**Nama : AMRIDZAR**

**NIM : 200250502011**

**Mata Kuliah : Sistem Basis Data**

**SQL Tutorial**

**SQL Home**

SQL adalah bahasa standar untuk menyimpan, memanipulasi dan mengambil data dalam database.

Tutorial SQL kami akan mengajarkan Anda cara menggunakan SQL di: MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres, dan sistem database lainnya.

Contoh di Setiap Bab

Dengan editor SQL online kami, Anda dapat mengedit pernyataan SQL.

Contoh

SELECT \* FROM Customers;

**SQL Intro**

SQL adalah bahasa standar untuk mengakses dan memanipulasi database.

Apa itu SQL?

SQL adalah singkatan dari Structured Query Language

SQL memungkinkan Anda mengakses dan memanipulasi database

SQL menjadi standar American National Standards Institute (ANSI) pada 1986, dan International Organization for Standardization (ISO) pada 1987

Apa yang Dapat SQL lakukan?

SQL dapat mengeksekusi query terhadap database

SQL dapat mengambil data dari database

SQL dapat menyisipkan catatan dalam database

SQL dapat memperbarui catatan dalam database

SQL dapat menghapus catatan dari database

SQL dapat membuat database baru

SQL dapat membuat tabel baru dalam database

SQL dapat membuat prosedur tersimpan dalam database

SQL dapat membuat tampilan dalam database

SQL dapat mengatur izin pada tabel, prosedur, dan tampilan

SQL adalah Standar - TAPI ....

Meskipun SQL adalah standar ANSI/ISO, ada beberapa versi bahasa SQL yang berbeda.

Namun, agar sesuai dengan standar ANSI, semuanya mendukung setidaknya perintah utama (seperti SELECT, UPDATE, DELETE, INSERT, WHERE) dengan cara yang sama.

Catatan: Sebagian besar program database SQL juga memiliki ekstensi milik sendiri selain standar SQL!

Menggunakan SQL di Situs Web Anda

Untuk membangun situs web yang menampilkan data dari database, Anda memerlukan:

Sebuah program database RDBMS (yaitu MS Access, SQL Server, MySQL)

Untuk menggunakan bahasa skrip sisi server, seperti PHP atau ASP

Untuk menggunakan SQL untuk mendapatkan data yang Anda inginkan

Untuk menggunakan HTML / CSS untuk menata halaman

RDBMS

RDBMS adalah singkatan dari Sistem Manajemen Basis Data Relasional.

RDBMS adalah dasar untuk SQL, dan untuk semua sistem database modern seperti MS SQL Server, IBM DB2, Oracle, MySQL, dan Microsoft Access.

Data dalam RDBMS disimpan dalam objek database yang disebut tabel. Tabel adalah kumpulan entri data terkait dan terdiri dari kolom dan baris.

Lihat tabel "Pelanggan":

Contoh

SELECT \* FROM Customers;

Setiap tabel dipecah menjadi entitas yang lebih kecil yang disebut bidang. Bidang dalam tabel Pelanggan terdiri dari ID Pelanggan, NamaPelanggan, NamaKontak, Alamat, Kota, Kode Pos, dan Negara. Bidang adalah kolom dalam tabel yang dirancang untuk mempertahankan informasi spesifik tentang setiap catatan dalam tabel.

Catatan, juga disebut baris, adalah setiap entri individu yang ada dalam tabel. Misalnya, ada 91 record dalam tabel Customers di atas. Record adalah entitas horizontal dalam sebuah tabel.

Kolom adalah entitas vertikal dalam tabel yang berisi semua informasi yang terkait dengan bidang tertentu dalam tabel.

**SQL Syntax**

Tabel Database

Database paling sering berisi satu atau lebih tabel. Setiap tabel diidentifikasi dengan nama (misalnya "Pelanggan" atau "Pesanan"). Tabel berisi catatan (baris) dengan data.

Dalam tutorial ini kita akan menggunakan database sampel Northwind yang terkenal (termasuk dalam MS Access dan MS SQL Server).

Pernyataan SQL

Sebagian besar tindakan yang perlu Anda lakukan pada database dilakukan dengan pernyataan SQL.

Pernyataan SQL berikut memilih semua catatan dalam tabel "Pelanggan":

Contoh

SELECT \* FROM Customers;

Ingatlah Bahwa...

Kata kunci SQL TIDAK peka huruf besar-kecil: selectsama dengan SELECT

Dalam tutorial ini kita akan menulis semua kata kunci SQL dalam huruf besar.

Titik koma setelah Pernyataan SQL?

Beberapa sistem database memerlukan titik koma di akhir setiap pernyataan SQL.

Titik koma adalah cara standar untuk memisahkan setiap pernyataan SQL dalam sistem database yang memungkinkan lebih dari satu pernyataan SQL dieksekusi dalam panggilan yang sama ke server.

Dalam tutorial ini, kita akan menggunakan titik koma di akhir setiap pernyataan SQL.

Beberapa Perintah SQL Paling Penting

SELECT - mengekstrak data dari database

UPDATE - memperbarui data dalam database

DELETE - menghapus data dari database

INSERT INTO - memasukkan data baru ke dalam database

CREATE DATABASE - membuat database baru

ALTER DATABASE - memodifikasi database

CREATE TABLE - membuat tabel baru

ALTER TABLE - memodifikasi tabel

DROP TABLE - menghapus tabel

CREATE INDEX - membuat indeks (kunci pencarian)

DROP INDEX - menghapus indeks

**SQL Select**

Pernyataan SQL SELECT

The SELECTpernyataan digunakan untuk memilih data dari database.

Data yang dikembalikan disimpan dalam tabel hasil, yang disebut kumpulan hasil.

PILIH Sintaks

SELECT column1, column2, ...

FROM table\_name;

Di sini, kolom1, kolom2, ... adalah nama bidang dari tabel yang ingin Anda pilih datanya. Jika Anda ingin memilih semua bidang yang tersedia dalam tabel, gunakan sintaks berikut:

SELECT \* FROM table\_name;

Basis Data Demo

Di bawah ini adalah pilihan dari tabel "Pelanggan" di database sampel Northwind:

CustomerID CustomerName ContactName Address City PostalCode Country

1 Alfreds Futterkiste Maria Anders Obere Str. 57 Berlin 12209 Germany

2 Ana Trujillo Emparedados y helados Ana Trujillo Avda. de la Constitución 2222 México D.F. 05021 Mexico

3 Antonio Moreno Taquería Antonio Moreno Mataderos 2312 México D.F. 05023 Mexico

4 Around the Horn Thomas Hardy 120 Hanover Sq. London WA1 1DP UK

5 Berglunds snabbköp Christina Berglund Berguvsvägen 8 Luleå S-958 22 Sweden

PILIH Kolom Contoh

Pernyataan SQL berikut memilih kolom "NamaPelanggan" dan "Kota" dari tabel "Pelanggan":

Contoh

SELECT CustomerName, City FROM Customers;

PILIH \* Contoh

Pernyataan SQL berikut memilih semua kolom dari tabel "Pelanggan":

Contoh

SELECT \* FROM Customers;

**SQL Select Distinct**

Pernyataan SQL SELECT DISTINCT

The SELECT DISTINCTpernyataan digunakan untuk kembali hanya berbeda nilai-nilai (yang berbeda).

Di dalam tabel, kolom sering kali berisi banyak nilai duplikat; dan terkadang Anda hanya ingin membuat daftar nilai yang berbeda (berbeda).

PILIH Sintaks BERBEDA

SELECT DISTINCT column1, column2, ...

FROM table\_name;

Basis Data Demo

Di bawah ini adalah pilihan dari tabel "Pelanggan" di database sampel Northwind:

CustomerID CustomerName ContactName Address City PostalCode Country

1 Alfreds Futterkiste Maria Anders Obere Str. 57 Berlin 12209 Germany

2 Ana Trujillo Emparedados y helados Ana Trujillo Avda. de la Constitución 2222 México D.F. 05021 Mexico

3 Antonio Moreno Taquería Antonio Moreno Mataderos 2312 México D.F. 05023 Mexico

4 Around the Horn Thomas Hardy 120 Hanover Sq. London WA1 1DP UK

5 Berglunds snabbköp Christina Berglund Berguvsvägen 8 Luleå S-958 22Sweden

PILIH Contoh Tanpa DISTINCT

Pernyataan SQL berikut memilih semua (termasuk duplikat) nilai dari kolom "Negara" di tabel "Pelanggan":

Contoh

SELECT Country FROM Customers;

**SQL Where**

Klausa SQL WHERE

The WHEREklausul digunakan untuk catatan filter.

Ini digunakan untuk mengekstrak hanya catatan yang memenuhi kondisi tertentu.

Sintaks DIMANA

SELECT column1, column2, ...

FROM table\_name

WHERE condition;

Catatan: The WHEREklausul tidak hanya digunakan dalam SELECTlaporan, itu juga digunakan dalam UPDATE, DELETE, dll!

Basis Data Demo

Di bawah ini adalah pilihan dari tabel "Pelanggan" di database sampel Northwind:

CustomerID CustomerName ContactName Address City PostalCode Country

1 Alfreds Futterkiste Maria Anders Obere Str. 57 Berlin 12209 Germany

2 Ana Trujillo Emparedados y helados Ana Trujillo Avda. de la Constitución 2222 México D.F. 05021 Mexico

3 Antonio Moreno Taquería Antonio Moreno Mataderos 2312 México D.F. 05023 Mexico

4 Around the Horn Thomas Hardy 120 Hanover Sq. London WA1 1DP UK

5 Berglunds snabbköp Christina Berglund Berguvsvägen 8 Luleå S-958 22 Sweden

Contoh Klausa DIMANA

Pernyataan SQL berikut memilih semua pelanggan dari negara "Meksiko", di tabel "Pelanggan":

Contoh

SELECT \* FROM Customers

WHERE Country='Mexico';

Bidang Teks vs. Bidang Numerik

SQL membutuhkan tanda kutip tunggal di sekitar nilai teks (kebanyakan sistem database juga akan mengizinkan tanda kutip ganda).

Namun, bidang numerik tidak boleh diapit tanda kutip:

Contoh

SELECT \* FROM Customers

WHERE CustomerID=1;

Operator dalam Klausa WHERE

Operator berikut dapat digunakan dalam WHEREklausa:

Operator Description

= Equal

> Greater than

< Less than

>= Greater than or equal

<= Less than or equal

<> Not equal. Note: In some versions of SQL this operator may be written as !=

BETWEEN Between a certain range

LIKE Search for a pattern

IN To specify multiple possible values for a column

**SQL And, Or, Not**

Operator SQL AND, OR dan NOT

The WHEREklausa dapat dikombinasikan dengan AND, OR, dan NOToperator.

The ANDdan ORoperator yang digunakan untuk catatan filter berdasarkan lebih dari satu syarat:

The ANDmenampilkan Operator catatan jika semua kondisi dipisahkan oleh AND BENAR.

The ORmenampilkan Operator catatan jika salah satu kondisi yang dipisahkan oleh ORBENAR.

The NOTmenampilkan Operator catatan jika kondisi (s) adalah TIDAK BENAR.

DAN Sintaks

SELECT column1, column2, ...

FROM table\_name

WHERE condition1 AND condition2 AND condition3 ...;

ATAU Sintaks

SELECT column1, column2, ...

FROM table\_name

WHERE condition1 OR condition2 OR condition3 ...;

BUKAN Sintaks

SELECT column1, column2, ...

FROM table\_name

WHERE NOT condition;

Basis Data Demo

Tabel di bawah ini menunjukkan tabel "Pelanggan" lengkap dari database sampel Northwind:

CustomerID CustomerName ContactName Address City PostalCode Country

1 Alfreds Futterkiste Maria Anders Obere Str. 57 Berlin 12209 Germany

2 Ana Trujillo Emparedados y helados Ana Trujillo Avda. de la Constitución 2222 México D.F. 05021 Mexico

3 Antonio Moreno Taquería Antonio Moreno Mataderos 2312 México D.F. 0502 Mexico

4 Around the Horn Thomas Hardy 120 Hanover Sq. London WA1 1DP UK

5 Berglunds snabbköp Christina Berglund Berguvsvägen 8 Luleå S-958 22 Sweden

6 Blauer See Delikatessen Hanna Moos Forsterstr. 57 Mannheim 68306 Germany

7 Blondel père et fils Frédérique Citeaux 24, place Kléber Strasbourg 67000 France

8 Bólido Comidas preparadas Martín Sommer C/ Araquil, 67 Madrid 28023 Spain

9 Bon app' Laurence Lebihans 12, rue des Bouchers Marseille 13008 France

10 Bottom-Dollar Marketse Elizabeth Lincoln 23 Tsawassen Blvd. Tsawassen T2F 8M4 Canada

11 B's Beverages Victoria Ashworth Fauntleroy Circus London EC2 5NT UK

12 Cactus Comidas para llevar Patricio Simpson Cerrito 333 Buenos Aires1010 Argentina

13 Centro comercial Moctezuma Francisco Chang Sierras de Granada 9993 México D.F. 05022 Mexico

14 Chop-suey Chinese Yang Wang Hauptstr. 29 Bern 3012 Switzerland

15 Comércio Mineiro Pedro Afonso Av. dos Lusíadas, 23 São Paulo 05432-043 Brazil

16 Consolidated Holdings Elizabeth Brown Berkeley Gardens 12 Brewery London WX1 6LT UK

17 Drachenblut Delikatessend Sven Ottlieb Walserweg 21 Aachen 52066 Germany

18 Du monde entier Janine Labrune 67, rue des Cinquante Otages Nantes 44000 France

19 Eastern Connection Ann Devon 35 King George London WX3 6FW UK

20 Ernst Handel Roland Mendel Kirchgasse 6 Graz 8010 Austria

21 Familia Arquibaldo Aria Cruz Rua Orós, 92 São Paulo 05442-030 Brazil

22 FISSA Fabrica Inter. Salchichas S.A. Diego Roel C/ Moralzarzal, 86 Madrid 28034 Spain

23 Folies gourmandes Martine Rancé 184, chaussée de Tournai Lille 59000 France

24 Folk och fä HB Maria Larsson Åkergatan 24 Bräcke S-844 67 Sweden

25 Frankenversand Peter Franken Berliner Platz 43 München 80805 Germany

26 France restauration Carine Schmitt 54, rue Royale Nantes 44000 France

27 Franchi S.p.A. Paolo Accorti Via Monte Bianco 34 Torino 10100 Italy

28 Furia Bacalhau e Frutos do Mar Lino Rodriguez Jardim das rosas n. 32 Lisboa 1675 Portugal

29 Galería del gastrónomo Eduardo Saavedra Rambla de Cataluña, 23 Barcelona 08022 Spain

30 Godos Cocina Típica José Pedro Freyre C/ Romero, 33 Sevilla 41101 Spain

31 Gourmet Lanchonetes André Fonseca Av. Brasil, 442 Campinas 04876-786 Brazil

32 Great Lakes Food Market Howard Snyder 2732 Baker Blvd. Eugene 97403 USA

33 GROSELLA-Restaurante Manuel Pereira 5ª Ave. Los Palos Grandes Caracas 1081 Venezuela

34 Hanari Carnes Mario Pontes Rua do Paço, 67 Rio de Janeiro 05454-876 Brazil

35 HILARIÓN-Abastos Carlos Hernández Carrera 22 con Ave. Carlos Soublette #8-35 San Cristóbal 5022 Venezuela

36 Hungry Coyote Import Store Yoshi Latimer City Center Plaza 516 Main St. Elgin 97827 USA

37 Hungry Owl All-Night Grocers Patricia McKenna 8 Johnstown Road Cork Ireland

38 Island Trading Helen Bennett Garden House Crowther Way Cowes PO31 7PJ UK

39 Königlich Essen Philip Cramer Maubelstr. 90 Brandenburg 14776 Germany

40 La corne d'abondance Daniel Tonini 67, avenue de l'Europe Versailles 78000 France

41 La maison d'Asie Annette Roulet 1 rue Alsace-Lorraine Toulouse 31000 France

42 Laughing Bacchus Wine Cellars Yoshi Tannamuri 1900 Oak St. Vancouver V3F 2K1 Canada

43 Lazy K Kountry Store John Steel 12 Orchestra Terrace Walla Walla 99362 USA

44 Lehmanns Marktstand Renate Messner Magazinweg 7 Frankfurt a.M. 60528 Germany

45 Let's Stop N Shop Jaime Yorres 87 Polk St. Suite 5 San Francisco 94117 USA

46 LILA-Supermercado Carlos González Carrera 52 con Ave. Bolívar #65-98 Llano Largo Barquisimeto 3508 Venezuela

47 LINO-Delicateses Felipe Izquierdo Ave. 5 de Mayo Porlamar I. de Margarita 4980 Venezuela

48 Lonesome Pine Restaurant Fran Wilson 89 Chiaroscuro Rd. Portland 97219 USA

49 Magazzini Alimentari Riuniti Giovanni Rovelli Via Ludovico il Moro 22 Bergamo 24100 Italy

50 Maison Dewey Catherine Dewey Rue Joseph-Bens 532 Bruxelles B-1180 Belgium

51 Mère Paillarde Jean Fresnière 43 rue St. Laurent Montréal H1J 1C3 Canada

52 Morgenstern Gesundkost Alexander Feuer Heerstr. 22 Leipzig 04179 Germany

53 North/South Simon Crowther South House 300 Queensbridge London SW7 1RZ UK

54 Océano Atlántico Ltda. Yvonne Moncada Ing. Gustavo Moncada 8585 Piso 20-A Buenos Aires 1010 Argentina

55 Old World Delicatessen Rene Phillips 2743 Bering St. Anchorage 99508 USA

56 Ottilies Käseladen Henriette Pfalzheim Mehrheimerstr. 369 Köln 50739 Germany

57 Paris spécialités Marie Bertrand 265, boulevard Charonne Paris 75012 France

58 Pericles Comidas clásicas Guillermo Fernández Calle Dr. Jorge Cash 321 México D.F. 05033 Mexico

59 Piccolo und mehr Georg Pipps Geislweg 14 Salzburg 5020 Austria

60 Princesa Isabel Vinhoss Isabel de Castro Estrada da saúde n. 58 Lisboa 1756 Portugal

61 Que Delícia Bernardo Batista Rua da Panificadora, 12 Rio de Janeiro 02389-673 Brazil

62 Queen Cozinha Lúcia Carvalho Alameda dos Canàrios, 891 São Paulo 05487-020 Brazil

63 QUICK-Stop Horst Kloss Taucherstraße 10 Cunewalde 01307 Germany

64 Rancho grande Sergio Gutiérrez Av. del Libertador 900 Buenos Aires 1010 Argentina

65 Rattlesnake Canyon Grocery Paula Wilson 2817 Milton Dr. Albuquerque 87110 USA

66 Reggiani Caseifici Maurizio Moroni Strada Provinciale 124 Reggio Emilia 42100 Italy

67 Ricardo Adocicados Janete Limeira Av. Copacabana, 267 Rio de Janeiro 02389-890 Brazil

68 Richter Supermarkt Michael Holz Grenzacherweg 237 Genève 1203 Switzerland

69 Romero y tomillo Alejandra Camino Gran Vía, 1 Madrid 28001 Spain

70 Santé Gourmet Jonas Bergulfsen Erling Skakkes gate 78 Stavern 4110 Norway

71 Save-a-lot Markets Jose Pavarotti 187 Suffolk Ln. Boise 83720 USA

72 Seven Seas Imports Hari Kumar 90 Wadhurst Rd. London OX15 4NB UK

73 Simons bistro Jytte Petersen Vinbæltet 34 København 1734 Denmark

74 Spécialités du monde Dominique Perrier 25, rue Lauriston Paris 75016 France

75 Split Rail Beer & Ale Art Braunschweiger P.O. Box 555 Lander 82520 USA

76 Suprêmes délices Pascale Cartrain Boulevard Tirou, 255 Charleroi B-6000 Belgium

77 The Big Cheese Liz Nixon 89 Jefferson Way Suite 2 Portland 97201 USA

78 The Cracker Box Liu Wong 55 Grizzly Peak Rd. Butte 59801 USA

79 Toms Spezialitäten Karin Josephs Luisenstr. 48 Münster 44087 Germany

80 Tortuga Restaurante Miguel Angel Paolino Avda. Azteca 123 México D.F. 05033 Mexico

81 Tradição Hipermercados Anabela Domingues Av. Inês de Castro, 414 São Paulo 05634-030 Brazil

82 Trail's Head Gourmet Provisioners Helvetius Nagy 722 DaVinci Blvd. Kirkland 98034 USA

83 Vaffeljernet Palle Ibsen Smagsløget 45 Århus 8200 Denmark

84 Victuailles en stock Mary Saveley 2, rue du Commerce Lyon 69004 France

85 Vins et alcools Chevalier Paul Henriot 59 rue de l'Abbaye Reims 51100 France

86 Die Wandernde Kuh Rita Müller Adenauerallee 900 Stuttgart 70563 Germany

87 Wartian Herkku Pirkko Koskitalo Torikatu 38 Oulu 90110 Finland

88 Wellington Importadora Paula Parente Rua do Mercado, 12 Resende 08737-363 Brazil

89 White Clover Markets Karl Jablonski 305 - 14th Ave. S. Suite 3B Seattle 98128 \ USA

90 Wilman Kala Matti Karttunen Keskuskatu 45 Helsinki 21240 Finland

91 Wolski Zbyszek ul. Filtrowa 68 Walla 01-012 Poland

DAN Contoh

Pernyataan SQL berikut memilih semua bidang dari "Pelanggan" di mana negara adalah "Jerman" DAN kota adalah "Berlin":

Contoh

SELECT \* FROM Customers

WHERE Country='Germany' AND City='Berlin';

ATAU Contoh

Pernyataan SQL berikut memilih semua bidang dari "Pelanggan" dengan kota "Berlin" ATAU "München":

Contoh

SELECT \* FROM Customers

WHERE City='Berlin' OR City='München';

Pernyataan SQL berikut memilih semua bidang dari "Pelanggan" dengan negara "Jerman" ATAU "Spanyol":

Contoh

SELECT \* FROM Customers

WHERE Country='Germany' OR Country='Spain';

BUKAN Contoh

Pernyataan SQL berikut memilih semua bidang dari "Pelanggan" di mana negara BUKAN "Jerman":

Contoh

SELECT \* FROM Customers

WHERE NOT Country='Germany';

Menggabungkan AND, OR dan NOT

Anda juga dapat menggabungkan operator AND, ORdan NOT.

Pernyataan SQL berikut memilih semua bidang dari "Pelanggan" di mana negara adalah "Jerman" DAN kota harus "Berlin" ATAU "München" (gunakan tanda kurung untuk membentuk ekspresi kompleks):

Contoh

SELECT \* FROM Customers

WHERE Country='Germany' AND (City='Berlin' OR City='München');

Pernyataan SQL berikut memilih semua bidang dari "Pelanggan" di mana negara BUKAN "Jerman" dan BUKAN "AS":

Contoh

SELECT \* FROM Customers

WHERE NOT Country='Germany' AND NOT Country='USA';

**SQL Order by**

SQL ORDER BY Kata Kunci

Kata ORDER BYkunci digunakan untuk mengurutkan hasil-set dalam urutan menaik atau menurun.

Kata ORDER BYkunci mengurutkan catatan dalam urutan menaik secara default. Untuk mengurutkan catatan dalam urutan menurun, gunakan DESCkata kunci.

ORDER DENGAN Sintaks

SELECT column1, column2, ...

FROM table\_name

ORDER BY column1, column2, ... ASC|DESC;

Basis Data Demo

Di bawah ini adalah pilihan dari tabel "Pelanggan" di database sampel Northwind:

CustomerID CustomerName ContactName Address City PostalCode Country

1 Alfreds Futterkiste Maria Anders Obere Str. 57 Berlin 12209 Germany

2 Ana Trujillo Emparedados y helados Ana Trujillo Avda. de la Constitución 2222 México D.F. 05021 Mexico

3 Antonio Moreno Taquería Antonio Moreno Mataderos 2312 México D.F. 05023 Mexico

4 Around the Horn Thomas Hardy 120 Hanover Sq. London WA1 1DP UK

5 Berglunds snabbköp Christina Berglund Berguvsvägen 8 Luleå S-958 22 Sweden

ORDER DENGAN Contoh

Pernyataan SQL berikut memilih semua pelanggan dari tabel "Pelanggan", diurutkan menurut kolom "Negara":

Contoh

SELECT \* FROM Customers

ORDER BY Country;

**SQL Insert Into**Pernyataan SQL INSERT INTO

The INSERT INTOpernyataan digunakan untuk menyisipkan catatan baru dalam sebuah tabel.

MASUKKAN KE Sintaks

Dimungkinkan untuk menulis INSERT INTO pernyataan dalam dua cara:

1. Tentukan nama kolom dan nilai yang akan disisipkan:

INSERT INTO table\_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

2. Jika Anda menambahkan nilai untuk semua kolom tabel, Anda tidak perlu menentukan nama kolom dalam kueri SQL. Namun, pastikan urutan nilai dalam urutan yang sama dengan kolom dalam tabel. Di sini, INSERT INTOsintaksnya adalah sebagai berikut:

INSERT INTO table\_name

VALUES (value1, value2, value3, ...);

Basis Data Demo

Di bawah ini adalah pilihan dari tabel "Pelanggan" di database sampel Northwind:

CustomerID CustomerName ContactName Address City PostalCode Country

89White Clover Markets Karl Jablonski 305 - 14th Ave. S. Suite 3B Seattle 98128 USA

90 Wilman Kala Matti Karttunen Keskuskatu 45 Helsinki 21240 Finland

91 Wolski Zbyszek ul. Filtrowa 68 Walla 01-012 Poland

**SQL null value**

Bidang dengan nilai NULL adalah bidang tanpa nilai.

Jika bidang dalam tabel adalah opsional, dimungkinkan untuk menyisipkan catatan baru atau memperbarui catatan tanpa menambahkan nilai ke bidang ini. Kemudian, bidang tersebut akan disimpan dengan nilai NULL.

Catatan: Nilai NULL berbeda dari nilai nol atau bidang yang berisi spasi. Bidang dengan nilai NULL adalah bidang yang dibiarkan kosong selama pembuatan rekaman!

Bagaimana Menguji Nilai NULL?

Tidak mungkin menguji nilai NULL dengan operator perbandingan, seperti =, <, atau <>.

Kita harus menggunakan operator IS NULLand IS NOT NULLsebagai gantinya.

Sintaks IS NULL

SELECT column\_names

FROM table\_name

WHERE column\_name IS NULL;

Sintaks BUKAN NULL

SELECT column\_names

FROM table\_name

WHERE column\_name IS NOT NULL;

Operator IS NULL

The IS NULLoperator yang digunakan untuk menguji untuk nilai kosong (nilai NULL).

SQL berikut mencantumkan semua pelanggan dengan nilai NULL di bidang "Alamat":

Contoh

SELECT CustomerName, ContactName, Address

FROM Customers

WHERE Address IS NULL;

Tip: Selalu gunakan IS NULL untuk mencari nilai NULL.

Operator BUKAN NULL

The IS NOT NULLoperator yang digunakan untuk menguji nilai-nilai non-kosong (NOT nilai NULL).

SQL berikut mencantumkan semua pelanggan dengan nilai di bidang "Alamat":

Contoh

SELECT CustomerName, ContactName, Address

FROM Customers

WHERE Address IS NOT NULL.

**SQL Update**

Pernyataan SQL UPDATE

The UPDATEpernyataan digunakan untuk memodifikasi catatan yang ada di meja.

PERBARUI Sintaks

UPDATE table\_name

SET column1 = value1, column2 = value2, ...

WHERE condition;

Catatan: Hati-hati saat memperbarui catatan dalam tabel! Perhatikan WHEREklausa dalam UPDATEpernyataan. The WHEREklausul menentukan yang record (s) yang harus diperbarui. Jika Anda menghilangkan WHEREklausa, semua catatan dalam tabel akan diperbarui!

PERBARUI Tabel

Pernyataan SQL berikut memperbarui pelanggan pertama (ID Pelanggan = 1) dengan kontak baru dan kota baru.

Contoh

UPDATE Customers

SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'

WHERE CustomerID = 1;

UPDATE Beberapa Catatan

Ini adalah WHEREklausa yang menentukan berapa banyak catatan yang akan diperbarui.

Pernyataan SQL berikut akan memperbarui ContactName menjadi "Juan" untuk semua catatan di mana negaranya adalah "Meksiko":

Contoh

UPDATE Customers

SET ContactName='Juan'

WHERE Country='Mexico';

Perbarui Peringatan!

Berhati-hatilah saat memperbarui catatan. Jika Anda menghilangkan WHEREklausa, SEMUA catatan akan diperbarui!

Contoh

UPDATE Customers

SET ContactName='Juan';

**SQL Delete**

Pernyataan SQL DELETE

The DELETEpernyataan digunakan untuk menghapus catatan yang ada di meja.

HAPUS Sintaks

DELETE FROM table\_name WHERE condition;

Catatan: Hati-hati saat menghapus record dalam tabel! Perhatikan WHEREklausa dalam DELETEpernyataan. The WHEREklausul menentukan yang record (s) harus dihapus. Jika Anda menghilangkan WHEREklausa, semua catatan dalam tabel akan dihapus!

Contoh SQL DELETE

Pernyataan SQL berikut menghapus pelanggan "Alfreds Futterkiste" dari tabel "Pelanggan":

Contoh

DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';

Hapus Semua Catatan

Dimungkinkan untuk menghapus semua baris dalam tabel tanpa menghapus tabel. Ini berarti bahwa struktur tabel, atribut, dan indeks akan tetap utuh:

DELETE FROM table\_name;

Pernyataan SQL berikut menghapus semua baris dalam tabel "Pelanggan", tanpa menghapus tabel:

Contoh

DELETE FROM Customers;

**SQL Select TOP**

Klausa TOP SELECT SQL

The SELECT TOPklausul digunakan untuk menentukan jumlah catatan untuk kembali.

The SELECT TOPklausul berguna pada tabel besar dengan ribuan catatan. Mengembalikan sejumlah besar catatan dapat memengaruhi kinerja.

Catatan: Tidak semua sistem database mendukung SELECT TOPklausa. MySQL mendukung LIMITklausa untuk memilih sejumlah catatan, sementara Oracle menggunakan dan . FETCH FIRST n ROWS ONLYROWNUM

Sintaks SQL Server / MS Access:

SELECT TOP number|percent column\_name(s)

FROM table\_name

WHERE condition;

Sintaks MySQL:

SELECT column\_name(s)

FROM table\_name

WHERE condition

LIMIT number;

Sintaks Oracle 12:

SELECT column\_name(s)

FROM table\_name

ORDER BY column\_name(s)

FETCH FIRST number ROWS ONLY;

Sintaks Oracle yang Lebih Lama:

SELECT column\_name(s)

FROM table\_name

WHERE ROWNUM <= number;

Sintaks Oracle Lama (dengan ORDER BY):

SELECT \*

FROM (SELECT column\_name(s) FROM table\_name ORDER BY column\_name(s))

WHERE ROWNUM <= number;

Contoh SQL TOP, LIMIT dan FETCH FIRST

Pernyataan SQL berikut memilih tiga catatan pertama dari tabel "Pelanggan" (untuk SQL Server/MS Access):

Contoh

SELECT TOP 3 \* FROM Customers;

Pernyataan SQL berikut menunjukkan contoh yang setara untuk MySQL:

Contoh

SELECT \* FROM Customers

LIMIT 3;

Pernyataan SQL berikut menunjukkan contoh yang setara untuk Oracle:

Contoh

SELECT \* FROM Customers

FETCH FIRST 3 ROWS ONLY;

Contoh PERSEN TERATAS SQL

Pernyataan SQL berikut memilih 50% pertama dari catatan dari tabel "Pelanggan" (untuk SQL Server/MS Access):

Contoh

SELECT TOP 50 PERCENT \* FROM Customers;

Pernyataan SQL berikut menunjukkan contoh yang setara untuk Oracle:

Contoh

SELECT \* FROM Customers

FETCH FIRST 50 PERCENT ROWS ONLY;

TAMBAHKAN KLAUSE MANA

Pernyataan SQL berikut memilih tiga catatan pertama dari tabel "Pelanggan", di mana negaranya adalah "Jerman" (untuk SQL Server/MS Access):

Contoh

SELECT TOP 3 \* FROM Customers

WHERE Country='Germany';

Pernyataan SQL berikut menunjukkan contoh yang setara untuk MySQL:

Contoh

SELECT \* FROM Customers

WHERE Country='Germany'

LIMIT 3;

Pernyataan SQL berikut menunjukkan contoh yang setara untuk Oracle:

Contoh

SELECT \* FROM Customers

WHERE Country='Germany'

FETCH FIRST 3 ROWS ONLY;

**SQL Min and Max**

Fungsi SQL MIN() dan MAX()

The MIN()fungsi mengembalikan nilai terkecil dari kolom yang dipilih.

The MAX()fungsi mengembalikan nilai terbesar dari kolom yang dipilih.

MIN() Sintaks

SELECT MIN(column\_name)

FROM table\_name

WHERE condition;

MAX() Sintaks

SELECT MAX(column\_name)

FROM table\_name

WHERE condition;

MIN() Contoh

Pernyataan SQL berikut menemukan harga produk termurah:

Contoh

SELECT MIN(Price) AS SmallestPrice

FROM Products;

MAX() Contoh

Pernyataan SQL berikut menemukan harga produk paling mahal:

Contoh

SELECT MAX(Price) AS LargestPrice

FROM Products;

**SQL Count, Avg, Sum**

Fungsi SQL COUNT(), AVG() dan SUM()

Fungsi SQL COUNT(), AVG() dan SUM()

The COUNT()fungsi mengembalikan jumlah baris yang cocok dengan kriteria tertentu.

COUNT() Sintaks

SELECT COUNT(column\_name)

FROM table\_name

WHERE condition;

The AVG()fungsi mengembalikan nilai rata-rata dari kolom numerik.

Sintaks AVG()

SELECT AVG(column\_name)

FROM table\_name

WHERE condition;

The SUM()mengembalikan fungsi jumlah total kolom numerik.

SUM() Sintaks

SELECT SUM(column\_name)

FROM table\_name

WHERE condition;

COUNT() Contoh

Pernyataan SQL berikut menemukan jumlah produk:

Contoh

SELECT COUNT(ProductID)

FROM Products;

Catatan: Nilai NULL tidak dihitung.

AVG() Contoh

Pernyataan SQL berikut menemukan harga rata-rata semua produk:

Contoh

SELECT AVG(Price)

FROM Products;

Catatan: Nilai NULL diabaikan.

SUM() Contoh

Pernyataan SQL berikut menemukan jumlah bidang "Kuantitas" di tabel "Detail Pesanan":

Contoh

SELECT SUM(Quantity)

FROM OrderDetails;

**SQL Like**

Operator SEPERTI SQL

The LIKEoperator yang digunakan dalam WHEREklausul untuk mencari pola yang ditentukan dalam kolom.

Ada dua wildcard yang sering digunakan bersama dengan LIKEoperator:

Tanda persen (%) mewakili nol, satu, atau beberapa karakter

Tanda garis bawah (\_) mewakili satu karakter tunggal

Catatan: MS Access menggunakan tanda bintang (\*) sebagai ganti tanda persen (%), dan tanda tanya (?) sebagai ganti garis bawah (\_).

Tanda persen dan garis bawah juga dapat digunakan dalam kombinasi!

SEPERTI Sintaks

SELECT column1, column2, ...

FROM table\_name

WHERE columnN LIKE pattern;

Tip: Anda juga dapat menggabungkan sejumlah kondisi menggunakan ANDatau ORoperator.

Contoh SEPERTI SQL

Pernyataan SQL berikut memilih semua pelanggan dengan NamaPelanggan dimulai dengan "a":

Contoh

SELECT \* FROM Customers

WHERE CustomerName LIKE 'a%';

Pernyataan SQL berikut memilih semua pelanggan dengan NamaPelanggan yang diakhiri dengan "a":

Contoh

SELECT \* FROM Customers

WHERE CustomerName LIKE '%a';

Pernyataan SQL berikut memilih semua pelanggan dengan NamaPelanggan yang memiliki "atau" di posisi apa pun:

Contoh

SELECT \* FROM Customers

WHERE CustomerName LIKE '%or%';

Pernyataan SQL berikut memilih semua pelanggan dengan NamaPelanggan yang memiliki "r" di posisi kedua:

Contoh

SELECT \* FROM Customers

WHERE CustomerName LIKE '\_r%';

Pernyataan SQL berikut memilih semua pelanggan dengan NamaPelanggan yang dimulai dengan "a" dan panjangnya minimal 3 karakter:

Contoh

SELECT \* FROM Customers

WHERE CustomerName LIKE 'a\_\_%';

Pernyataan SQL berikut memilih semua pelanggan dengan ContactName yang dimulai dengan "a" dan diakhiri dengan "o":

Contoh

SELECT \* FROM Customers

WHERE ContactName LIKE 'a%o';

Pernyataan SQL berikut memilih semua pelanggan dengan NamaPelanggan yang TIDAK dimulai dengan "a":

Contoh

SELECT \* FROM Customers

WHERE CustomerName NOT LIKE 'a%';

**SQL Wilcards**

Karakter Wildcard SQL

Karakter wildcard digunakan untuk menggantikan satu atau lebih karakter dalam sebuah string.

Karakter wildcard digunakan dengan operator. The operator yang digunakan dalam klausul untuk mencari pola yang ditentukan dalam kolom. LIKELIKE WHERE

Karakter Wildcard di MS Access

Symbol Description Example

\* Represents zero or more characters bl\* finds bl, black, blue, and blob

? Represents a single character h?t finds hot, hat, and hit

[] Represents any single character within the brackets h[oa]t finds hot and hat, but not hit

! Represents any character not in the brackets h[!oa]t finds hit, but not hot and hat

- Represents any single character within the specified range c[a-b]t finds cat and cbt

# Represents any single numeric character 2#5 finds 205, 215, 225, 235, 245, 255, 265, 275, 285, and 295

Karakter Wildcard di SQL Server

Symbol Description Example

% Represents zero or more characters bl% finds bl, black, blue, and blob

\_ Represents a single character h\_t finds hot, hat, and hit

[] Represents any single character within the brackets h[oa]t finds hot and hat, but not hit

^ Represents any character not in the brackets h[^oa]t finds hit, but not hot and hat

- Represents any single character within the specified range c[a-b]t finds cat and cbt

Semua wildcard juga dapat digunakan dalam kombinasi!

Berikut adalah beberapa contoh yang menunjukkan LIKEoperator yang berbeda dengan wildcard '%' dan '\_':

LIKE Operator Description

WHERE CustomerName LIKE 'a%' Finds any values that starts with "a"

WHERE CustomerName LIKE '%a' Finds any values that ends with "a"

WHERE CustomerName LIKE '%or%' Finds any values that have "or" in any position

WHERE CustomerName LIKE '\_r%' Finds any values that have "r" in the second position

WHERE CustomerName LIKE 'a\_\_%' Finds any values that starts with "a" and are at least 3 characters in length

WHERE ContactName LIKE 'a%o' Finds any values that starts with "a" and ends with "o"

Menggunakan % Wildcard

Pernyataan SQL berikut memilih semua pelanggan dengan Kota yang dimulai dengan "ber":

Contoh

SELECT \* FROM Customers

WHERE City LIKE 'ber%';

Pernyataan SQL berikut memilih semua pelanggan dengan Kota yang berisi pola "es":

Contoh

SELECT \* FROM Customers

WHERE City LIKE '%es%';

Menggunakan \_ Wildcard

Pernyataan SQL berikut memilih semua pelanggan dengan Kota yang dimulai dengan karakter apa pun, diikuti oleh "ondon":

Contoh

SELECT \* FROM Customers

WHERE City LIKE '\_ondon';

Pernyataan SQL berikut memilih semua pelanggan dengan Kota yang dimulai dengan "L", diikuti oleh karakter apa pun, diikuti oleh "n", diikuti oleh karakter apa pun, diikuti oleh "on":

Contoh

SELECT \* FROM Customers

WHERE City LIKE 'L\_n\_on';

Menggunakan [charlist] Wildcard

Pernyataan SQL berikut memilih semua pelanggan dengan Kota yang dimulai dengan "b", "s", atau "p":

Contoh

SELECT \* FROM Customers

WHERE City LIKE '[bsp]%';

Pernyataan SQL berikut memilih semua pelanggan dengan Kota yang dimulai dengan "a", "b", atau "c":

Contoh

SELECT \* FROM Customers

WHERE City LIKE '[a-c]%';

Menggunakan Wildcard [!charlist]

Dua pernyataan SQL berikut memilih semua pelanggan dengan Kota TIDAK dimulai dengan "b", "s", atau "p":

Contoh

SELECT \* FROM Customers

WHERE City LIKE '[!bsp]%';

Atau:

Contoh

SELECT \* FROM Customers

WHERE City NOT LIKE '[bsp]%';

**SQL In**

Operator SQL IN

The INOperator memungkinkan Anda untuk menentukan beberapa nilai dalam WHEREklausa.

The INoperator adalah singkatan untuk beberapa ORkondisi.

DI Sintaks

SELECT column\_name(s)

FROM table\_name

WHERE column\_name IN (value1, value2, ...);

atau:

SELECT column\_name(s)

FROM table\_name

WHERE column\_name IN (SELECT STATEMENT);

IN Operator Contoh

Pernyataan SQL berikut memilih semua pelanggan yang berlokasi di "Jerman", "Prancis" atau "Inggris Raya":

Contoh

SELECT \* FROM Customers

WHERE Country IN ('Germany', 'France', 'UK');

Pernyataan SQL berikut memilih semua pelanggan yang TIDAK berlokasi di "Jerman", "Prancis" atau "Inggris Raya":

Contoh

SELECT \* FROM Customers

WHERE Country NOT IN ('Germany', 'France', 'UK');

Pernyataan SQL berikut memilih semua pelanggan yang berasal dari negara yang sama dengan pemasok:

Contoh

SELECT \* FROM Customers

WHERE Country IN (SELECT Country FROM Suppliers);

**SQL Between**

Operator SQL ANTARA

The BETWEENnilai Operator menyeleksi dalam kisaran tertentu. Nilai dapat berupa angka, teks, atau tanggal.

The BETWEENoperator adalah inklusif: mulai dan nilai-nilai akhir disertakan.

ANTARA Sintaks

SELECT column\_name(s)

FROM table\_name

WHERE column\_name BETWEEN value1 AND value2;

ANTARA Contoh

Pernyataan SQL berikut memilih semua produk dengan harga antara 10 dan 20:

Contoh

SELECT \* FROM Products

WHERE Price BETWEEN 10 AND 20;

BUKAN ANTARA Contoh

Untuk menampilkan produk di luar jangkauan contoh sebelumnya, gunakan NOT BETWEEN:

Contoh

SELECT \* FROM Products

WHERE Price NOT BETWEEN 10 AND 20;

ANTARA dengan IN Contoh

Pernyataan SQL berikut memilih semua produk dengan harga antara 10 dan 20. Selain itu; jangan tampilkan produk dengan ID Kategori 1,2, atau 3:

Contoh

SELECT \* FROM Products

WHERE Price BETWEEN 10 AND 20

AND CategoryID NOT IN (1,2,3);

Contoh ANTARA Nilai Teks

Pernyataan SQL berikut memilih semua produk dengan ProductName antara Carnarvon Tigers dan Mozzarella di Giovanni:

Contoh

SELECT \* FROM Products

WHERE ProductName BETWEEN 'Carnarvon Tigers' AND 'Mozzarella di Giovanni'

ORDER BY ProductName;

Pernyataan SQL berikut memilih semua produk dengan ProductName antara Carnarvon Tigers dan Bumbu Cajun Chef Anton:

Contoh

SELECT \* FROM Products

WHERE ProductName BETWEEN "Carnarvon Tigers" AND "Chef Anton's Cajun Seasoning"

ORDER BY ProductName;

BUKAN ANTARA Nilai Teks Contoh

Pernyataan SQL berikut memilih semua produk dengan ProductName bukan antara Carnarvon Tigers dan Mozzarella di Giovanni:

Contoh

SELECT \* FROM Products

WHERE ProductName NOT BETWEEN 'Carnarvon Tigers' AND 'Mozzarella di Giovanni'

ORDER BY ProductName;

Tabel Contoh

Di bawah ini adalah pilihan dari tabel "Pesanan" di database sampel Northwind:

OrderID CustomerID EmployeeID OrderDate ShipperID

10248 90 5 7/4/1996 3

10249 81 6 7/5/1996 1

10250 34 4 7/8/1996 2

10251 84 3 7/9/1996 1

10252 76 4 7/10/1996 2

Contoh ANTARA Tanggal

Pernyataan SQL berikut memilih semua pesanan dengan OrderDate antara '01-July-1996' dan '31-July-1996':

Contoh

SELECT \* FROM Orders

WHERE OrderDate BETWEEN #07/01/1996# AND #07/31/1996#;

ATAU:

Contoh

SELECT \* FROM Orders

WHERE OrderDate BETWEEN '1996-07-01' AND '1996-07-31';

**SQL Aliases**

SQL Alias ​​​​

Alias ​​​​SQL digunakan untuk memberi tabel, atau kolom dalam tabel, nama sementara.

Alias ​​​​sering digunakan untuk membuat nama kolom lebih mudah dibaca.

Alias ​​​​hanya ada selama kueri itu.

Alias ​​​​dibuat dengan ASkata kunci.

Sintaks Kolom Alias ​​​​

SELECT column\_name AS alias\_name

FROM table\_name;

Sintaks Tabel Alias

SELECT column\_name(s)

FROM table\_name AS alias\_name;

Alias ​​​​untuk Contoh Kolom

Pernyataan SQL berikut membuat dua alias, satu untuk kolom CustomerID dan satu lagi untuk kolom CustomerName:

Contoh

SELECT CustomerID AS ID, CustomerName AS Customer

FROM Customers;

Pernyataan SQL berikut membuat dua alias, satu untuk kolom NamaPelanggan dan satu lagi untuk kolom NamaKontak. Catatan: Ini membutuhkan tanda kutip ganda atau tanda kurung siku jika nama alias berisi spasi:

Contoh

SELECT CustomerName AS Customer, ContactName AS [Contact Person]

FROM Customers;

Pernyataan SQL berikut membuat alias bernama "Alamat" yang menggabungkan empat kolom (Alamat, Kode Pos, Kota, dan Negara):

Contoh

SELECT CustomerName, Address + ', ' + PostalCode + ' ' + City + ', ' + Country AS Address

FROM Customers;

Catatan: Agar pernyataan SQL di atas berfungsi di MySQL, gunakan yang berikut ini:

SELECT CustomerName, CONCAT(Address,', ',PostalCode,', ',City,', ',Country) AS Address

FROM Customers;

Alias ​​​​untuk Contoh Tabel

Pernyataan SQL berikut memilih semua pesanan dari pelanggan dengan CustomerID=4 (Around the Horn). Kami menggunakan tabel "Pelanggan" dan "Pesanan", dan memberi mereka alias tabel masing-masing "c" dan "o" (Di sini kami menggunakan alias untuk membuat SQL lebih pendek):

Contoh

SELECT o.OrderID, o.OrderDate, c.CustomerName

FROM Customers AS c, Orders AS o

WHERE c.CustomerName='Around the Horn' AND c.CustomerID=o.CustomerID;

Pernyataan SQL berikut ini sama seperti di atas, tetapi tanpa alias:

Contoh

SELECT Orders.OrderID, Orders.OrderDate, Customers.CustomerName

FROM Customers, Orders

WHERE Customers.CustomerName='Around the Horn' AND Customers.CustomerID=Orders.CustomerID;

Alias ​​​​dapat berguna ketika:

Ada lebih dari satu tabel yang terlibat dalam kueri

Fungsi digunakan dalam kueri

Nama kolom besar atau tidak terlalu mudah dibaca

Dua atau lebih kolom digabungkan bersama

Uji Diri Anda Dengan Latihan

Alias ​​​​untuk Contoh Kolom

Pernyataan SQL berikut membuat dua alias, satu untuk kolom CustomerID dan satu lagi untuk kolom CustomerName:

Contoh

SELECT CustomerID AS ID, CustomerName AS Customer

FROM Customers;

Pernyataan SQL berikut membuat dua alias, satu untuk kolom NamaPelanggan dan satu lagi untuk kolom NamaKontak. Catatan: Ini membutuhkan tanda kutip ganda atau tanda kurung siku jika nama alias berisi spasi:

Contoh

SELECT CustomerName AS Customer, ContactName AS [Contact Person]

FROM Customers;

Pernyataan SQL berikut membuat alias bernama "Alamat" yang menggabungkan empat kolom (Alamat, Kode Pos, Kota, dan Negara):

Contoh

SELECT CustomerName, Address + ', ' + PostalCode + ' ' + City + ', ' + Country AS Address

FROM Customers;

Catatan: Agar pernyataan SQL di atas berfungsi di MySQL, gunakan yang berikut ini:

SELECT CustomerName, CONCAT(Address,', ',PostalCode,', ',City,', ',Country) AS Address

FROM Customers;

Alias ​​​​untuk Contoh Tabel

Pernyataan SQL berikut memilih semua pesanan dari pelanggan dengan CustomerID=4 (Around the Horn). Kami menggunakan tabel "Pelanggan" dan "Pesanan", dan memberi mereka alias tabel masing-masing "c" dan "o" (Di sini kami menggunakan alias untuk membuat SQL lebih pendek):

Contoh

SELECT o.OrderID, o.OrderDate, c.CustomerName

FROM Customers AS c, Orders AS o

WHERE c.CustomerName='Around the Horn' AND c.CustomerID=o.CustomerID;

Pernyataan SQL berikut ini sama seperti di atas, tetapi tanpa alias:

Contoh

SELECT Orders.OrderID, Orders.OrderDate, Customers.CustomerName

FROM Customers, Orders

WHERE Customers.CustomerName='Around the Horn' AND Customers.CustomerID=Orders.CustomerID;

Alias ​​​​dapat berguna ketika:

Ada lebih dari satu tabel yang terlibat dalam kueri

Fungsi digunakan dalam kueri

Nama kolom besar atau tidak terlalu mudah dibaca

Dua atau lebih kolom digabungkan bersama

Uji Diri Anda Dengan Latihan

**SQL Joins**

SQL BERGABUNG

Sebuah JOINklausul digunakan untuk menggabungkan baris dari dua atau lebih tabel, berdasarkan kolom terkait antara mereka.

Mari kita lihat pilihan dari tabel "Pesanan":

OrderID CustomerID OrderDate

10308 2 1996-09-18

10309 37 1996-09-19

10310 77 1996-09-20

Kemudian, lihat pilihan dari tabel "Pelanggan":

CustomerID CustomerName ContactName Country

1 Alfreds Futterkiste Maria Anders Germany

2 Ana Trujillo Emparedados y helados Ana Trujillo Mexico

3 Antonio Moreno Taquería Antonio Moreno Mexico

Perhatikan bahwa kolom "IDPelanggan" di tabel "Pesanan" merujuk ke "IDPelanggan" di tabel "Pelanggan". Hubungan antara dua tabel di atas adalah kolom "CustomerID".

Kemudian, kita dapat membuat pernyataan SQL berikut (yang berisi INNER JOIN), yang memilih catatan yang memiliki nilai yang cocok di kedua tabel:

Contoh

SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate

FROM Orders

INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;

dan itu akan menghasilkan sesuatu seperti ini:

OrderID CustomerName OrderDate

10308 Ana Trujillo Emparedados y helados 9/18/1996

10365 Antonio Moreno Taquería 11/27/1996

10383 Around the Horn 12/16/1996

10355 Around the Horn 11/15/1996

10278 Berglunds snabbköp 8/12/1996

Berbagai Jenis SQL BERGABUNG

Berikut adalah berbagai jenis JOIN dalam SQL:

(INNER) JOIN: Mengembalikan record yang memiliki nilai yang cocok di kedua tabel

LEFT (OUTER) JOIN: Mengembalikan semua record dari tabel kiri, dan record yang cocok dari tabel kanan

RIGHT (OUTER) JOIN: Mengembalikan semua record dari tabel kanan, dan record yang cocok dari tabel kiri

FULL (OUTER) JOIN: Mengembalikan semua catatan ketika ada kecocokan di tabel kiri atau kanan

**SQL Inner Join**

SQL INNER JOIN Kata Kunci

Kata INNER JOINkunci memilih rekaman yang memiliki nilai yang cocok di kedua tabel.

Contoh SQL INNER JOIN

Pernyataan SQL berikut memilih semua pesanan dengan informasi pelanggan:

Contoh

SELECT Orders.OrderID, Customers.CustomerName

FROM Orders

INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;

Catatan: Kata INNER JOINkunci memilih semua baris dari kedua tabel selama ada kecocokan antar kolom. Jika ada catatan di tabel "Pesanan" yang tidak cocok dengan "Pelanggan", pesanan ini tidak akan ditampilkan!

GABUNG Tiga Meja

Pernyataan SQL berikut memilih semua pesanan dengan informasi pelanggan dan pengirim:

Contoh

SELECT Orders.OrderID, Customers.CustomerName, Shippers.ShipperName

FROM ((Orders

INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID)

INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID);

**SQL Union**

The SQL UNION Operator

The UNION operator is used to combine the result-set of two or more SELECT statements.

Every SELECT statement within UNION must have the same number of columns

The columns must also have similar data types

The columns in every SELECT statement must also be in the same order

UNION Syntax

SELECT column\_name(s) FROM table1

UNION

SELECT column\_name(s) FROM table2;

UNION ALL Syntax

The UNION operator selects only distinct values by default. To allow duplicate values, use UNION ALL:

SELECT column\_name(s) FROM table1

UNION ALL

SELECT column\_name(s) FROM table2;

Note: The column names in the result-set are usually equal to the column names in the first SELECT statement.

Pernyataan SQL berikut mengembalikan kota (hanya nilai yang berbeda) dari tabel "Pelanggan" dan "Pemasok":

Contoh

SELECT City FROM Customers

UNION

SELECT City FROM Suppliers

ORDER BY City;

Catatan: Jika beberapa pelanggan atau pemasok memiliki kota yang sama, setiap kota hanya akan dicantumkan satu kali, karena UNIONhanya memilih nilai yang berbeda. Gunakan UNION ALLjuga untuk memilih nilai duplikat!

SQL UNION ALL Contoh

Pernyataan SQL berikut mengembalikan kota (nilai duplikat juga) dari tabel "Pelanggan" dan "Pemasok":

Contoh

SELECT City FROM Customers

UNION ALL

SELECT City FROM Suppliers

ORDER BY City;

SQL UNION Dengan WHERE

Pernyataan SQL berikut mengembalikan kota-kota Jerman (hanya nilai yang berbeda) dari tabel "Pelanggan" dan "Pemasok":

Contoh

SELECT City, Country FROM Customers

WHERE Country='Germany'

UNION

SELECT City, Country FROM Suppliers

WHERE Country='Germany'

ORDER BY City;

SQL UNION SEMUA Dengan MANA

Pernyataan SQL berikut mengembalikan kota-kota Jerman (nilai duplikat juga) dari tabel "Pelanggan" dan "Pemasok":

Contoh

SELECT City, Country FROM Customers

WHERE Country='Germany'

UNION ALL

SELECT City, Country FROM Suppliers

WHERE Country='Germany'

ORDER BY City;

Contoh UNION lainnya

Pernyataan SQL berikut mencantumkan semua pelanggan dan pemasok:

Contoh

SELECT 'Customer' AS Type, ContactName, City, Country

FROM Customers

UNION

SELECT 'Supplier', ContactName, City, Country

FROM Suppliers;

Notice the "AS Type" above - it is an alias. SQL Aliases are used to give a table or a column a temporary name. An alias only exists for the duration of the query. So, here we have created a temporary column named "Type", that list whether the contact person is a "Customer" or a "Supplier".

**SQL Group by**

Pernyataan SQL GROUP BY

The GROUP BYkelompok pernyataan baris yang memiliki nilai yang sama dalam baris Singkatnya, seperti "menemukan jumlah pelanggan di masing-masing negara".

The GROUP BYpernyataan sering digunakan dengan fungsi agregat ( COUNT(), MAX(), MIN(), SUM(), AVG()) untuk kelompok hasil-set oleh satu atau lebih kolom.

KELOMPOK BERDASARKAN Sintaks

SELECT column\_name(s)

FROM table\_name

WHERE condition

GROUP BY column\_name(s)

ORDER BY column\_name(s);

Contoh SQL GROUP BY

Pernyataan SQL berikut mencantumkan jumlah pelanggan di setiap negara:

Contoh

SELECT COUNT(CustomerID), Country

FROM Customers

GROUP BY Country;

Pernyataan SQL berikut mencantumkan jumlah pelanggan di setiap negara, diurutkan dari tinggi ke rendah:

Contoh

SELECT COUNT(CustomerID), Country

FROM Customers

GROUP BY Country

ORDER BY COUNT(CustomerID) DESC;

GROUP BY Dengan JOIN Contoh

Pernyataan SQL berikut mencantumkan jumlah pesanan yang dikirim oleh setiap pengirim:

Contoh

SELECT Shippers.ShipperName, COUNT(Orders.OrderID) AS NumberOfOrders FROM Orders

LEFT JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID

GROUP BY ShipperName;

**SQL having**

Klausa SQL HAVING

The HAVINGklausul ditambahkan ke SQL karena WHEREkata kunci tidak dapat digunakan dengan fungsi agregat.

MEMILIKI Sintaks

SELECT column\_name(s)

FROM table\_name

WHERE condition

GROUP BY column\_name(s)

HAVING condition

ORDER BY column\_name(s);

SQL MEMILIKI Contoh

Pernyataan SQL berikut mencantumkan jumlah pelanggan di setiap negara. Hanya sertakan negara dengan lebih dari 5 pelanggan:

Contoh

SELECT COUNT(CustomerID), Country

FROM Customers

GROUP BY Country

HAVING COUNT(CustomerID) > 5;

Pernyataan SQL berikut mencantumkan jumlah pelanggan di setiap negara, diurutkan dari tinggi ke rendah (Hanya sertakan negara dengan lebih dari 5 pelanggan):

Contoh

SELECT COUNT(CustomerID), Country

FROM Customers

GROUP BY Country

HAVING COUNT(CustomerID) > 5

ORDER BY COUNT(CustomerID) DESC;

**SQL Null Fungsions**

Fungsi SQL IFNULL(), ISNULL(), COALESCE(), dan NVL()

Lihat tabel "Produk" berikut:

P\_Id ProductName UnitPrice UnitsInStock UnitsOnOrder

1 Jarlsberg 10.45 16 15

2 Mascarpone 32.56 23

3 Gorgonzola 15.67 9 20

Misalkan kolom "UnitsOnOrder" adalah opsional, dan mungkin berisi nilai NULL.

Perhatikan pernyataan SELECT berikut:

SELECT ProductName, UnitPrice \* (UnitsInStock + UnitsOnOrder)

FROM Products;

Dalam contoh di atas, jika salah satu nilai "UnitsOnOrder" adalah NULL, hasilnya akan NULL.

Solusi

MySQL

Fungsi MySQL IFNULL()memungkinkan Anda mengembalikan nilai alternatif jika ekspresi NULL:

SELECT ProductName, UnitPrice \* (UnitsInStock + IFNULL(UnitsOnOrder, 0))

FROM Products;

atau kita bisa menggunakan fungsinya, seperti ini: COALESCE()

SELECT ProductName, UnitPrice \* (UnitsInStock + COALESCE(UnitsOnOrder, 0))

FROM Products;

SQL Server

Fungsi SQL Server ISNULL()memungkinkan Anda mengembalikan nilai alternatif saat ekspresi NULL:

SELECT ProductName, UnitPrice \* (UnitsInStock + ISNULL(UnitsOnOrder, 0))

FROM Products;

Akses MS

Fungsi MS Access IsNull()mengembalikan TRUE (-1) jika ekspresi adalah nilai null, jika tidak FALSE (0):

SELECT ProductName, UnitPrice \* (UnitsInStock + IIF(IsNull(UnitsOnOrder), 0, UnitsOnOrder))

FROM Products;

Peramal

Fungsi Oracle NVL()mencapai hasil yang sama:

SELECT ProductName, UnitPrice \* (UnitsInStock + NVL(UnitsOnOrder, 0))

FROM Products;

**SQL Operators**

Operator Aritmatika SQL

Operator Description Example

+ Add

- Subtract

\* Multiply

/ Divide

% Modulo

Operator SQL Bitwise

Operator Description

& Bitwise AND

| Bitwise OR

^ Bitwise exclusive OR

Operator Perbandingan SQL

Operator Description Example

= Equal to

> Greater than

< Less than

>= Greater than or equal to

<= Less than or equal to

<> Not equal to

Operator Senyawa SQL

Operator Description

+= Add equals

-= Subtract equals

\*= Multiply equals

/= Divide equals

%= Modulo equals

&= Bitwise AND equals

^-= Bitwise exclusive equals

|\*= Bitwise OR equals

Operator Logika SQL

Operator Description Example

ALL TRUE if all of the subquery values meet the condition

AND TRUE if all the conditions separated by AND is TRUE

ANY TRUE if any of the subquery values meet the condition

BETWEEN TRUE if the operand is within the range of comparisons

EXISTS TRUE if the subquery returns one or more records

IN TRUE if the operand is equal to one of a list of expressions

LIKE TRUE if the operand matches a pattern

NOT Displays a record if the condition(s) is NOT TRUE

OR TRUE if any of the conditions separated by OR is TRUE

SOME TRUE if any of the subquery values meet the condition