesz ją na odpowiedniej bazie.

Wyrażenie wejściowe ▼

```
EXEC sp_fulltext_database 'enable';
```

Analiza ▼

Procedura `sp_fulltext_database` przyjmuje argument 'enable' (włącz) lub 'disable' (wyłącz), określający wsparcie wyszukiwania pełnotekstowego.

Uwaga

Włączenie wyszukiwania pełnotekstowego podczas utworzenia bazy

Podczas tworzenia nowej bazy w programie Microsoft SQL Server Management Studio możesz zaznaczyć opcję *Use Full-Text Indexing* (użyj indeksowania pełnotekstowego), która spowoduje automatyczne wywołanie wspomnianej procedury.

Wskazówka

Nie jesteś pewien, czy wyszukiwanie pełnotekstowe jest włączone?

Jeżeli nie wiesz, czy wyszukiwanie pełnotekstowe jest włączone, po prostu uruchom procedurę włączającą. Jeśli funkcjonalność była wyłączona, wywołanie procedury włączy ją. Jeżeli była już włączona, wywołanie procedury nic nie zmieni.

Tworzenie katalogu pełnotekstowego

Jak już wcześniej wytłumaczono, SQL Server przechowuje dane pełnotekstowe w katalogu (który należy utworzyć). Pojedynczy katalog może być wykorzystany do przechowywania wielu tabel i indeksów, więc jeśli w Twojej bazie istnieje taki katalog, możesz z niego skorzystać. Możesz również utworzyć własny katalog poleceniem `CREATE FULLTEXT CATALOG`:

Wyrażenie wejściowe ▼

```
CREATE FULLTEXT CATALOG katalog_naukasql;
```

Analiza ▼

W powyższym przykładzie utworzono katalog *katalog_naukasql* w domyślnej lokalizacji. Inną lokalizację można określić za pomocą polecenia `IN PATH`.

Tworzenie indeksu pełnotekstowego

Po utworzeniu katalogu możesz utworzyć indeksy pełnotekstowe. Tworzymy je wyrażeniem `CREATE FULLTEXT INDEX`, jak w poniższym przykładzie:

Wyrażenie wejściowe ▼

```
CREATE FULLTEXT INDEX ON prod_uwagi (tekst_uw LANGUAGE 'Polish')  
KEY INDEX pk_prod_uwagi  
ON katalog_naukasql;
```

Analiza ▼

W przykładzie utworzono indeks na tabeli *prod_uwagi*, indeksując kolumnę *tekst_uw*. Fraza `LANGUAGE` określa język, w którym wykonywane będzie wyszukiwanie pełnotekstowe. Wymagany jest też klucz, który w umożliwi jednoznaczną identyfikację rekordów, dlatego polecenie `KEY INDEX` określa nazwę klucza głównego tabeli. Fraza `ON` określa katalog do przechowywania danych pełnotekstowych — w naszym przypadku zastosowaliśmy katalog stworzony w poprzednim przykładzie.

Uwaga

Możliwe ostrzeżenia

Jeśli powyższe wyrażenie spowoduje pojawienie się błędów dotyczących kolumn, nie podejmuj się, wszystko i tak będzie działać prawidłowo.

Zaindeksować można wiele kolumn. Wystarczy określić ich nazwy po przecinku.

Uwaga**Definiowanie domyślnego katalogu**

W służącym do tworzenia nowego katalogu wyrażeniu `CREATE FULLTEXT CATALOG` możesz dodać frazę `AS DEFAULT`. Utworzony katalog stanie się w ten sposób katalogiem domyślnym dla kolejnych indeksów pełnotekstowych, dzięki czemu będziesz mógł ominąć frazę `ON` podczas ich tworzenia.

Po utworzeniu indeksu pełnotekstowego wszystkie dane zostały zaindeksowane. Wszystkie zapytania `INSERT`, `UPDATE` i `DELETE` wykonane na tabeli `prod_uwagi` spowodują aktualizację indeksu.

Wskazówka**Nie używaj indeksów pełnotekstowych przy importowaniu danych**

Aktualizowanie indeksów trwa określony czas. Niedługo, ale zawsze jest to dodatkowy czas. Aktualizowanie indeksów pełnotekstowych trwa jeszcze dłużej. Nie powinieneś włączać indeksowania pełnotekstowego, gdy importujesz dane do nowej tabeli. Lepszym rozwiązaniem będzie zaimportowanie wszystkich danych, a następnie zdefiniowanie indeksów. Dzięki temu import będzie znacznie szybszy (zaindeksowanie całej tabeli naraz zajmuje mniej czasu niż indeksowanie każdego importowanego wiersza z osobna).

Zarządzanie katalogami i indeksami

Katalogi i indeksy mogą być zmieniane poprzez `ALTER FULLTEXT` i usuwane poprzez `DROP FULLTEXT`. W praktyce rzadko stosuje się te wyrażenia, lecz istnieje jeden wyjątek. Jeżeli indeks staje się wadliwy (zwraca niespójne dane) lub jest zbyt wolny, pomóc może jego przebudowanie. Możesz zrobić to w następujący sposób:

Wyrażenie wejściowe ▼

```
ALTER FULLTEXT CATALOG katalog_naukasql REBUILD;
```

Analiza ▼

Wyrażenie usuwa i buduje od nowa indeksy w folderze, wymuszając pełne przeindeksowanie.

Więcej o istniejących widokach i indeksach możesz dowiedzieć się dzięki widokom systemowym:

Wyrażenie wejściowe ▼

```
SELECT * FROM sys.fulltext_catalogs;
```

Analiza ▼

Powyższe wyrażenie zwraca informacje o obecnie używanym katalogu, w tym lokalizację fizyczną i informację, czy jest to domyślny katalog.

Wyrażenie wejściowe ▼

```
SELECT * FROM sys.fulltext_indexes;
```

Analiza ▼

To wyrażenie zwraca informacje o zdefiniowanych indeksach, w tym ID katalogu. Informuje też, czy indeks jest aktualizowany automatycznie i kiedy rozpoczęła się i zakończyła ostatnia aktualizacja.

Wskazówka

Funkcja FULLTEXTCATALOGPROPERTY()

Funkcja systemowa FULLTEXTCATALOGPROPERTY() może być wykorzystana do pobrania informacji o katalogach. Funkcja przyjmuje nazwę katalogu i cechę, która ma być sprawdzona. Dwie najważniejsze to IndexSize i PopulateStatus (która informuje nas o tym, czy indeks jest aktualny, czy jest obecnie tworzony itp.).

Wyszukiwanie pełnotekstowe

Gdy dane zostały zaindeksowane, możemy przejść do wykonywania wyszukiwania za pomocą dwóch predykatów:

- ▶ FREETEXT wykonuje proste wyszukiwanie, dopasowując pod względem znaczenia, a nie dokładnego porównania.
- ▶ CONTAINS szuka słów lub wyrażeń, uwzględniając sąsiedztwo, wyrazy pochodne i synonimy.

Zarówno FREETEXT, jak i CONTAINS wykorzystywane są we frazach WHERE zapytań SELECT.

Wyszukiwanie za pomocą FREETEXT

Słowo kluczowe FREETEXT wykorzystywane jest do wyszukiwania wyrazów lub zwrotów, które mogą znaczyć to samo co określone słowo lub są podobne.

Przeanalizuj poniższy przykład. Jest to zwyczajne zapytanie korzystające z LIKE i znaków wieloznacznych.

Wyrażenie wejściowe ▼

```
SELECT id_uw, tekst_uw  
FROM prod_uwagi  
WHERE tekst_uw LIKE 'jedzenie dla królików';
```

Analiza ▼

Zapytanie szuka zwrotu *jedzenie dla królików* w kolumnie tekst_uw. Nie zwrócono żadnych wyników, ponieważ nie ma takiego zwrotu w żadnej tabeli.

Poszukajmy tego samego zwrotu za pomocą wyszukiwania pełnotekstowego FREETEXT:

Wyrażenie wejściowe ▼

```
SELECT id_uw, tekst_uw
FROM prod_uwagi
WHERE FREETEXT(tekst_uw, '%jedzenie dla królików%');
```

Wynik ▼

id_uw	tekst_uw
104	Różna ilość, sprzedawane na worki. Gwarancja jasnego, pomarańczowego koloru, odpowiednie jako przynęta na królika.
110	Reklamacja klienta: królik zdołał wykryć pułapkę, jedzenie jest teraz mniej skuteczne.

Analiza ▼

FREETEXT(tekst_uw, '%jedzenie dla królików%') oznacza „wykonaj wyszukiwanie FREETEXT w kolumnie tekst_uw, wyszukując wszystko, co może oznaczać *jedzenie dla królików*”. Zapytanie zwróciło dwa wiersze. W jednym z nich występuje zarówno słowo *królik*, jak i *jedzenie*, w drugim występuje jedynie *królik*, a z kontekstu wynika, że chodzi o jedzenie, chociaż słowo *jedzenie* nie pada bezpośrednio.

Wyszukiwanie za pomocą FREETEXT jest bardzo łatwe w użyciu. Niestety, prostota ma swój koszt. Ten rodzaj wyszukiwania nie ma bardziej złożonej kontroli, której można by oczekiwać od wyszukiwania pełnotekstowego. Dlatego mamy też do dyspozycji predykat CONTAINS.

Uwaga

Wsparcie dla innych języków

Jeśli nie określono języka, FREETEXT wykorzystuje domyślny język katalogu, by określić, które słowa indeksować, a których nie (pomijane są np. spójniki, których częstość występowania zakłóciłaby wyniki wyszukiwania). Aby określić język, wystarczy podać parametr podczas tworzenia indeksu, tak jak opisano to w sekcji „Tworzenie indeksu pełnotekstowego”. Język musi znajdować się w tabeli systemowej sys.syslanguage. Możesz sprawdzić jej zawartość za pomocą zapytania:

```
SELECT * FROM sys.syslanguage;
```

Wyszukiwanie za pomocą CONTAINS

CONTAINS jest stosowane do wyszukiwania rekordów zawierających określone wyrazy, zwroty, części zwrotów, wyrazy o tym samym rdzeniu, do wyszukiwania sąsiedztwa, synonimów i innych.

Zacznijmy od prostego przykładu:

Wyrażenie wejściowe ▼

```
SELECT id_uw, tekst_uw
FROM prod_uwagi
WHERE CONTAINS(tekst_uw, 'piły');
```

Wynik ▼

id_uw	tekst_uw
112	Reklamacja klienta: Można łatwo wyciąć otwór w podłodze przy pomocy piły ręcznej.

Analiza ▼

WHERE CONTAINS(tekst_uw, 'piły') oznacza „znajdź wyraz *piły* w kolumnie tekst_uw”.

Wskazówka

CONTAINS czy LIKE?

Fraza WHERE CONTAINS(tekst_uw, 'piły') jest funkcjonalnie tożsama z LIKE tekst_uw = '%piły%'. Jednak wyszukiwanie CONTAINS będzie szybsze, zwłaszcza w bazach z większą liczbą rekordów.

CONTAINS obsługuje również znaki wieloznaczne:

Wyrażenie wejściowe ▼

```
SELECT id_uw, tekst_uw
FROM prod_uwagi
WHERE CONTAINS(tekst_uw, '"kowadł*");
```

Wynik ▼

id_uw	tekst_uw
108	Wielokrotne zwroty, kowadła nie spadają dostatecznie szybko lub spadają na kupującego. Zalecamy klientowi wzięcie pod uwagę cięższych kowadeł.

Analiza ▼

"kowadł*" oznacza „dopasuj wyrazy zaczynające się od *kowadł*". Zauważ, że w przeciwieństwie do LIKE, CONTAINS wykorzystuje * jako znak wieloznaczny (zamiast %). Znaki wieloznaczne można stosować na początku i na końcu ciągu tekstowego.

Ostrzeżenie

Uwaga na cudzysłowy

Poszukiwanym słowem w powyższym przykładzie było '"kowadł*"'. W apostrofach zostało umieszczone "kowadł*" (w cudzysłowie). Zwykły tekst w wyszukiwaniu CONTAINS umieszczany jest w apostrofach. Jednak jeśli stosowane są znaki wieloznaczne, należy umieścić tekst ze znakami w cudzysłowie. W przeciwnym razie wyszukiwanie nie zwróci wyników.

CONTAINS obsługuje również operatory logiczne, takie jak AND, OR i NOT. Poniżej kilka przykładów:

Wyrażenie wejściowe ▼

```
SELECT id_uw, tekst_uw
FROM prod_uwagi
WHERE CONTAINS(tekst_uw, 'łatwo AND piły');
```

Wynik ▼

id_uw	tekst_uw
112	Reklamacja klienta: Można łatwo wyciąć otwór w podłodze przy pomocy piły ręcznej.

Analiza ▼

Powyższe zapytanie pobierze tylko te rekordy, w których jednocześnie znajdują się wyrazy *łatwo* i *piły*.

Wyrażenie wejściowe ▼

```
SELECT id_uw, tekst_uw
FROM prod_uwagi
WHERE CONTAINS(tekst_uw, 'królika AND NOT jedzenie');
```

Wynik ▼

id_uw	tekst_uw
104	Różna ilość, sprzedawane na worki. Gwarancja jasnego, pomarańczowego koloru, odpowiednie jako przynęta na królika.

Analiza ▼

Zapytanie wyszukuje wiersze, w których znajduje się słowo *królika* i nie znajduje się *jedzenie*.

Podczas przeszukiwania długich tekstów istnieje prawdopodobieństwo, że wyrazy, których szukasz, znajdują się blisko siebie. Warunek AND szuka dopasowań w całym tekście. Aby kazać silnikowi wyszukiwania pełnotekstowego znaleźć tylko te rekordy, w których poszukiwane słowa znajdują się blisko siebie, należy wykorzystać słowo NEAR, jak w przykładzie poniżej:

Wyrażenie wejściowe ▼

```
SELECT id_uw, tekst_uw
FROM prod_uwagi
WHERE CONTAINS(tekst_uw, 'detonują NEAR szybko');
```

Wynik ▼

id_uw	tekst_uw
105	Lonty są krótkie i klienci skarżą się, że detonują zbyt szybko. Dostępne są dłuższe lonty (produkt FU1), które powinny być zalecane.

Analiza ▼

Powyższe zapytanie pobierze te wiersze, w których wyraz *detonują* znajduje się w pobliżu *szybko*.

Czasami chcemy znaleźć wyraz należący do tej samej grupy (na podstawie rdzenia), jeżeli na przykład szukasz wyrazu *różne* i chciałbyś, aby zapytanie znalazło też wyraz *różna*. Wyszukiwanie słowa *różn** ze znakiem wieloznacznym może dać wiele niepożądanych dopasowań. Możemy rozwiązać ten problem, stosując wyszukiwanie fleksyjne:

Wyrażenie wejściowe ▼

```
SELECT id_uw, tekst_uw
FROM prod_uwagi
WHERE CONTAINS(tekst_uw, 'FORMSOF(INFLECTIONAL, różne)');
```

Wynik ▼

id_uw	tekst_uw
104	Różna ilość, sprzedawane na worki. Gwarancja jasnego, pomarańczowego koloru, odpowiednie jako przynęta na królika.

Analiza ▼

W powyższym przykładzie zastosowano 'FORMSOF(INFLECTIONAL, różne)' do wyszukiwania wszystkich wyrazów, które mają taki sam rdzeń jak *różne*. Dlatego wiersz zawierający wyraz *różna* spełnił kryteria wyszukiwania.

Uwaga

Wyszukiwanie tezaurusowe

Funkcja FORMSOF() wspiera również wyszukiwania tezaurusowe (należy podać argument THESAURUS), w którym można wyszukiwać synonimy. Aby korzystać z tej funkcjonalności, należy najpierw utworzyć i wypełnić plik XML tezauryusa.

Wskazówka

Łączenie wyszukiwanych typów

W celu zachowania czytelności pokazano osobne przykłady wykorzystujące operatory logiczne, znaki wieloznaczne, wyszukiwanie sąsiedztwa i fleksyjne. Możesz jednak łączyć te funkcjonalności w zależności od potrzeb.

Pozycjonowanie wyników

Silnik wyszukiwania pełnotekstowego wykorzystuje zaawansowane algorytmy, aby zlokalizować poszukiwaną wartość. Może również przypisać rangę każdemu z wyników — im bliżej do dopasowania, tym wyższa ranga.

Pozycjonowanie odbywa się za pomocą funkcji. Wyszukiwanie FREETEXT korzysta z funkcji FREETEXTTABLE(), a CONTAINS z funkcji CONTAINSTABLE(). Obie funkcje wykorzystuje się w ten sam sposób i obie przyjmują te same wzorce wyszukiwania (które już wyjaśniono w tej lekcji).

Spójrzmy na poniższy przykład:

Wyrażenie wejściowe ▼

```
SELECT f.rank AS ranga, id_uw, tekst_uw
FROM prod_uwagi,
     FREETEXTTABLE(prod_uwagi, tekst_uw, 'jedzenie dla królików') f
WHERE prod_uwagi.id_uw=f.[key]
ORDER BY ranga DESC;
```

Wynik ▼

ranga	id_uw	tekst_uw
267	110	Reklamacja klienta: królik zdołał wykryć pułapkę, jedzenie jest teraz mniej skuteczne.

42 104 Różna ilość, sprzedawane na worki.
Gwarancja jasnego, pomarańczowego koloru,
odpowiednie jako przynęta na królika.

Analiza ▼

W przykładzie zastosowano wyszukiwanie FREETEXT, lecz zamiast filtrować wyniki za pomocą frazy WHERE, wykorzystuje funkcję FREETEXTTABLE() ze wzorcem wyszukiwania, aby SZBD odnalazł wszystkie rekordy zawierające wyrazy *jedzenie* i *królik*. Funkcja zwraca tabelę z aliasem f (aby można było odwoływać się do jej kolumn w zapytaniu). Tabela ta zawiera kolumnę key, która odpowiada kluczowi głównemu indeksowanej tabeli (w tym przypadku prod_uwagi), i kolumnę rank (w naszym przykładzie z aliasem ranga), w której zapisano wartość rangi. Pierwszy wiersz ma rangę 267, ponieważ bardziej spełnia kryteria wyszukiwania (zawiera dwa z poszukiwanych słów), drugi ma rangę 42, ponieważ mniej pasuje do kryteriów.

Tę funkcjonalność można wykorzystywać również w wyszukiwaniu CONTAINS w ten sam sposób. Należy wtedy pamiętać o zastosowaniu funkcji CONTAINSTABLE().

Uwaga

Przypisywanie wag do warunków wyszukiwania

Pozycjonowanie z powyższego przykładu zakłada, że wszystkie słowa są jednakowo ważne. Jeżeli jednak niektóre słowa są ważniejsze niż inne, należy zastosować funkcję ISABOUT() do przypisania wag poszczególnym słowom. SZBD, przeprowadzając wyszukiwanie pełnotekstowe, uwzględni wagi podczas wyznaczania rangi.

Podsumowanie

W tej lekcji dowiedziałeś się, kiedy stosuje się wyszukiwanie pełnotekstowe i jak wykorzystać funkcje FREETEXT() i CONTAINS(). Nauczyłeś się też wykorzystywać operatory logiczne, znaki wieloznaczne, wyszukiwanie sąsiedztwa i fleksyjne oraz pozycjonowanie wyników w wyszukiwaniu pełnotekstowym.

Skorowidz

A

agregacja, 99
aktualizacja
 danych, 171, 175
 perspektyw, 194
 tabel, 183
aliasy, 76, 198
aliasy tabel, 133
apostrofy, 55

B

baza danych, 15
bezpieczeństwo, 257
białe znaki, 39
blokowanie użytkownika, 260

D

debugger, 199
deklaracja zmiennych, 196
dołączanie, 146
dopasowanie, 54, 57
dostęp do serwera, 25
działanie kursorów, 217

E

eliminowanie zduplikowanych wierszy, 146

F

filtrowanie
 danych, 51
 grup, 105
 niechcianych danych, 192
 zaawansowane, 59
 zapytania zagnieżdżone, 111

formatowanie

 instrukcji, 178
 zapytań, 113
 zwracanych danych, 190

funkcja

 AVG(), 94
 CONTAINSTABLE(), 158, 159
 COUNT, 139
 COUNT(), 95
 FN_HELPCOLLATIONS(), 250
 FREETEXTTABLE(), 158
 FULLTEXTCATALOGPROPERTY(), 153
 ISABOUT(), 159
 MAX(), 96
 MIN(), 97
 RTRIM(), 76
 SUM(), 98

funkcje, 81

 agregujące, 93, 139
 daty i czasu, 84
 JSON, 247
 numeryczne, 90
 tekstowe, 82

G

globalizacja, 249
grupowanie, 107
 danych, 103
 wyrażeń, 203

I

iloczyn kartezjański, 126
indeksy pełnotekstowe, 151
informacje o bazie, 33
instalacja SQL Servera, 26

instrukcja

- ALTER TABLE, 184
- AS DEFAULT, 152
- CREATE LOGIN, 259
- DISABLE TRIGGER, 227
- DROP TRIGGER, 226
- ELSE, 202
- FROM, 46
- GROUP BY, 103, 107
- HAVING, 105, 106
- IDENTITY, 162
- IF, 202, 203
- OPEN, 219
- ORDER BY, 46, 50, 107
- OUTPUT, 168
- SELECT, 37, 109
- TOP, 43
- UNION, 147
- UPDATE, 171
- WHERE, 51, 106, 125

integralność referencyjna, 123

inteligentne wyniki, 149

J

jawna kontrola, 149

język SQL, 20

JSON, 244

K

katalogi pełnotekstowe, 151

kierunek sortowania, 48

klient, 24

klucz

- główny, 19, 179

- obcy, 122

kodowanie, 249

kolejność

- fraz instrukcji SELECT, 109

- wykonywania działań, 61

kolumny, 17

komentarze, 201

konkatenacja pól, 74

kontrola dostępu, 257

kończenie wyrażień, 38

kryteria wyszukiwania, 51

kursory, 217

- działanie, 217

- otwieranie, 218

tworzenie, 218

usuwanie, 218

wykorzystanie danych, 220

zamykanie, 218

L

lokalizacja, 249

Ł

łączenie

- fraz WHERE, 59

- funkcji agregujących, 100

- tabel, 121

- wartości, 74

- wielu tabel, 130

- zapytań, 143

M

modyfikacja danych, 81

N

narzędzia

- administratora, 258

- SQL Server, 25

- SZBD, 123

nawiasy, 62

- kwadratowe, 71, 246

nawiązywanie połączenia, 29

nazwy kwalifikowane, 44

O

obliczenia matematyczne, 77

obsługa

- JSON, 244

- Unicode, 253

- wielkości znaków, 251

- XML, 237

ograniczanie wyników, 42

opcja EXPLICIT, 240

operator

- AND, 59

- BETWEEN, 55

- IN, 63

- LIKE, 67

operator

NOT, 64

OR, 60

operatory frazy WHERE, 52

P

parametry, 211

perspektywy, 187

aktualizowanie, 194

filtrowanie niechcianych danych, 192

formatowanie danych, 190

upraszczanie złożonych złączeń, 189

z polami obliczanymi, 193

zasady tworzenia, 188

pętla WHILE, 205

pobieranie

danych, 37

danych jako JSON, 245

danych jako XML, 237

nieznanych kolumn, 41

pojedynczych kolumn, 37

unikatowych wierszy, 41

wielu kolumn, 39

wszystkich kolumn, 40

pola obliczane, 73, 115

polecenie

ALTER TABLE, 273

BEGIN TRANSACTION, 273

CLOSE, 219

COMMIT, 234

COMMIT TRANSACTION, 274

CREATE INDEX, 274

CREATE LOGIN, 274

CREATE PROCEDURE, 210, 274

CREATE TABLE, 177, 274

CREATE TRIGGER, 226

CREATE VIEW, 275

DEALLOCATE, 218

DELETE, 173, 275

DROP, 210, 275

FETCH, 220

INSERT, 275

INSERT SELECT, 276

ROLLBACK, 233

ROLLBACK TRANSACTION, 276

SAVE TRANSACTION, 276

SELECT, 37, 276

sp_columns, 35

sp_databases, 34

sp_helplogins, 36

sp_helpuser, 36

sp_server_info, 35

sp_spaceused, 35

sp_statistics, 35

sp_tables, 34

UNION, 144

UPDATE, 277

pozycjonowanie wyników, 158

predykat, 68

CONTAINS, 153, 155

FREETEXT, 153

IDENTITY, 180

procedura składowana sp_rename, 185

procedury składowane, 33, 207

inteligentne, 213

parametry, 211

tworzenie, 209

uruchamianie, 209

usuwanie, 210

program typu klient-serwer, 23

przechowywanie danych XML, 241

przetwarzanie

warunkowe, 201

wsadowe, 199

punkt kontrolny, 235

R

raportowanie

usuwanych danych, 174

wprowadzanych danych, 168

relacyjne bazy danych, 121

rodzaje złączeń zewnętrznych, 138

S

sekwencje sortowania znaków, 249

serwer bazy danych, 23

skalowalność, 122

składnia

AS DEFAULT, 152

INNER JOIN, 129

OUTER JOIN, 137

zapytania T-SQL, 273

słowo kluczowe, 32, 283–285

ASC, 50

COLLATE, 253

- DESC, 48, 49
- DISTINCT, 42
- ELEMENTS, 240
- EXISTS, 118
- EXPLICIT, 240
- FREETEXT, 153
- FROM, 37
- IN, 118
- INTO, 167
- LEFT, 137
- LIKE, 155
- PERCENT, 43
- RIGHT, 137
- SELECT, 37
- UNION, 144
- sortowanie, 45, 107, 147
 - niezwracanych kolumn, 47
 - słownikowe, 49
 - względem wielu kolumn, 47
 - znaków, 249
- sprawdzanie
 - braku wartości, 56
 - istnienia rekordów, 117
 - zakresu wartości, 55
- SQL, Structured Query Language, 20
- SQL Server, 23
- SQL Server Management Studio, 30
- symbol gwiazdki, 40

T

- tabele, 16, 267–271
 - aktualizacja, 183
 - nazwy kwalifikowane, 44
 - tworzenie, 31, 177
 - usuwanie, 185
 - zmiana nazwy, 185
- testowanie obliczeń, 79
- tożsamość, 36
- transakcje, 231
 - działanie, 231
 - sterowanie, 233
- T-SQL, 195
 - składnia zapytania, 273
 - słowa kluczowe, 283
 - typy danych, 279
- tworzenie
 - grup, 103
 - indeksu pełnotekstowego, 151

- katalogu pełnotekstowego, 151
- kont użytkowników, 259
- kursorów, 218
- perspektyw, 188
- pól obliczanych, 73
- procedur składowanych, 209
- rozbudowanych złączeń, 133
- tabel, 31, 177
- unii, 143
- wyzwalaczy, 225
- złączeń, 124
- złączeń własnych, 134
- typy danych, 17, 279
 - binarne, 282
 - daty i czasu, 281
 - numeryczne, 281
 - tekstowe, 279
- typy złączeń, 134

U

- Unicode, 253
- unie, 143
- upraszczanie złożonych złączeń, 189
- ustawianie dostępu, 261
- usuwanie
 - danych, 173, 175
 - dostępów, 261
 - kont, 259
 - kursorów, 218
 - procedur składowanych, 210
 - tabel, 185
 - wyzwalaczy, 226
- użytkownik, 258
 - blokowanie, 260
 - konto, 259
 - odblokowywanie, 260
 - usuwanie konta, 259
 - zmiana nazwy, 260

W

- wartości domyślne, 182
- wartość NULL, 35, 56, 69, 179
- wersje SQL Servera, 24
- wielkość liter, 39, 49, 252
- wiersze, 18

- wstawianie
 - pobranych danych, 161, 166
 - wielu wierszy, 161, 165
- wybór bazy danych, 32
- wydajność, 149, 263
- wyszukiwanie
 - danych XML, 243
 - pełnotekstowe, 149, 150, 153
- wyzwalacz
 - DELETE, 229
 - UPDATE, 230
- wyzwalacze, 225
 - dla polecenia INSERT, 228
 - przypisane, 227
 - tworzenie, 225
 - usuwanie, 226
 - włączanie, 227
 - wyłączanie, 227

X

- XML, 237
- XPath, 244
- XQuery, 243

Z

- zagnieżdżanie
 - zapytań, 111
 - skorelowane, 116
- zakres wartości, 55
- zapytania zagnieżdżone, 111, 117
 - jako pola obliczane, 115

- zarządzanie
 - dostępami, 260
 - indeksami, 152
 - katalogami, 152
 - transakcjami, 231
 - użytkownikami, 258
- zasady stosowania unii, 146
- zestaw znaków, 249
- złączenia, 121, 140
 - krzyżowe, 128
 - naturalne, 136
 - wewnętrzne, 129
 - własne, 134
 - zewnętrzne, 136, 138
- zmiana
 - hasła, 260
 - nazwy tabeli, 185
 - nazwy użytkownika, 260
 - zatwierdzania automatycznego, 236
- zmienne, 195
 - przypisywanie wartości, 196
 - sprawdzanie zawartości, 197
 - zastosowanie, 199
- znak
 - podkreślenia, 70
 - procentu, 68
 - spacji, 69
 - wieloznaczny, 41, 67, 72
- zwiększanie wydajności, 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**

SQL Server jest potężnym i godnym zaufania narzędziem, które sprawdzi się wszędzie tam, gdzie potrzebny jest szybki i niezawodny system do przechowywania i udostępniania danych. Narzędzie to jednak trzeba poznać, aby w pełni wykorzystać jego możliwości. Niezbędna jest również umiejętność posługiwania się językiem T-SQL. Tymczasem nie każdy dysponuje wystarczającą ilością czasu, aby przyswoić sobie wszystkie istotne zagadnienia za pomocą standardowych podręczników. Zdarzają się sytuacje, kiedy trzeba wdrożyć się w pracę z systemem SQL Server w ciągu paru dni!

Trzymasz w rękach zwięzły, bardzo dobrze napisany podręcznik, dzięki któremu błyskawicznie zaczniesz posługiwać się systemem SQL Server i językiem T-SQL. Materiał podzielono na 30 lekcji. Przyswojenie materiału każdej z nich powinno zająć nie więcej niż 10 minut. Po opanowaniu prostego pobierania danych zapoznasz się z bardziej złożonymi zagadnieniami, takimi jak podzapytania, funkcje i procedury składowane, kursory, wyzwalacze, więzy tabel, XML, JSON i wiele innych. Niezbędne informacje uporządkowano w sposób metodyczny, systematyczny i precyzyjny, a liczne wskazówki, uzupełnienia i ostrzeżenia stanowią świetne uzupełnienie każdej lekcji.

W książce przedstawiono między innymi:

- Stosowanie T-SQL w środowisku Microsoft SQL Server
- Tworzenie złożonych zapytań T-SQL
- Stosowanie filtrów, pobieranie, sortowanie i formatowanie danych
- Zastosowanie globalizacji i lokalizacji w Microsoft SQL Server
- Złączenia i zapytania zagnieżdżone
- Tworzenie i modyfikowanie tabel

SQL Server i T-SQL? Naucz się tego na wczoraj!

Ben Forta od 30 lat projektuje i rozwija oprogramowanie, a także udziela wsparcia technicznego i prowadzi szkolenia. Obecnie pracuje w firmie Adobe. Jest autorem bardzo popularnych książek poświęconych bazom danych, SQL, Flashowi, Javie, systemom Windows i wielu innym zagadnieniom. Ma szerokie doświadczenie w projektowaniu i implementacji baz danych.

Helion

księgarnia internetowa



<http://helion.pl>

zamówienia telefoniczne



0 801 339900



0 601 339900

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

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

sięgnij po **WIĘCEJ**



KOD KORZYŚCI

ISBN 978-83-283-3565-3



9 788328 335653

Informatyka w najlepszym wydaniu

cena: 49,00 zł