

TUGAS II

MATA KULIAH : SISTEM BASIS DATA

OLEH DOSEN : MUSLIADI KH, S.KOM



DISUSUN OLEH :

NAMA : FITRIAH

NIM : 200250502022

UNIVERSIAS TOMAKAKA MAMUJU

FAKULTAS ILMU KOMPUTER

TEKNIK INFORMATIKA

TENTANG SQL

SQL (Structured Query Language) adalah bahasa pemrograman khusus yang digunakan untuk manajemen data dalam RDBMS. SQL biasanya berupa perintah sederhana yang berisi instruksi-instruksi untuk manipulasi data. Perintah SQL ini sering juga disingkat dengan sebutan '**query**'. SQL merupakan bahasa standar untuk menyimpan, memanipulasi, dan mengambil data dalam database.

KUMPULAN) PERINTAH-PERINTAH DASAR MYSQL

1. SQL Create Database

Digunakan untuk membuat database baru.

Syntax dasar: `CREATE DATABASE database_nama`

Contoh: `CREATE DATABASE databaseku`

2. SQL Create Table

Digunakan untuk membuat tabel data baru dalam sebuah database.

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 buktamu (Id int, Nama varchar (255), Email
varchar(50), Kota varchar(255)`

3. SQL Select

Digunakan untuk memilih data dari table database.

Syntax dasar: `SELECT column_name(s) FROM table_name Atau SELECT *
FROM table_name`

Contoh 1: `SELECT nama, email FROM buku tamu`

Contoh 2: `SELECT * FROM buku tamu`

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 Distinct

Digunakan untuk memilih data-data yang berbeda (menghilangkan duplikasi) dari sebuah table database.

Syntax dasar: `SELECT DISTINCT column_name(s) FROM table_name`

Contoh: `SELECT DISTINCT kota FROM buku tamu`

6. SQL Where

Digunakan untuk memfilter data pada perintah Select

Syntax dasar: `SELECT column name(s) FROM table_name WHERE column_name operator value`

Contoh: `SELECT * FROM buku tamu WHERE kota='PAMULANG'`

7. SQL and, or, not

AND, OR dan NOT merupakan perintah dasar SQL yang biasanya di kombinasikan dengan perintah WHERE.

Operator AND dan OR digunakan untuk memfilter record berdasarkan lebih dari satu kondisi:

- Operator AND menampilkan record jika semua kondisi yang dipisahkan oleh AND adalah TRUE.
- Operator OR menampilkan record jika salah satu kondisi yang dipisahkan oleh OR adalah TRUE.
- Operator NOT menampilkan record jika kondisi TIDAK BENAR.

AND Syntax

`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;
```

8. SQL Order By

Digunakan untuk mengurutkan data berdasarkan kolom (field) tertentu. Secara default, urutan tersusun secara ascending (urut kecil ke besar). Anda dapat mengubahnya menjadi descending (urut besar ke kecil) dengan menambahkan perintah DESC.

Syntax dasar: `SELECT column_name(s) FROM table_name ORDER BY column_name(s) ASC|DESC`

Contoh 1: `SELECT * FROM bukutamu ORDER BY nama`

Contoh 2: `SELECT * FROM bukutamu ORDER BY id DESC`

9. 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 1: `SELECT * FROM bukutamu WHERE nama LIKE 'a%'`

Keterangan : Contoh di atas digunakan untuk pencarian berdasarkan kolom nama yang berhuruf depan "a".

Contoh 2: `SELECT * FROM bukutamu WHERE nama LIKE '%a'`

Keterangan : Contoh di atas digunakan untuk pencarian berdasarkan kolom nama yang berhuruf belakang "a".

10. SQL In

Digunakan untuk pencarian data menggunakan lebih dari satu filter pada perintah Where.

Syntax dasar : `SELECT column_name(s) FROM table_name WHERE column_name IN (value1,value2, . . .)`

Contoh: `SELECT * FROM bukutamu WHERE kota IN ('Tangerang','Depok')`

11. 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 bukutamu WHERE id BETWEEN 5 and 15`

Keterangan : Contoh di atas digunakan untuk mencari data yang memiliki nomor id antara 5 dan 15.

12. SQL Insert Into

Digunakan untuk menambahkan data baru di tabel database.

Syntax dasar : `INSERT INTO table_name VALUES (value1, value2, value3, . . .)`

Atau `INSERT INTO table_name (column1, column2, column3, . . .) VALUES (value1, value2, value3, . . .)`

Contoh 1 : `INSERT INTO buku tamu VALUES (1, 'Arlicious', 'arly@arlicious.com', 'Tangerang')`

Contoh 2: `INSERT INTO buku tamu (id , nama , email , kota) VALUES (1, 'Arlicious', 'arly@arlicious.com', 'Tangerang')`

13. SQL Update

Digunakan untuk mengubah/memperbarui data di tabel database.

Syntax dasar: `UPDATE table_name SET column1=value,column2=value, . . . WHERE some_column=some_value`

Contoh : `UPDATE bukutamu SET email='arly@arlicious.com', kota='Tangerang' WHERE`

14. Delete

Digunakan untuk menghapus data di table database. Tambahkan perintah Where untuk memfilter data-data tertentu yang akan dihapus. 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 bukutamu WHERE id=1`

15. 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 perintah join yang sering digunakan.

Syntax dasar : `SELECT column_name(s) FROM table_name1 INNER JOIN table_name2 ON table_name1.column_name=table_name2 column-name`

contoh :

```
SELECT   bukutamu.nama,bukutamu.email,order.no_order   FROM   bukutamu
INNER JOIN order ON bukutamu.id=order.id ORDER BY bukutamu.nama
```

16. SQL Left Join

Digunakan untuk menghasilkan baris data dari tabel kiri (nama tabel pertama) yang tidak 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 bukutamu.nama,bukutamu.email,order.no_order FROM bukutamu LEFT
JOIN order ON bukutamu.id=order.id ORDER BY bukutamu.nama
```

17. 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 bukutamu.nama,bukutamu.emailmorder.no_order FROM bukutamu
RIGHT JOIN order ON bukutamu.id=order.i ORDER BY bukutamu.nama`

18. 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 bukutamu.nama,bukutamu.email,order.no_order FROM bukutamu
FULL JOIN order ON bukutamu.id=order.id ORDER BY bukutamu.nama`

19. 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`

20. SQL Alter Table

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

Syntax untuk menambah kolom : `ALTAR TABLE table_name ADD column_name
datatyoe`

Contoh : `ALTER TABLE Persons ADD DateOfBirth date`
Syntax untuk menghapus kolom : `ALTER TABLE table_name DROP COLUMN column_name`

Contoh : `ALTER TABLE Persons DROP COLUMN DateOfBirth`

Syntax untuk mengubah kolom : `ALTER TABLE table_name ALTER TABLE
clumn_name datatype`

Contoh : `ALTER TABLE Persons ALTER COLUMN DateOfBirth year`

21. SQL Now ()

Digunakan untuk mendapatkan informasi waktu (tanggal dan jam saat ini.)

Syntax dasar : Now()

Contoh : SELECT NOW()

22. SQL Curdate

Digunakan untuk mendapatkan informasi tanggal saat ini.

Syntax dasar : Curdate()

Contoh : SELECT CURDATE()

23. SQL Curtime()

Digunakan untuk mendapatkan informasi jam saat ini.

Syntax dasar : Curtime()

Contoh : SELECT CURTIME()

24. SQL Extract

Digunakan untuk mendapatkan informasi bagian-bagian dari data waktu tertentu, seperti tahun, bulan, hari, jam, menit, dan detik tertentu.

Syntax dasar : Extract(unit FROM date)

Keterangan : Parameter unit dapat berupa :

- MICROSECOND
- SECOND
- MINUTE
- HOUR
- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

- SECOND_MICROSECOND
- MINUTE_SECOND
- HOUR_MICROSECOND
- HOUR_SECOND
- HOUR_MINUTE
- DAY_MICROSECOND
- DAY_SECOND
- DAY_MINUTE
- DAY_HOUR
- YEAR_MONTH

Contoh :

```
SELECT EXTRACT (YEAR FROM tglorder( AS Th_Order, EXTRACT (MONTH
FROM tglorder) AS Bulan_Order,EXTRACT (DAY FROM tglorder AS Hari_Order,
FROM order
WHERE
```

25. SQL Date_Add() dan Date_Sub()

Fungsi Date_Add() digunakan untuk menambahkan interval waktu tertentu pada sebuah tanggal, sedangkan fungsi Date_Sub() digunakan untuk pengurangan sebuah tanggal dengan interval tertentu.

Syntax dasar :

DATE_ADD (date,INTERVAL expr type)

DATE_SUB (date,INTERVAL expr type)

Keterangan :

Tipe data parameter INTERVAL dapat berupa :

- MICROSECOND
- SECOND
- MINUTE
- HOUR
- DAY

- WEEK
- MONTH
- QUARTER
- YEAR
- SECOND_MICROSECOND
- MINUTE_MICROSECOND
- MINUTE_SECOND
- HOUR_MICROSECOND
- HOUR_SECOND
- HOUR_MINUTE
- DAY_MICROSECOND
- DAY_SECOND
- DAY_MINUTE
- DAY_HOUR
- YEAR_MONTH

Contoh 1:

```
SELECT id,DATE_ADD (tglorder,INTERVAL 30 DAY)
AS Waktu_pembayaran
FROM order
```

Contoh 2:

```
SELECT id,DATE_SUB(tglorder,INTERVAL 5 DAY)
AS Pengurangan_Waktu
FROM order
```

26. SQL DateDiff()

Digunakan untuk mendapatkan informasi waktu di antara 2 buah tanggal.

Syntax dasar :

```
DATEIFF(date1,date2)
```

Contoh :

```
SELECT DATEIFF('2010-06-30','2010-06-29')
AS Selisih_waktu
```

27. SQL Date_Format()

Digunakan untuk menampilkan informasi jam dan tanggal dengan format tertentu.

Syntax dasar :

DATE_FORMAT(date,format)

Keterangan :

Parameter format dapat berupa :

- %a, nama hari yang disingkat
- %b, nama bulan yang disingkat
- %c, bulan (numerik)
- %D hari dalam sebulan dengan format English
- %d, hari dalam sebulan (numerik 00-31)
- %e, hari dalam sebulan (numerik 0-31)
- %f, micro detik
- %H, jam (00-23)
- %h, jam (01-12)
- %I, jam (01-12)
- %i, menit (00-59)
- %j, hari dalam setahun (001-366)
- %k, jam (0-23)
- %l, jam (1-12)
- %M, nama bulan
- %m, bulan (numerik 00-12)
- %p, AM atau PM
- %r, waktu jam dalam format 12 jam (hh:mm:ss AM or PM)
- %S, detik (00-59)
- %s, detik (00-59)
- %T, waktu jam dalam format 24 jam (hh:mm:ss)
- %U, minggu (00-53) dimana Sunday sebagai hari pertama dalam seminggu
- %u, minggu (00-53) dimana Monday sebagai hari pertama dalam seminggu

- %W, nama hari kerja
- %w, hari dalam seminggu (0=Sunday, 6=Saturday)
- %X, tahun dalam seminggu dimana Sunday sebagai hari pertama dalam seminggu (4 digits) digunakan dengan %V
- %x, tahun dalam seminggu di mana Monday sebagai hari pertama dalam seminggu (4 digits) digunakan dengan %v
- %Y, tahun 4 digit
- %y, tahun 2 digit

Contoh :

DATA_FORMAT (NOW(), '%b %d %Y %h : %i %p')

DATE_FORMAT (NOW(), '%m-%d-%Y')

DATE_FORMAT (NOW(), '%d %b %Y')

DATE_FORMAT (NOW(), '%d %b %Y %T : %f')

28. Drop Table

Digunakan untuk menghapus tabel beserta seluruh datanya.

Syntax dasar :

DROP TABLE table_name

Contoh :

DROP TABLE mhs

29. SQL Drop Database()

Digunakan untuk menghapus database.

Syntax dasar :

DROP DATABASE database_name

30. 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
```

31. 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 bukutamu
```

32. 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
```

33. 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
```

34. 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
```

35. 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
```

36. 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='Arlicious' OR nama_customer='Dvallen'
```

```
GROUP BY nama_customer
```

```
HAVING SUM (harga)>25000
```

37. SQL Ucase()

Digunakan untuk mengubah huruf pada data tertentu menjadi huruf besar.

Syntax dasar :

```
SELECT UCASE (column_name) FROM table_name
```

Contoh :

```
SELECT UCASE(nama) as Nama FROM bukutamu
```

38. SQL Lcase()

Digunakan untuk mengubah huruf pada data tertentu menjadi huruf kecil.

Syntax dasar :

```
SELECT LCASE (column_name) FROM table_name
```

Contoh :

```
SELECT LCASE(nama) as Nama FROM bukutamu
```

39. SQL Mid()

Digunakan untuk mengambil beberapa karakter dari field teks.

Syntax dasar:

```
SELECT MID(column_name,start[,length]) FROM table_name
```

Contoh:

```
SELECT MID (kota,1,4) as singkatan_kota FROM  
Buku tamu
```

40. SQL Len()

Digunakan untuk mendapatkan informasi jumlah karakter dari field teks.

Syntax dasar:

```
SELECT LEN (column_name) FROM table_name
```

Contoh:

```
SELECT LEN(nama) as panjang_nama  
FROM bukutamu
```

41. SQL Round()

Digunakan untuk pembuatan bilangan pecahan. Syntax dasar:

```
SELECT ROUND (column_name,decimals)FROM table_name
```

Contoh:

```
SELECT no_mhs, ROUND (nilai,0) as nilai_bulat  
FROM tnilai
```

42. SQL NULL VALUES

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

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

- Tidak mungkin mengujinilai NULL dengan operator perbandingan, seperti =, <, atau <>.
- Kita harus menggunakan operator IS NULL dan IS NOT NULL sebagaigantinya.

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;
```

43. SQL wildcards

Karakter wildcard digunakan untuk menggantikan satu atau lebih karakter dalam sebuah string. Karakter wildcard digunakan dengan operator LIKE. Operator LIKE digunakan dalam klausa WHERE untuk mencari pola tertentu dalam kolom.

Contoh

```
SELECT * FROM Customers
WHERE City LIKE 'ber%';
```

44. SQL aliases

Aliases digunakan untuk memberi tabel, atau kolom dalam tabel, nama sementara. Aliases sering digunakan untuk membuat nama kolom lebih mudah dibaca. Aliases dibuat dengan kata kunci AS.

Alias Column Syntax

```
SELECT column_name AS alias_name
FROM table_name;
```

Alias Table Syntax

```
SELECT column_name(s)
FROM table_name AS alias_name;
```


45. SQL JOINS

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

46. SQL EXISTS

EXISTS digunakan untuk menguji keberadaan record apa pun disubquery .EXISTS mengembalikan nilai jika subquery. mengembalikan satu atau lebih record.

EXISTS Syntax

SELECT *column_name(s)*

FROM *table_name*

WHERE EXISTS

(SELECT *column_name* FROM *table_name* WHERE *condition*);

47. SQL ANY, ALL

ANY dan ALL memungkinkan kita melakukan perbandingan antara satu nilai kolom dan rentang nilai lainnya. ANY mengembalikan nilai boolean sebagai hasilnya, mengembalikan TRUE jika salah satu nilai subkueri memenuhi kondisi. ANY berarti bahwa kondisi tersebut akan menjadi benar jika operasi tersebut benar untuk salah satu nilai dalam rentang tersebut.

ANY Syntax

SELECT *column_name(s)*

FROM *table_name*

WHERE *column_name operator ANY*

(SELECT *column_name*

FROM *table_name*

WHERE *condition*);

Operator ALL:

- mengembalikan nilai boolean sebagai hasilnya
- mengembalikan TRUE jika SEMUA nilai subkueri memenuhi kondisi
- digunakan dengan pernyataan SELECT, WHERE dan HAVING

ALL berarti bahwa kondisi akan menjadi benar hanya jika operasi itu benar untuk semua nilai dalam rentang tersebut.

ALL Syntax With SELECT

```
SELECT ALL column_name(s)
FROM table_name
WHERE condition;
```

48. SQL SELECT INTO

SELECT INTO digunakan untuk menyalin data dari satu tabel ke tabel baru.

SELECT INTO Syntax

Salin semua kolom ke dalam tabel baru:

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

Salin hanya beberapa kolom ke dalam tabel baru:

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

Di mana kondisi tabel baru akan dibuat dengan nama kolom dan tipe seperti yang ditentukan dalam tabel lama. Anda dapat membuat nama kolom baru menggunakan klausa AS.

49. SQL insert into select

INSERT INTO SELECT digunakan untuk menyalin data dari satu tabel dan menyisipkannya ke tabel lain. Pernyataan INSERT INTO SELECT mengharuskan tipe data di tabel sumber dan target cocok.

INSERT INTO SELECT Syntax

Salin semua kolom dari satu tabel ke tabel lain:

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

Salin hanya beberapa kolom dari satu tabel ke tabel lain:

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

50. SQL CASE

CASE digunakan untuk melewati kondisi dan mengembalikan nilai saat 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.

CASE Syntax

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

51. SQL STORED PROCEDURES

Stored Procedures adalah kode SQL yang disiapkan yang dapat kita simpan, sehingga kode tersebut dapat digunakan kembali berulang kali.

Jadi jika kita memiliki kueri SQL yang kita tulis berulang kali, simpan sebagai Stored Procedures, lalu panggil saja untuk menjalankannya.

Kita juga dapat meneruskan parameter keprosedur tersimpan, sehingga prosedur tersimpan dapat bertindak berdasarkan nilai parameter yang diteruskan.

Stored Procedure Syntax

```
CREATE PROCEDURE procedure_name
AS
sql_statement
GO;
```

52. SQL comments

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

53. SQL operators

Operator adalah kata atau karakter khusus yang digunakan terutama dalam klausa WHERE pernyataan SQL untuk melakukan operasi, seperti perbandingan dan operasi aritmatika. Operator ini digunakan untuk menentukan kondisi dalam pernyataan SQL dan berfungsi sebagai konjungsi untuk beberapa kondisi dalam sebuah pernyataan.

54. SQL CONSTRAINTS

Constraints adalah sebuah opsi atau atribut yang berfungsi untuk membatasi nilai setiap data yang akan dimasukkan ke dalam suatu kolom di dalam tabel database

Constraints dapat ditentukan saat tabel dibuat dengan pernyataan CREATE TABLE, atau setelah tabel dibuat dengan pernyataan ALTER TABLE.

Syntax

```
CREATE TABLE table_name (
    column1 datatype constraint,
    column2 datatype constraint,
    column3 datatype constraint,
    ....
);
```

55. SQL NOT NULL

NOT NULL merupakan batasan yang digunakan untuk menjamin pengisian record sebuah tabel agar nilai record tersebut harus berisi data. Sebuah kolom jika diberikan constraint NOT NULL, maka kolom tersebut harus berisi nilai untuk recordnya dan tidak boleh di kosongkan. NOT NULL hanya bisa di definisikan dalam sebuah tabel pada level kolom dan tidak bisa pada level tabel. Perbedaan antara NOT NULL dan PRIMARY KEY adalah bahwa setiap kolom yang dijadikan PRIMARY KEY pasti NOT NULL dan tidak berlaku sebaliknya.

Example

SQL berikut memastikan bahwa kolom "ID", "LastName", dan "FirstName" TIDAK akan menerima nilai NULL saat tabel "Persons" dibuat:

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

56. SQL UNIQUE

UNIQUE merupakan sebuah constraint yang akan membatasi pengisian record yang sama ke dalam sebuah kolom jika kolom tersebut diberikan constraint UNIQUE dalam sebuah tabel. Constraint ini hampir sama dengan PRIMARY KEY, yaitu menjamin bahwa setiap nilai record yang ada dalam sebuah kolom UNIQUE tidak boleh ada yang sama (unik satu sama lain).

Untuk memberi nama batasan UNIK, dan untuk menentukan batasan UNIK di beberapa kolom, gunakan sintaks SQL berikut:

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

```
CONSTRAINT UC_Person UNIQUE (ID,LastName)
);
```

57. SQL PRIMARY KEY

Primary Key digunakan untuk menyelesaikan permasalahan dimana ingin membuat beberapa kolom data dengan isi yang berbeda dengan kolom lain, dengan kata lain hanya menginginkan tidak ada data yang sama atau kembar dari total kolom dalam baris data. Primary key berfungsi bertujuan agar data tidak kembar, dengan membuat sebuah kolom yang ditujukan sebagai kunci primer atau *primary key*. Primary key hanya boleh digunakan sekali saja. Primary key sebenarnya diciptakan untuk dijadikan kolom utama sebuah tabel, dengan demikian data pada kolom kunci utama tidak mungkin kembar.

SQL berikut membuat PRIMARY KEY di kolom "ID" ketika tabel "Persons" dibuat:

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

58. SQL FOREIGN KEY

Foreign key disebut juga sebagai constraint Referential Integrity, constraint ini memastikan bahwa data pada suatu tabel didefinisikan pada tabel lainnya dan mengikat kedua tabel tersebut dalam hubungan parent/child atau referenced/dependent.

SQL berikut membuat FOREIGN KEY di kolom "PersonID" ketika tabel "Pesanan" dibuat:

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

59. SQL CHECK

CHECK berfungsi untuk membatasi nilai-nilai yang dapat dimasukkan kedalam suatu kolom di dalam tabel. Sebagai contoh sederhana, kolom 'status_produk' nilainya dapat kita batasi misalnya hanya berupa "ready stock" dan "out stock" saja. Selain nilai tersebut server database akan menolak.

SQL berikut membuat batasan PERIKSA pada kolom "Usia" saat tabel "Orang" dibuat. Batasan PERIKSA memastikan bahwa usia seseorang harus berusia 18 tahun atau lebih:

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

60. SQL DEFAULT

DEFAULT digunakan untuk menetapkan nilai default untuk kolom. Nilai default akan ditambahkan ke semua record baru, jika tidak ada nilai lain yang ditentukan.

SQL berikut menetapkan nilai DEFAULT untuk kolom "Kota" saat tabel "Orang" dibuat:

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

61. SQL INDEX

INDEX berfungsi untuk mempercepat proses pengambilan data, pengurutan dan pencarian data dari suatu tabel. Data pada tabel yang sudah di indekskan diurutkan berdasarkan kolom indeks. Dengan demikian proses pencarian data dapat lebih cepat. Saat data baru sedikit mungkin pengindeksan belum terlalu berguna, namun jika data

sudah mencapai ribuan atau bahkan ratusan ribu barulah terasa perbedaanya jika field tersebut belum di indeks.

Syntax

```
CREATE INDEX index_name  
ON table_name (column1, column2, ...);
```

62. SQL AUTO INCREMENT

AUTO INCREMENT digunakan apa bila ingin menciptakan sebuah kolom yang dapat merekam data. Dengan menggunakan fungsi ini, akan otomatis masuk sendiri tanpa harus dilakukan secara manual. Penambahan ini adalah penambahan otomatis yang sifatnya berurutan dari kecil hingga besar. *Auto increment* hanya dapat dibuat apabila data berupa angka atau integer.

Auto Increment merupakan fitur pada database yang memungkinkan pengguna membuat angka unik yang dihasilkan secara otomatis dan berurut untuk setiap rekor / catatan baru yang dimasukkan / di input kedalam table database tersebut. Dan kebanyakan orang menggunakan angka unik Auto Increment ini sebagai Primary Key / Kunci Utama pada tabel database tersebut.

Pernyataan SQL berikut mendefinisikan kolom "Personid" menjadi auto-increment primary key di tabel " Personid ":

```
CREATE TABLE Persons (  
    Personid int IDENTITY(1,1) PRIMARY KEY,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int  
);
```

63. SQL DATES

Dates atau tanggal merupakan Bagian tersulit saat bekerja dengan tanggal adalah memastikan bahwa format tanggal yang Anda coba sisipkan, cocok dengan format kolom tanggal dalam database. Selama data Anda hanya berisi bagian tanggal, kueri Anda akan berfungsi seperti yang diharapkan. Namun, jika melibatkan porsi waktu, itu menjadi lebih rumit.

SQL Server dilengkapi dengan tipe data berikut untuk menyimpan tanggal atau nilai tanggal / waktu dalam database:

- DATE - format YYYY-MM-DD
- DATETIME - format: YYYY-MM-DD HH:MI:SS
- SMALLDATETIME - format: YYYY-MM-DD HH:MI:SS
- TIMESTAMP - format: a unique number

64. SQL VIEWS

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

CREATE VIEW Syntax

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

65. SQL INJECTION

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

Example

```
txtUserId = getRequestString("UserId");  
txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;
```

66. 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 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.

67. SQL DATA TYPES

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.

SQL REFERENCES

68. SQL KEYWORDS

Dalam SQL, Keywords adalah kata khusus yang digunakan untuk melakukan berbagai operasi dalam database. Ada banyak kata kunci dalam SQL dan karena SQL tidak membedakan huruf besar / kecil, tidak masalah jika kita menggunakan misalnya SELECT atau select.

69. MYSQL FUNCTIONS

MySQL merupakan sebuah perangkat lunak system manajemen basis data SQL (bahasa inggris : data management system) atau DNMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL . Tidak seperti Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan cipta untuk code sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh

sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah : davidaxmark, allanlarsson, dan Michael “monthy widenius.

Kelebihan MySQL antara lain :

- free (bebas di download)
- stabil dan Tangguh
- fleksibel dengan berbagai pemrograman
- Security yang baik
- Dukungan dari banyak komunitas
- kemudahan management database
- mendukung transaksi
- perkembangan software yang cukup cepat.

Kekurangan :

- Tidak cocok untuk menangani data dengan jumlah yang besar, baik untuk menyimpan data maupun untuk memproses data.
- Memiliki keterbatasan kemampuan kinerja pada server ketika data yang disimpan telah melebihi batas maksimal kemampuan daya tampung server karena tidak menerapkan konsep Technology Cluster Server.

70. 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 management 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. Dalam Microsoft SQL Server juga sangat mendukung SQL sebagai bahasa untuk memproses query kedalam. Microsoft SQL server sudah.

SQL sendiri merupakan kependekan dari Structures Query Language, yang merupakan suatu jenis bahasa komputer atau kueri yang digunakan untuk mengakses

data – data yang terdapat dan juga tersimpan di dalam sebuah basis data alias database. Untuk sekarang ini , hampir dari semua DBMS (database management system) sudah mempergunakan bahasa SQL untuk membantu mengimplementasikan dan juga mengkorelasikan dari pengolahan dan juga management dari sebuah databae/ basis data.

Kelebihan :

- Cocok untuk perusahaan dengan skala kecil, menengah, dan besar sehingga mampu untuk mengolah data dengan jumlah yang besar.
- Memiliki kemampuan untuk management user dan tiap user bisa diatur hak akses terhadap suatu database oleh database administrator.
- Untuk diterapkan pada pembangunan suatu program aplikasi, akan mudah dalam melakukan koneksi dengan computer client yang pembangunan aplikasinya menggunakan software yang sama platform dengan MS-SQL, misalnya Microsoft Visual Basic.
- Memiliki tingkat pengamanan / security data yang baik.
- Memiliki kemampuan untuk back-up data, rollback data, dan recovery data.
- Memiliki kemampuan untuk membuat database mirroring dan clustering.

Kekurangan :

- Hanya dapat di implemantasikan pada 1 unit server, jika terdapat tambahan server maka hanya akan berfungsi sebagai pasif / standby server (tidak memiliki kemampuan Technology Cluster Server sepertihalnya pada DMBS Oracle).
- Hanya bisa berjalan pada satu plat form system operasiyaitu Microsoft Windows.
- Merupakan software berlisensi dan berharga mahal untuk perusahaan skala kecil dan menengah.

71. 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. Adapun fungsi lain dari program aplikasi ini antara lain :

- Untuk membuat program aplikasi jumlah peserta didik.
- Untuk membuat laporan keuangan.
- Untuk membuat program aplikasi gaji karyawan.
- Untuk membuat form dan sub form.
- Untuk membuat basis data.
- Untuk membuat program aplikasi penyimpanan buku perpustakaan.
- Untuk membuat program aplikasi daftar hadir.
- Untuk membuat program aplikasi persediaan barang.

Kelebihan dan Kekurangan Microsoft Access

Kelebihan

1. Berbasis file sehingga lebih portabel (tinggal dicopy).
2. Lebih murah
3. Lebih mudah karena ditujukan ke desktop dan tool-tool yang memudahkan serta kemampuannya lebih dibatasi.
4. Mendukung SQL.
5. Mendukung Relational database (terbatas)

Kekurangan

1. Bukan merupakan database server sehingga tidak bisa/tdk cocok untuk keperluan enterprise dan terdistribusi (misalnya Web).
2. Bukan RDBMS murni
3. Tidak mendukung multithreaded
4. Tidak begitu cepat aksesnya (karena berbasis file)
5. Tidak bisa untuk Store procedure
6. Security kalah jauh dibanding SQL Server

Adapun manfaat microsoft access antara lain,

- Menghemat waktu dan tenaga.
- Mempermudah pelaksanaan suatu pekerjaan yang terasabarat dan melelahkan.