Nama :wendi pebriyanto

Nim : 200250502066

Kelas : TI.B semester 3

Mata kuliah : SISTEM BASIS DATA

Dosen : MUSLIADI KH, S.kom M.kom

SQL TUTORIAL

1. Structured Query Language atau SQL mungkin merupakan poin penting dalam sebuah studi teknik komputer jaringan. Keberadaan SQL sebagai sebuah bahasa pemrograman memiliki sejarah dan perkembangan yang panjang.Tentu saja perkembangan tersebut dipengaruhi oleh perkembangan dunia teknologi yang dari tahun ke tahun semakin membaik dan terus diperbaharui. Sehingga mau tidak mau, banyak [*software*](https://www.dewaweb.com/blog/kelebihan-dan-kekurangan-menggunakan-open-source-software/) diciptakan dengan teknologi baru namun tidak menghilangkan keberadaan SQL ini
2. SQL HOME

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

1. SQL INTRO SQL

adalah singkatan dari Structured Query LanguageSQL memungkinkan Anda mengakses dan memanipulasi databaseSQL menjadi standar American National Standards Institute (ANSI) pada 1986, dan International Organization for Standardization (ISO) pada 1987

1. SQL syntax Database Tables

A database most often contains one or more tables. Each table is identified by a name (e.g. "Customers" or "Orders"). Tables contain records (rows) with data.In this tutorial we will use the well-known Northwind sample database (included in MS Access and MS SQL Server).Below is a selection from the "Customers" table:

The table above contains five records (one for each customer) and seven columns (CustomerID, CustomerName, ContactName, Address, City, PostalCode, and Country).

1. SQL SELECT

Pernyataan SELECT digunakan untuk memilih data dari database.Data yang dikembalikan disimpan dalam tabel hasil, yang disebut kumpulan hasil.

Contoh:

SELECT column1, column2,...  
FROM table\_name;

1. SQL SELECT DISTINCT

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

Contoh:

SELECT DISTINCT column1, column2,...  
FROM table\_name;

1. SQL WHERE

Klausa WHERE digunakan untuk memfilter record.Ini digunakan untuk mengekstrak hanya catatan yang memenuhi kondisi tertentu.

Contoh:

SELECT column1, column2, ...  
FROM table\_name  
WHERE condition;

1. SQL AND, OR and NOT

Operator SQL AND, OR dan NOTKlausa WHERE dapat digabungkan dengan operator AND, OR, dan NOT.Operator AND dan OR digunakan untuk memfilter record berdasarkan lebih dari satu kondisi: Operator AND menampilkan record jika semua kondisi yang dipisahkan oleh AND adalahTRUE. Operator OR menampilkan record jika salah satu kondisi yang dipisahkan oleh OR adalah TRUE.Operator NOT menampilkan record jika kondisi NOT TRUE.

Contoh:

SELECT *column1*,*column2, ...*  
FROM *table\_name*  
WHERE *condition1* AND *condition2* AND *condition3 ...*;

OR Syntax

SELECT *column1*,*column2, ...*  
FROM *table\_name*  
WHERE *condition1* OR *condition2* OR *condition3 ...*;

NOT Syntax

SELECT *column1*,*column2, ...*  
FROM *table\_name*  
WHERE NOT *condition*;

# SQL ORDER BY

# SQL ORDER BY Kata KunciKata kunci ORDER BY digunakan untuk mengurutkan kumpulan hasil dalam urutan menaik atau menurun.Kata kunci ORDER BY mengurutkan catatan dalam urutan menaik secara default. Untuk mengurutkan catatan dalam urutan menurun, gunakan kata kunci DESC.

# SQL INSERT INTO

The SQL INSERT INTO StatementThe INSERT INTO statement is used to insert new records in a table.INSERT INTO SyntaxIt is possible to write the INSERT INTO statement in two ways:

1. Specify both the column names and the values to be inserted:

INSERT INTO table\_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);

1. 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, sintaks INSERT INTO adalah sebagai berikut:

INSERT INTO table\_name  
VALUES (value1, value2, value3, ...);

1. SQL NULL Values

Apa itu Nilai NULL? 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.

IS NULL Syntax

SELECT *column\_names*FROM *table\_name*  
WHERE *column\_name* IS NULL;

IS NOT NULL Syntax

SELECT *column\_names*FROM *table\_name*  
WHERE *column\_name* IS NOT NULL;

1. SQL Select Top

SQL Select Top digunakan untuk menentukan jumlah record yang akan di tampilkan. Kalusa SELECT TOP berguna pad atabel dengan row data yang banyak dengan ribuan record data. Menampilkan sejumlah besar record data dapat berdampak pada kinerja database.

Syntax :

SELECT nama\_kolom(s)

FROM nama\_table

WHERE kondisi

LIMIT nomor;

Contoh :

SELECT \*FROM Customers

LIMIT 3;

1. SQL Min and Max

SQL Max digunakan untuk mendapatkan nilai terbesar dari data-data yang ada.

Syntax dasar :

SELECT MAX (column\_name) FROM table\_name

Contoh :

SELECT MAX(harga) AS Harga\_termahal FROM order

SQL Min digunakan untuk mendapatkan nilai terkecil dari data-data yang ada.

Syntax dasar :

SELECT MIN (column\_name) FROM table\_name

Contoh:

SELECT MIN(harga) AS Harga\_termurah FROM order

1. SQL Count, Avg, Sum

SQL Count Digunakan untuk menghitung jumlah (cacah) suatu data.

Syntax dasar :

SELECT COUNT (column\_name) FROM table\_name

Contoh :

SELECT COUNT(id) AS Jumlah\_tamu FROM tabeldata

SQL Avg digunakan untuk menghitung nilai-rata-rata dari suatu data.

Syntax dasar :

SELECT AVG (column\_name) FROM table\_name

Contoh :

SELECT AVG(harga) AS Harga\_rata2FROM order

SQL Sum digunakan untuk mendapatkan nilai total penjumlahan dari data-data yang ada.

Syntax dasar :

SELECT SUM (column\_name) FROM table\_name

Contoh :

SELECT SUM(harga) AS Harga\_total FROM order

1. SQL Like

SQL Like digunakan bersama dengan perintah Where, untuk proses pencarian data dengan spesifikasi tertentu.

Syntax dasar:

SELECT column\_name(s)

FROM table\_name

WHERE column\_name LIKE pattern

Contoh :

SELECT\*FROM tabeldata WHERE nama

LIKE’z%’

Ket : contoh di atas digunakan untuk pencarian berdasarkan kolom nama yang berhuruf depan”z”

SELECT\*FROM tabeldata WHERE nama

LIKE’z%’

Ket : contoh di atas digunakan untuk pencarian berdasarkan kolom nama yang berhuruf belakang”z”

1. SQL Wildcards

SQL Wildcards di gunakan untuk menggantikan satu atau lebih krakter dalam sebuah string. Wildcards di gunakan dengan operator LIKE . Operator LIKE digunakan dalam klausa WHERE untuk mencari pola tertentu dalam kolom.

Contoh :

SELECT \* FROM Customer

WHERE kota LIKE ‘ber%’:

1. SQL In

SQL In digunakan untuk pencarian data menggunakan lebih dari satu filter pada perintah Where

Syntas dasar:

SELECT column-name(s)

FROM table\_name

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

Contoh :

SELECT\*FROM tabeldata WHERE kota IN

(‘kediri’,malang’)

BE

1. SQL Between

SQL Between digunakan untuk menentukan jangkauan pencarian.

Syntax dasar:

SELECT column\_name(s)

FROM table-name

WHERE column-name

BETWEEN value1 AND value2

Contoh :

SELECT\*FROM\*tabeldata WHERE id

BETWEEN 15 and 45

Ket : contoh di atas digunakanuntuk mencari data yang memiliki nomor idantara 15 dan 45.

1. SQL Aliases

As memungkinkan anda untuk menberi nama ulang/nama alias/nama baru pada sebuah kolom atau tabel tanpa mengubah nama aslinya. Beberapa programmer sering menggunakan fitur AS ini untuk menbantu mereka dalam mengenali sebuah tabel atau kolom serta menpermudah dalam penulisan query secara ringkas.

Syntax dasar :

SELECT nama\_kolom as nama\_alias

FROM nama\_tabel

Contoh :

SELECT ID as produkID, ProductName AS produk

FROM Produks;

SELECT p. ID FROM produks AS p WHERE p. ID = 2 ;

1. SQL Joins

SQL Joins digunakan untuk menggabungkan baris dari dua atau lebih tabel, berdasarkan kolom terkait diantara mereka.

Contoh :

SELECT Orders, OrderID,

Customers, customersName,

Orders. OrderDate

FROM Orders

INNER Join customers ON

Orders, CustomerID = Customers. CustomerID ;

1. SQL Inner Join

SQL Inner Join digunakan untuk menghasilakan baris data dengan cara menggabungkan 2 buah tabel atau lebih menggunakan pasangan data yang match pada masing-masingtabel. Perintah ini sama dengan join yang sering di gunakan.

Contoh :

SELECT tabeldata.nama.tabeldata.email,orde.no\_order

FROM tabeldata

INNER JOIN order

ON tabeldata id=order.id

ORDER BY tabeldata.nama

1. SQL Left Join

SQL Left Join digunakan untuk menghasilkan baris data dari tabel kiri (nama tabel pertama) yang ada pasangan datanya pada tabel kanan (nama tabel kedua).

Syntax dasar :

SELECT column\_name(s)

FROM table\_name1

LEFT JOIN table\_name2

ON table\_name1.column\_name=table\_name2.

column\_name

contoh :

SELECT tabeldata.nama,tabeldata.email,order.no\_order

FROM tabeldata

LEFT JOIN order

ON tabeldata.id=order.id

ORDER BY tabeldata.nama

1. SQL Right Join

SQL Right Join Digunakan untuk menghasilkan baris data dari tabel kanan (nama tabel kedua) yang tidak ada pasangan datanya pada tabel kiri (nama tabel pertama).

Syntax dasar :

SELECT column\_name(s)

FROM table\_name1

RIGHT JOIN table\_name2

ON table\_name1.column\_name=table\_name2

column\_name

contoh :

SELECT tabeldata.nama,tabeldata.emailmorder.no\_order

FROM tabeldata

RIGHT JOIN order

ON tabeldata.id=order.i

ORDER BY tabeldata.nama

1. SQL Full Join

Digunakan untuk menghasilkan baris data jika ada data yang sama pada salah satu tabel.

Syntax dasar :

SELECT column\_name(s)

FROM table\_name1

FULL JOIN table\_name2

ON table\_name1.column\_name=table\_name2

column\_name

Contoh :

SELECT tabeldata.nama,tabeldata.email,order.no\_order

FROM tabeldata

FULL JOIN order

ON tabeldata.id=order.id

ORDER BY tabeldata.nama

1. SQL Selft Join

SQL Selft Join adalah join biasa tetapi tabel bergabung dengan dirinya sendiri.

Syntax :

SELECT column\_name(s)

FROM tabel TI, table T2

WHERE condition;

Contoh :

SELECT A CustomerName AS

CustomerName1, B.CustomerName AS

CustomerName2, A,city

FROM Customer A, customerID<> B. CustomerID

AND A City = b.City

ORDER BY A.City;

1. SQL union

Digunakan untuk menggabungkan hasil dari 2 atau lebih perintah Select. Syntax dasar :

SELECT column\_name(s)FROM table\_name1

UNION column\_name(s) FROM table\_name2

Atau

SELECT column\_name(s) FROM table\_name1

UNION ALL

SELECT column\_name(s) FROM table\_name2

Contoh :

SELECT nama FROM mhs\_kampus1

UNION

SELECT nama FROM mhs\_kampus2

1. SQL GROUP BY

Pernyataan SQL GROUP BY Pernyataan GROUP BY mengelompokkan baris yang memiliki nilai yang sama ke dalam baris ringkasan, seperti "temukan jumlah pelanggan di setiap negara". Pernyataan GROUP BY sering digunakan dengan fungsi agregat (COUNT(), MAX(), MIN(), SUM(), AVG()) untuk mengelompokkan kumpulan hasil menurut satu atau beberapa kolom. KELOMPOK BERDASARKAN Sintaks

Contoh:

SELECT column\_name(s)  
FROM table\_name  
WHERE condition  
GROUP BY column\_name(s)ORDER BY column\_name(s);

1. SQL HAVING

Klausa SQL HAVINGKlausa HAVING ditambahkan ke SQL karena kata kunci WHERE tidak dapat digunakan dengan fungsi agregat. MEMILIKI Sintaks

Contoh:

SELECT column\_name(s)  
FROM table\_name  
WHERE condition  
GROUP BY column\_name(s)HAVING conditionORDER BY column\_name(s);

1. SQL EXISTS

Operator SQL ADA Operator EXISTS digunakan untuk menguji keberadaan record apapun dalam subquery. Operator EXISTS mengembalikan TRUE jika subquery mengembalikan satu atau lebih record.

Contoh:

SELECT column\_name(s)  
FROM table\_name  
WHERE EXISTS  
(SELECT column\_name FROM table\_name WHERE condition);

1. SQL ANY,ALL

SQL APAPUN dan SEMUA Operator Operator ANY dan ALL memungkinkan Anda melakukan perbandingan antara nilai kolom tunggal dan rentang nilai lainnya. Operator APAPUN SQLOperator APAPUN: mengembalikan nilai boolean sebagai hasilnyamengembalikan TRUE jika APAPUN dari nilai subquery memenuhi kondisiAPAPUN berarti kondisi akan benar jika operasi benar untuk salah satu nilai dalam rentang.

Contoh:

### ANY Syntax

SELECT column\_name(s)  
FROM table\_name  
WHERE column\_name operator ANY  
(SELECT column\_name  FROM table\_name  WHERE condition);

1. SQL SELECT INTO

Pernyataan SQL SELECT INTO Pernyataan SELECT INTO menyalin data dari satu tabel ke tabel baru.

Contoh:

Copy all columns into a new table:

SELECT \*  
INTO *newtable* [IN *externaldb*]  
FROM *oldtable*WHERE *condition*;

Copy only some columns into a new table:

SELECT *column1*, *column2*, *column3*, ...  
INTO *newtable* [IN *externaldb*]  
FROM *oldtable*WHERE *condition;*

1. SQL INSERT INTO SELECT

Pernyataan SQL INSERT INTO SELECT Pernyataan INSERT INTO SELECT menyalin data dari satu tabel dan menyisipkannya ke tabel lain. Pernyataan INSERT INTO SELECT mengharuskan tipe data dalam tabel sumber dan target cocok. Catatan: Catatan yang ada di tabel target tidak terpengaruh.

Contoh:

Copy all columns from one table to another table:

INSERT INTO *table2*  
SELECT \* FROM *table1*WHERE *condition*;

Copy only some columns from one table into another table:

INSERT INTO *table2*(*column1*, *column2*, *column3*,...)  
SELECT *column1*, *column2*, *column3*,...  
FROM *table1*  
WHERE *condition*;

1. SQL CASE

Pernyataan KASUS SQL Pernyataan CASE melewati kondisi dan mengembalikan nilai ketika kondisi pertama terpenuhi (seperti pernyataan if-then-else). Jadi, begitu suatu kondisi benar, itu akan berhenti membaca dan mengembalikan hasilnya. Jika tidak ada kondisi yang benar, ini mengembalikan nilai dalam klausa ELSE. Jika tidak ada bagian ELSE dan tidak ada kondisi yang benar, ia mengembalikan NULL.

Contoh:

**ASE Syntax**

CASE  
WHEN condition1 THEN result1  
WHEN condition2 THEN result2  
WHEN conditionN THEN resultN  
ELSE result  
END;

1. SQL NULL Functions

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

Contoh:

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 |

1. SQL Stored Procedures

Apa itu Prosedur Tersimpan?Prosedur tersimpan adalah kode SQL yang disiapkan yang dapat Anda simpan, sehingga kode tersebut dapat digunakan kembali berulang kali.Jadi, jika Anda memiliki kueri SQL yang Anda tulis berulang kali, simpan sebagai prosedur tersimpan, lalu panggil saja untuk menjalankannya.Anda juga dapat meneruskan parameter ke prosedur tersimpan, sehingga prosedur tersimpan dapat bertindak berdasarkan nilai parameter yang diteruskan.

Contoh:

**Stored Procedure Syntax**

CREATE PROCEDURE procedure\_name  
AS  
sql\_statement  
GO;

**Execute a Stored Procedure**

EXEC procedure\_name;

1. SQL Comments

Komentar SQL

Komentar digunakan untuk menjelaskan bagian dari pernyataan SQL, atau untuk mencegah eksekusi pernyataan SQL.

Contoh:

Komentar Satu BarisKomentar satu baris dimulai dengan Teks apa pun antara -- dan akhir baris akan diabaikan (tidak akan dieksekusi).Contoh berikut menggunakan komentar satu baris sebagai penjelasan

**SQL Database**

1. SQL CREATE DB

Pernyataan SQL CREATE DATABASE

Pernyataan CREATE DATABASE digunakan untuk membuat database SQL baru.

1. SQL DROP DB

Pernyataan SQL DROP DATABASE

Pernyataan DROP DATABASE digunakan untuk menjatuhkan database SQL yang ada.

Contoh:

**Syntax**

DROP DATABASE databasename

1. SQL BACKUP DATABASE

Pernyataan SQL CADANGAN DATABASE

Pernyataan CADANGAN DATABASE digunakan di SQL Server untuk membuat cadanganpenuhdari database SQL yang ada.

1. SQL CREATE TABLE

Pernyataan SQL CREATE TABLE

Pernyataan CREATE TABLE digunakan untuk membuat tabel baru dalam database.

Contoh:

CREATE TABLE table\_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
   ....  
);

1. SQL DROP TABLE

Pernyataan SQL DROP TABLE

Pernyataan DROP TABLE digunakan untuk menjatuhkan tabel yang ada dalam database.

1. SQL ALTER TABLE

Pernyataan SQL ALTER TABLEPernyataan ALTER TABLE digunakan untuk menambah, menghapus, atau memodifikasi kolom dalam tabel yang sudah ada.Pernyataan ALTER TABLE juga digunakan untuk menambah dan menghapus berbagai batasan pada tabel yang ada.

1. SQL Constraints

SQL Buat Kendala

Batasan dapat ditentukan ketika tabel dibuat dengan pernyataan CREATE TABLE, atau setelah tabel dibuat dengan pernyataan ALTER TABLE.

Contoh:

CREATE TABLE table\_name (  
    column1 datatype *constraint*,  
    column2 datatype *constraint*,  
    column3 datatype *constraint*,  
    ....  
);

1. SQL NOT NULL

Batasan SQL NOT NULL

Secara default, kolom dapat menampung nilai NULL.

Batasan NOT NULL memaksa kolom untuk NOT menerima nilai NULL.

Ini memaksa bidang untuk selalu berisi nilai, yang berarti Anda tidak dapat menyisipkan catatan baru, atau memperbarui catatan tanpa menambahkan nilai ke bidang ini.

Contoh:

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    Age int  
);

1. SQL UNIQUE

Batasan UNIK SQLBatasan UNIQUE memastikan bahwa semua nilai dalam kolom berbeda.

Kendala UNIQUE dan PRIMARY KEY memberikan jaminan keunikan untuk kolom atau kumpulankolom.Kendala PRIMARY KEY secara otomatis memiliki kendala UNIK.

Namun, Anda dapat memiliki banyak batasan UNIK per tabel, tetapi hanya satu batasan KUNCI UTAMA per tabel.

1. SQL PRIMARY KEY

SQL berikut membuat KUNCI UTAMA pada kolom "ID" saat tabel "Orang" dibuat:

MySQL:

Contoh:

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY (ID)  
);

1. SQL FOREIGN KEY

Batasan KUNCI ASING SQLBatasan FOREIGN KEY digunakan untuk mencegah tindakan yang akan menghancurkan link antar tabel.KUNCI ASING adalah bidang (atau kumpulan bidang) dalam satu tabel, yang merujuk ke KUNCI UTAMA di tabel lain.Tabel dengan kunci asing disebut tabel anak, dan tabel dengan kunci utama disebut tabel referensi atau tabel induk.

Contoh:

CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);

1. SQL CHECK

Batasan SQL CHECKBatasan CHECK digunakan untuk membatasi rentang nilai yang dapat ditempatkan dalam kolom.Jika Anda menentukan batasan CHECK pada kolom, itu hanya akan mengizinkan nilai tertentu untuk kolom ini.Jika Anda mendefinisikan batasan CHECK pada tabel, itu dapat membatasi nilai di kolom tertentu berdasarkan nilai di kolom lain di baris.

Contoh:

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CHECK (Age>=18)  
);

1. SQL DEFAULT

Batasan SQL DEFAULT

Batasan DEFAULT digunakan untuk menetapkan nilai default untuk kolom.

Nilai default akan ditambahkan ke semua catatan baru, jika tidak ada nilai lain yang ditentukan.

Contoh:

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    City varchar(255) DEFAULT 'Sandnes'  
);

1. SQL CREATE INDEX

Pernyataan SQL CREATE INDEX Pernyataan CREATE INDEX digunakan untuk membuat indeks dalam tabel.Indeks digunakan untuk mengambil data dari database lebih cepat daripada sebaliknya. Pengguna tidak dapat melihat indeks, mereka hanya digunakan untuk mempercepat pencarian/kueri.

Contoh:

CREATE INDEX index\_name  
ON table\_name (column1, column2, ...);

CREATE UNIQUE INDEX index\_name  
ON table\_name (column1, column2, ...);

1. SQL AUTO INCREMENT

Bidang PENINGKATAN OTOMATIS Peningkatan otomatis memungkinkan nomor unik dihasilkan secara otomatis ketika catatan baru dimasukkan ke dalam tabel.Seringkali ini adalah bidang kunci utama yang ingin kita buat secara otomatis setiap kali catatan baru dimasukkan.

Contoh:

CREATE TABLE Persons (  
    Personid int NOT NULL AUTO\_INCREMENT,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY (Personid)  
);

1. SQL Working With Dates

Tanggal SQL Bagian tersulit saat bekerja dengan tanggal adalah memastikan bahwa format tanggal yang Anda coba masukkan, cocok dengan format kolom tanggal dalam database. Selama data Anda hanya berisi bagian tanggal, kueri Anda akan berfungsi seperti yang diharapkan. Namun, jika ada porsi waktu yang terlibat, itu menjadi lebih rumit.

1. SQL Views

Pernyataan SQL CREATE VIEW Dalam SQL, tampilan adalah tabel virtual berdasarkan kumpulan hasil dari pernyataan SQL.Tampilan berisi baris dan kolom, seperti tabel nyata. Bidang dalam tampilan adalah bidang dari satu atau beberapa tabel nyata dalam database.Anda dapat menambahkan pernyataan dan fungsi SQL ke tampilan dan menyajikan data seolah-olah data berasal dari satu tabel.Tampilan dibuat dengan pernyataan CREATE VIEW.

Contoh:

CREATE VIEW Syntax

CREATE VIEW view\_name AS  
SELECT column1, column2, ...  
FROM table\_name  
WHERE condition;

1. SQL Injection

Injeksi SQL Injeksi SQL adalah teknik injeksi kode yang dapat menghancurkan database Anda.Injeksi SQL adalah salah satu teknik peretasan web yang paling umum.Injeksi SQL adalah penempatan kode berbahaya dalam pernyataan SQL, melalui input halaman web.

1. SQL Hosting

SQL Hosting Jika Anda ingin situs web Anda dapat menyimpan dan mengambil data dari database, server web Anda harus memiliki akses ke sistem database yang menggunakan bahasa SQL.Jika server web Anda di-host oleh Penyedia Layanan Internet (ISP), Anda harus mencari paket hosting SQL.Database hosting SQL yang paling umum adalah MS SQL Server, Oracle, MySQL, dan MS Access.

Contoh:

**MS SQL Server**

Microsoft's SQL Server is a popular database software for database-driven web sites with high traffic.

SQL Server is a very powerful, robust and full featured SQL database system.

**Oracle**

Oracle is also a popular database software for database-driven web sites with high traffic.

Oracle is a very powerful, robust and full featured SQL database system.

**MySQL**

MySQL is also a popular database software for web sites.

MySQL is a very powerful, robust and full featured SQL database system.

MySQL is an inexpensive alternative to the expensive Microsoft and Oracle solutions.

**MS Access**

When a web site requires only a simple database, Microsoft Access can be a solution.

MS Access is not well suited for very high-traffic, and not as powerful as MySQL, SQL Server, or Oracle.

1. SQL Data Types

Tipe Data SQL Setiap kolom dalam tabel database harus memiliki nama dan tipe data. Pengembang SQL harus memutuskan jenis data apa yang akan disimpan di dalam setiap kolom saat membuat tabel. Tipe data adalah pedoman bagi SQL untuk memahami tipe data apa yang diharapkan di dalam setiap kolom, dan juga mengidentifikasi bagaimana SQL akan berinteraksi dengan data yang disimpan.

Contoh: