

Nama : Nasrawati
Nim : 2000250502048
Jurusan : Teknik informatika
Kelas : TI.B
Semester : 3

TUGAS SISTEM BASIS DATA

Penjelasan serta contoh tentang SQL

1. SQL tutorial

SQL adalah bahasa standar untuk mengakses dan memanipulasi database. Dalam tutorial ini anda akan belajar bagaimana menggunakan SQL untuk mengakses dan memanipulasi data di MySQL, SQL Server, MS Access, Oracle, sybase, DB2, dan sistem database lainnya.

2. SQL Home

SQL Home adalah platform sensor sql yang memungkinkan anda menggunakan nilai dari database SQL yang didukung oleh perusahaan sql alchemy, untuk mengisi status sensor dan atribut. Ini dapat digunakan untuk menyajikan statistik tentang sensor home assistant jika digunakan dengan database integrasi perekam. Ini juga dapat digunakan dengan sumber data eksternal.

3. SQL Intro

SQL Intro digunakan untuk membuat, menghapus, mengubah database dan objek database dalam sistem manajemen database untuk menyimpan, mengambil, memperbaharui data dalam database.

4. SQL Syntax

Syntax adalah aturan menulis 'kalimat' agar mampu dimengerti dengan benar oleh bahasa pemrograman. Maksud adanya sintaks atau aturan ini adalah agar kode atau script yang ditulis dapat dibenarkan dan dimengerti oleh komputer.

5. SQL Select

SQL Select adalah query yang bisa digunakan untuk menampilkan maupun mengambil data pada sebuah tabel, data yang diambil tidak hanya terbatas pada satu jenis.

Syntax dasar:

```
SELECT column_name(s)
```

```
FROM table_name
```

Atau

```
SELECT * FROM table_name
```

Contoh :

```
SELECT nama, email
```

```
FROM tabeldata
```

6. SQL Select Distinct

SQL Select Distinct adalah sql yang digunakan untuk memilih data yang berbeda (menghilangkan duplikasi) dari sebuah table database.

Syntax dasar:

```
SELECT DISTINCT column_name(s)
```

```
FROM table_name
```

Contoh :

```
SELECT DISTINCT kota
```

```
FROM tabeldata
```

7. SQL WhereSQL And, Or, Not

SQL WhereSQL And, Or, Not merupakan perintah dasar SQL yang biasanya di kombinasikan dengan perintah WHERE. Ketiganya digunakan untuk memfilter record berdasarkan suatu kondisi, operator AND akan menampilkan record apabila semua kondisi bernilai TRUE, operator OR akan menampilkan record apabila salah satu kondisi bernilai TRUE, sedangkan operator NOT akan menampilkan record apabila semua kondisi bernilai FALSE.

Syntax AND :

```
SELECT kolom1, kolom2,....
```

```
FROM nama_tabel
```

```
WHERE kondisi1 AND kondisi2 AND kondisi3...;
```

Contoh AND :

```
SELECT nis, nama FROM siswa WHERE
```

```
alamat='Jakarta' AND
```

```
tahun_lahir='2000'
```

Syntax OR:

```
SELECT kolom1, kolom2,...
```

```
FROM nama_table
```

```
WHERE kondisi1 OR kondisi2 OR kondisi3
```

Contoh OR:

```
SELECT nis, nama FROM siswa WHERE  
Alamat='jakarta' OR alamat='Bandung'
```

Syntax NOT:

```
SELECT kolom1, kolom2,...  
FROM nama_table WHERE NOT kondisi
```

Contoh NOT :

```
SELECT nis, nama  
FROM siswa WHERE  
NOT alamat= 'Jakarta'
```

8. SQL Order By

SQL order by di gunakan untuk mengurutkan data berdasarkan kolom (filed) tertentu.

Secara default, urutan tersusun secara ascending (urut kecil ke besar). Anda dapat mengubahnya menjadi discending (urut besar ke kecil) dengan menambah perintah DESC.

Syntax dasar:

```
SELECT column_name(s)  
FROM table_name  
ORDER BY column_name(s) ASCIDESSC
```

Contoh :

```
SELECT*FROM tabeldata ORDER BY nama
```

9. SQL Insert Int

Dalam SQL, perintah INSERT INTO merupakan perintah dasar SQL bagian dari perintah untuk DML.

Syntax dasar :

```
INSERT INTO nama_tabel VALUES  
(nilai1, nilai2, nilai3,.....)
```

Contoh :

```
INSERT INTO siswa VALUES ('12345', Abdul');
```

10. SQL Null Values

SQL Null Values 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.

Sintaks IS NULL

```
SELECT column_names
```

```
FROM table_name
```

```
WHERE column_name IS NOT NULL
```

Operator IS NULL digunakan untuk menguji nilai kosong (nilai null). Sql berikut mencantumkan semua pelanggan dengan null di bidang "ALAMAT":

Contoh :

```
SELECT customerName, contactName, Address
```

```
FROM customers
```

```
WHERE Address IS NUL;
```

11. SQL Update SQL Delete

SQL Update digunakan untuk mengubah/menperbarui data di tabel database.

Syntax dasar:

```
UPDATE table_name
```

```
SET column1=value,column2=value,...
```

```
WHERE some_column=some_value
```

Contoh :

```
UPDATE tabeldata SET
```

```
Email='redaksi@kuliahkomputer.Com',
```

```
Kota='kediri'
```

```
WHERE id = 1
```

SQL Delete digunakan untuk menghapus data di tabel database. Tambahkan perintah Where untuk memfilter data-data tertentu yang akan di hapus. Jika tanpa perintah Where, maka seluruh data dalam tabel akan terhapus

Syntax dasar :

```
DELETE FROM table_name
```

```
WHERE some_column=some_value
```

Contoh

```
DELETE FROM tabeldata WHERE id =1
```

12. 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;
```

13. 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
```

14. 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
```

15. 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"

16. SQL Wildcards

SQL Wildcards di gunakan untuk menggantikan satu atau lebih karakter 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%':
```

17. 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
```

18. 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 digunakan untuk mencari data yang memiliki nomor id antara 15 dan 45.

19. SQL Aliases

As memungkinkan anda untuk memberi nama ulang/nama alias/nama baru pada sebuah kolom atau tabel tanpa mengubah nama aslinya. Beberapa programmer sering menggunakan fitur AS ini untuk membantu mereka dalam mengenali sebuah tabel atau kolom serta mempermudah 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 Produk;
SELECT p. ID FROM produk AS p WHERE p. ID = 2 ;
```

20. 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 ;
```

21. SQL Inner Join

SQL Inner Join digunakan untuk menghasilkan baris data dengan cara menggabungkan 2 buah tabel atau lebih menggunakan pasangan data yang match pada masing-masing tabel. 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
```

22. 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
```

23. SQL Right Join

SQL Right Join Digunakan untuk menghasilkan baris data dari tabel kanan (nama tabel kedua) yang tidak ada pasangannya 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.email,order.no_order  
FROM tabeldata  
RIGHT JOIN order  
ON tabeldata.id=order.i  
ORDER BY tabeldata.nama
```

24. 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
```

25. SQL Self Join

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

Syntax :

```
SELECT column_name(s)
```

```
FROM tabel T1, 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;
```

26. 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
```

27. SQL Group By

SQL Group By digunakan untuk mengelompokkan data dengan kriteria tertentu.

Syntax dasar :

```
SELECT column_name,aggregate_function(column_name)
```

```
FROM table_name
```

```
WHERE column_name operator value
```

```
GROUP BY column_name
```

Contoh :

```
SELECT nama_customer,SUM(harga) FROM order GROUP BY nama_customer
```

28. SQL Having

SQL Having digunakan untuk memfilter data dengan fungsi tertentu.

Syntax dasar :

```
SELECT column_name,aggregate_function(column_name)
```

```
FROM table_name
```

```
WHERE column_name operator value
```

```
GROUP BY column_name
```

```
HAVING aggregate_function(column_name) operator value
```

Contoh :

```
SELECT nama_customer,SUM(harga) FROM order
```

```
WHERE nama_customer='Rico' OR nama_customer='Bastian'
```

```
GROUP BY nama_customer
```

```
HAVING SUM (harga)>125000
```

29. SQL Exists

SQL Exists merupakan sql yang digunakan untuk mengecek kembali ketersediaan sebuah data dengan menulis sebuah subquery saja. Jika data ditemukan, maka hasilnya akan di tampilkan dalam statement yang sedang dijalankan. Anda juga bisa mengombinasikan filter ini dengan filter SQL lain seperti INSERT,SELECT,DELETE DAN UPDATE.

Syntax :

```
SELECT nama_kolom1, nama_kolom2
```

```
FROM nama_tabel
```

```
WHERE EXISTS ( SELECT nama_kolom
```

```
FROM nama_tabel WHERE kondisi AND komdisi);
```

Contoh :

```
SELECT Produk ID, NamaProduk
```

```
FROM Produks
```

```
WHERE EXISTS ( SELECT Produk_id
```

```
FROM Orders WHERE Orders. Produk_id = produkID AND Produks. Kategori = "kursi" );
```

30. SQL Any All

SQL Any All memungkinkan anda melakukan perbandingan antara nilai kolom tunggal dan rentang nilai lainnya.

Syntax :

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ANY
( SELECT column_name
FROM table_name
WHERE condition) ;
```

Contoh :

```
SELECT ProductName
FROM Products
WHERE ProductID = ANY
(SELECT ProductName
FROM ordedetsils
WHERE Quantity= 10);
```

31. SQL Select Into

SQL Select Into menyalin data dari satu tabel ke tabel baru.

Syntax :

```
SELECT *
INTO newtable [IN externaldo]
FROM oldtable
WHERE condtion;
```

Contoh :

```
SELECT * INTO customersBackup2017
FROM Customers;
```

32. SQL Insert Into Select

SQL Insert Into Select digunakan untuk menambahkan data bari di tabel database.

Syntax dasar:

```
INSER INTO table_name
VALUES (value1, value2, value3.....)
Atau
```

```
INSERT INTO table_name (column1, column2, column3, ....)
```

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

Contoh :

```
INSERT INTO tabledata VALUES
```

```
(1, 'redaksi', 'redaksi@kuliahkomputer.com',
```

```
Kediri')
```

33. SQL Case

SQL CASE melewati kondisi dan mengembalikan nilai ketika kondisi pertama terpenuhi seperti pernyataan if-the-else jadi begitu suatu kondisi benar itu akan berhenti membaca dan mengembalikan hasilnya. Jika tidak ada kondisi yang benar, ia mengembalikan nilai dalam klausa ELSE.

Syntax :

```
CASE
```

```
    WHEN conditionn1 THEN result1
```

```
    WHEN conditionn2 THEN result2
```

```
    WHEN conditionn3 THEN result3
```

```
END;
```

Contoh :

```
SELECT Order ID, Quantity,
```

```
CASE
```

```
    WHEN Quantity > 30 THEN ' The  
quantity is greater than 30'
```

```
    WHEN Quantity = 30 THEN 'The  
quantity is 30'
```

```
    ELSE ' The quantity is under 30'
```

```
END AS QuantityText
```

```
FROM Orderdetails;
```

34. SQL Null Functions

SQL Null Functions adalah tempat penampung dalam database saat kita memiliki data yang hilang, atau data yang di perlukan tidak tersedia. Nilai null bukan merupakan bagian dari tipe data tertentu ini adalah tipe data yang fleksibel dan dapat di letakkan pada kolom tipe data apapun baik itu tipe data string, int, blob dan CLOB.

Contoh :

```
SELECT col1,col2, IFNULL( col3, ) FROM tableName ;
```

35. SQL Stored Procedures

SQL Stored Procedures adalah kode SQL yang disiapkan yang dapat anda simpan, sehingga kode tersebut dapat digunakan kembali berulang kali. Jadi jika anda memiliki query 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.

Syntax :

```
CREATE PROCEDURE procedure_name
```

```
AS
```

```
Sql_statement
```

```
GO;
```

Contoh :

```
CREATE PROCEDURE selectAllcustomers
```

```
AS
```

```
SELECT * FROM Customers
```

```
GO;
```

36. SQL Comments

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

Contoh :

```
--select all:
```

```
SELECT * FROM customers ;
```

37. SQL Operators

SQL Operators berisikan tentang

SQL Arithmetic Operators : operators = +(add), -(subtract), *(multiply),/(divide),%(modulo) contoh Try it. SQL Bitwise Operators : operator &(Bitwise) , SQL Comparison Operators, SQL compound Operators, dan SQL Logical Operators.

Contoh :

Try it

38. SQL Database

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Database adalah representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

39. SQL Create DB

SQL Create DB adalah perintah yang digunakan untuk membuat sebuah database baru, baik dalam bentuk tabel baru atau kolom baru.

40. SQL Drop DB

SQL Drop DB adalah query drop yang digunakan untuk menghapus elemen database apapun yang anda inginkan, mulai dari database, tabel sampai index.

41. SQL Backup DB

Backup DB adalah proses membuat data cadangan dengan cara menyalin atau membuat arsip data komputer sehingga data tersebut dapat digunakan kembali apabila terjadi kerusakan atau kehilangan data.

42. SQL Create Table

SQL Create Table adalah Create yang digunakan untuk membuat tabel data baru dalam Syntax dasar:

```
CREATE TABLE
(
Column_name1 table_nama data_Type
Column_name2 table_nama data_Type
Column_name3 table_nama data_Type
)
```

Contoh :

```
CREATE TABLE table data
```

```
(id int, nama varchar (255), email
```

```
Varchar (50),
```

```
Kota varchar(255))
```

43. SQL Drop Table

SQL Drop Table digunakan untuk menghapus tabel beserta seluruh datanya.

Syntax dasar :

```
DROP TABLE table_name
```

Contoh :

```
DROP TABLE mhs
```

44. SQL Alter Table

SQL Alter Table Digunakan untuk menambah, menghapus, atau mengubah kolom (field) pada tabel yang sudah ada.

Syntax dasar :

```
ALTER TABLE table_name
```

```
ADD column-name datatype
```

Or

```
ALTER TABLE table_name
```

```
DROP COLUMN column-name
```

Contoh :

```
ALTER TABLE Persons
```

```
ADD DateOfBirth date
```

Syntax untuk menghapus kolom :

```
ALTER TABLE table_name
```

```
DROP COLUMN column_name
```

45. SQL Constrains

SQL Constrains digunakan untuk menentukan aturan untuk data dalam tabel. Constraint digunakan untuk membatasi jenis data yang dapat masuk ke dalam table. Ini memastikan keakuratan dan kendala data dalam tabel. Jika ada pelanggaran antara batasan dan tindakan data, tindakan dibatalkan. Batasan dapat berupa level kolom atau level tabel berlaku untuk seluruh tabel.

Contoh :

```
CREATE TABLE table_name (  
    Column1 datatype constraint,  
    Column2 datatype constraint,  
    Column3 datatype constraint,  
    ...  
);
```

46. SQL Not Null

SQL berikut memastikan bahwa kolom "ID", LastName ", dan "FirstName" TIDAK akan menerima nilai NULL saat tabel "orang".

Contoh :

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

47. SQL Unique

SQL Unique memastikan bahwa semua nilai dalam kolom berbeda. Kendala unique dan primary key memberikan jaminan keunikan untuk kolom atau kumpulan kolom. 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.

Contoh :

```
CREATE TABLE Persons (  
    ID int Not NULL UNIQUE,  
    LastName varchar (255) NOT NULL  
    firstName varchar (255),  
    Age int  
);
```

48. SQL Primary Key

SQL Primary Key secara unik mengidentifikasi setiap record dalam sebuah tabel. Kunci utama harus berisi nilai UNIK, dan tidak boleh berisi nilai NULL. Sebuah tabel hanya dapat memiliki satu kunci utama ini dapat terdiri dari satu atau beberapa kolom (bidang).

Contoh :

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

PRIMARY KEY (ID)

);

49. SQL Foreign Key

SQL Foreign Key adalah sebuah sql yang memiliki batasan Foreign key mencegah data yang tidak valid di masukkan ke dalam kolom kunci asing, karena harus menjadi salah satu nilai yang terdapat dala tabel induk.

Contoh :

```
CREATE TABLE Orders (  
    orderID int NOT NULL,  
    order Number int NOT NULL,  
    PRIMARY KEY (orderID),  
    FOREIGN KEY (PersonID)  
REFERENCES Persons (PersonID)  
);
```

50. SQL Check

SQL 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 kolom di baris.

Contoh :

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

51. SQL Default

SQL Default digunakan untuk menetapkan nilai default untuk kolom. Nilai default akan ditambahkan ke semua catatan baru, jika tidak ada nilai yan di tentukan.

Contoh :

```
CREATE TABLE Persons (  

```

```

ID int NOT NULL,
LastName varchar (255) NOT NULL,
FirstName varchar (255),
Age int,
City varchar (255) DEFAULT
'sandnes'
);

```

52. SQL Index

SQL Index merupakan objek struktur data tersendiri yang tidak bergantung kepada struktur table. Setiap index terdiri dari nilai kolom dan petunjuk ke baris yang berisi nilai tersebut.

Contoh :

```

SELECT COUNT (*) FROM
DATA_PASIEN

```

53. SQL Auto Increment

SQL Auto Increment peningkatan otomatis memungkinkan no unit dihasilkan secara otomatis ketika catatan baru dimasukkan kedalam 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 Varcher (255) ,
    Age int ,
    PRIMARY KEY ( personid)

```

54. SQL Dates

SQL Dates merupakan bagian tersulit saat bekerja dengan tanggal adalah memastikan bahwa format yang anda masukkan cocok dengan format kolom tanggal databse.

Contoh :

DATE – format YYYY-MM-DD

DATETIME – Format YYYY-MM-DD-HH-MI-SS

55. SQL Views

SQL Views merupakan tampilan dalam SQL adalah jenis tabel virtual. Tampilan juga memiliki baris dan kolom seperti pada tabel nyata di database.

Contoh :

```
CREATE VIEW DetailsView AS  
SELECT NAME, ADDRESS  
FROM StudentDetails  
WHERE S_ID<5;
```

56. SQL Injection

SQL Injection adalah tekni 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.

Contoh :

```
txUserID =  
getRequestString ("user id");  
txtSQL="SELECT * FROM Users WHERE  
UserId="+txUserid;
```

57. SQL Hosting

SQL Hosting adalah database hosting SQL yang paling umum adalah MS SQL Server, Oracle, MySQL dan MS Access. 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 dihosting oleh Penyedia Layanan Internet (ISP), Anda harus mencari paket hosting SQL. Database SQL hosting yang paling umum adalah MS SQL Server, Oracle, MySQL, dan MS Access.

58. SQL Data Types

Tipe data SQL dapat dibagi secara luas ke dalam kategori. Tipe data digunakan untuk mendefinisikan suatu field atau kolom. Setiap kolom yang dibuat harus didefinisikan terlebih dahulu. 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.

59. SQL References

SQL References adalah nama database. di antara nama database dan nama objek terdapat skema untuk tampilan katalog/DMV, anda tidak dapat mengabaikannya, tetapi jika anda menggunakan skema default (biasanya dbo), anda dapat mengabaikan referensi eksplisit

60. SQL Keywords

SQL keywords terdiri dari add (menambahkan batasan tabel dibuat), alter (menambahkan, menghapus, atau memodifikasi kolom dalam tabel atau mengubah tipe data kolom dalam tabel).

61. MySQL Functions

SQL Server merupakan sistem manajemen basis data relasional yang dirancang untuk aplikasi dengan arsitektur client server. Microsoft SQL merupakan salah satu produk unggulan dari Microsoft mengenai manajemen dari basis data alias database, dimana software ini menggunakan Transact SQL sebagai kueri utamanya. Fitur yang terdapat pada Microsoft SQL Server yaitu memiliki kemampuan untuk membuat basis data clustering dan mirroring.

62. SQL Server Functions

SQL Server merupakan sistem manajemen basis data relasional yang dirancang untuk aplikasi dengan arsitektur client server. Microsoft SQL merupakan salah satu produk unggulan dari Microsoft mengenai manajemen dari basis data alias database, dimana software ini menggunakan Transact SQL sebagai kueri utamanya. Fitur yang terdapat pada Microsoft SQL Server yaitu memiliki kemampuan untuk membuat basis data clustering dan mirroring.

63. MS Access Functions

Microsoft Access Functions adalah aplikasi yang berguna untuk membuat, mengolah, dan mengelola basis data atau lebih dikenal dengan database. Fungsi/kegunaan utama dari Ms. Access yaitu untuk menangani proses manipulasi data dan pembuatan sebuah system. Hadirnya Microsoft Access dapat dimanfaatkan sebagai media untuk membuat aplikasi web dasar

64. SQL Quick Ref

SQL Quick Ref adalah sebuah database yang memiliki sql yang berada di dalamnya termasuk sql hosting dan sql summary.

