

1. Create Database

Digunakan untuk membuat database baru.

Syntax dasar:

```
CREATE DATABASE database_nama
```

2. 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 bukutamu  
(  
  Id int,  
  Nama varchar (255),  
  Email varchar(50),  
  Kota varchar(255)  
)
```

3. 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 bukutamu
```

Contoh 2:

```
SELECT * FROM bukutamu
```

4. 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 bukutamu
```

5. 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 bukutamu  
WHERE kota='PAMULANG'
```

6. 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
```

7. 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".

8. 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')
```

9. 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.

10. 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 bukutamu
VALUES (1,'Arlicious','arly@arlicious.com','Tangerang')
```

Contoh 2:

```
INSERT INTO bukutamu (id,nama,email,kota)
VALUES (1,'Arlicious','arly@arlicious.com','Tangerang')
```

11. 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
```

12. 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
```

13. 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
```

14. Left Join

Digunakan untuk menghasilkan baris data dari tabel kiri (nama tabel pertama) yang tidak ada pasangannya 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
```

15. 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 bukutamu.nama,bukutamu.email,order.no_order
FROM bukutamu
RIGHT JOIN order
ON bukutamu.id=order.id
ORDER BY bukutamu.nama

16. 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
```

17. 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
```

18. Alter Table

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

Syntax untuk menambah kolom :

```
ALTER TABLE table_name
ADD column_name datatype
```

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 COLUMN clumn_name datatype
```

Contoh :

```
ALTER TABLE Persons
ALTER COLUMN DateOfBirth year
```

19. Now ()

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

Syntax dasar :

Now()

Contoh :
SELECT NOW()

20. Curdate

Digunakan untuk mendapatkan informasi tanggal saat ini.

Syntax dasar :

Curdate()

Contoh :

SELECT CURDATE()

21. Curtime()

Digunakan untuk mendapatkan informasi jam saat ini.

Syntax dasar :

Curtime()

Contoh :

SELECT CURTIME()

22. 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
```

23. 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
```

24. 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
```

25. 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')

26. Drop Table

Digunakan untuk menghapus tabel beserta seluruh datanya.

Syntax dasar :

DROP TABLE table_name

Contoh :

DROP TABLE mhs

27. Drop Database()

Digunakan untuk menghapus database.

Syntax dasar :

DROP DATABASE database_name

28. 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_rata2 FROM order

29. 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

30. 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

31. 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

32. 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

33. 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
```

34. 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
```

35. 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
```

36. 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
```

37. 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
```

38. 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
```

39. 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
```