**NAMA :MIFTAHUL JANNAH**

**NIM :200250502032**

**JURUSAN/KELAS :TEKNIK INFORMATIKA/A**

**ANGKATAN :2020**

**MATA KULIAH :SISTEM BASIS DATA**

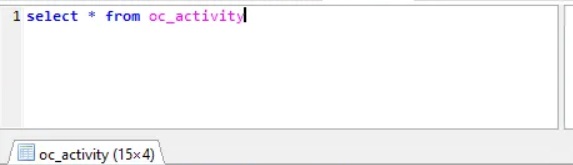
**Tugas:**

**SQL TUTORIAL:**

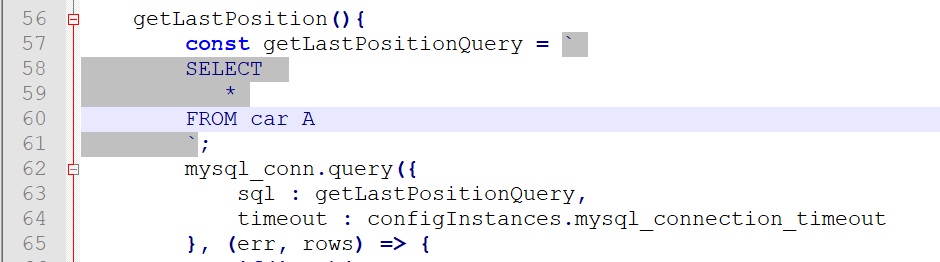
1. **Menjalankannya dengan mengakses langsung ke konsol database**



1. **menjalanannya dengan mengakses melalui databaseclient** seperti (Heidisql, Mysqlworkbench dan SQL Server)



1. **menjalankannya melalui kode pemrograman** contohnya pada java terdapat library bernama jdbc (javadatabaseconnectivity) sehingga aplikasi backend dapat terhubung dengan database.



contoh diatas ini adalah menjalankan sqlquery menggunakan nodejs

**SQL HOME:**

Structured Query Language atau SQL adalah perintah pemrograman yang digunakan untuk mengakses dan mengelola data pada sistem database SQL pertama kali dibahas oleh seorang peneliti bernama Jhonny Oracle pada artikel yang diterbitkan bulan Juni tahun 1970.Pada artikel tersebut Jhonny Oracle membahas pengertian dari bahasa pengelolaan sistem database yang disebut dengan SEQUEL (Structured English Query Language).Namun karena istilah tersebut terlalu panjang dan sulit untuk dieja, istilah SQL pun mulai digunakan sampai sekarang. Setelah melalui proses standarisasi mulai dari tahun tersebut sampai tahun 1986, SQL mengalami perbaikan pertama pada tahun 1989.Berkat kemudahannya, SQL mulai digunakan oleh berbagai RDBMS (Relational Database Management System) dengan versi yang berbeda-beda.

SQL sendiri juga memiliki banyak fungsi yang berbeda dalam pengelolaan database. Berikut ini adalah beberapa kegunaan SQL dan perintah yang digunakan:

1. Membuat database — perintah yang digunakan untuk membuat database adalah create database nama\_database;
2. Mengaktifkan data base — untuk mengaktifkan database, berikut adalah perintah yang digunakan use nama\_database;
3. Menampilkan database — perintah yang digunakan untuk menampilkan database adalah show databases;
4. Menghapus database — untuk menghapus database, perintah yang digunakan adalah drop database nama\_database;
5. Membuat tabel —  untuk membuat tabel, perintah yang digunakan adalah create table nama\_tabel;
6. Menghapus tabel — perintah yang digunakan untuk menghapus tabel adalah drop table nama\_tabel;
7. Melihat struktur tabel — perintah yang digunakan untuk melihat struktur tabel adalah describe nama\_tabel; atau desc nama\_tabel;
8. Menghapus data — perintah query yang digunakan untuk menghapus data adalah delete from nama\_tabel;

**SQL INTRO:**

SQL adalah singkatan dari Structured Query Language.SQL memungkinkan Anda mengakses dan memanipulasi database SQL menjadi standar American national standar institute (ANSI) pada 1986, dan international organization for standardization (ISO) PADA 1987

Hal-hal yang dapat sql lakukalan:

1. SQL dapat mengeksekusi query terhadap database
2. SQL dapat mengambil data dari database
3. SQL dapat menyisipkan catatan dalam database
4. SQL dapat memperbarui catatan dalam database
5. SQL dapat menghapus catatan dari database
6. SQL dapat membuat database baru
7. SQL dapat membuat tabel baru dalam database
8. SQL dapat membuat prosedur tersimpan dalam database
9. SQL dapat membuat tampilan dalam database
10. SQL dapat mengatur izin pada tabel, prosedur, dan tampilan

Meskipun SQL adalah standar ANSI/ISO, ada beberapa versi bahasa SQL yang berbeda.Namun, agar sesuai dengan standar ANSI, semuanya mendukung setidaknya perintah utama (seperti SELECT, UPDATE, DELETE, INSERT, WHERE) dengan cara yang sama.

**SQL SYINTAX:**

Syntax adalah aturan menulis ‘kalimat’ agar mampu dimengerti dengan benar oleh bahasa pemrograman.Aturan syntax ini secara baku harus dipenuhi, karena saat proses compilasi setiap baris script akan di-check dan dipastikan apakah Compiler benar2 mengerti maksud kalimat atau tidak. Jika terdapat syntax yang salah maka compiler akan melaporkan kalo ada bagian kalimat yang dia nggak faham atau istilahnya “error message” dan nggak bakalan meneruskan membikin bytecode-nya.

**SQL SELECT:**

Perintah SELECT merupakan perintah dasar SQL yang di gunakan untuk memilih data dari database.Data yang di kembalikan di simpan dalam tabel yang di sebut result-set. Contoh:

*SELECT kolom1, kolom2, … FROM nama\_tabel;*  
*SELECT \* FROM nama\_tabel;*

**SQLSELECT DISTINCT:**

Perintah SELECT DISTINCT merupakan perintah dasar SQL yang di gunakan untuk mengembalikan hanya nilai yang berbeda dari dalam sebuah tabel, dengan kata lain semua record duplikat (record dengan nilai yang sama) yang terdapat pada tabel akan di anggap sebagai satu record/nilai. Contoh:

*SELECT DISTINCT kolom1, kolom2, … FROM nama\_tabel;*

**SQL WHERE SQL AND, OR NOT:**

Perintah WHERE merupakan perintah dasar SQL yang di gunakan untuk mem-filter hasil SELECT dengan mengekstrak record yang memenuhi persyaratan tertentu. Contoh:

*SELECT kolom1, kolom2, … FROM nama\_tabel WHERE kondisi;*

Operator AND, OR dan NOT merupakan perintah dasar SQL yang biasanya di kombinasikan dengan perintah WHERE. Ketiganya di gunakan untuk mem-filter 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.

*SELECT kolom1, kolom2, … FROM nama\_tabel WHERE kondisi1 AND kondisi2 AND kondisi3;*  
*SELECT kolom1, kolom2, … FROM nama\_tabel WHERE kondisi1 OR kondisi2 OR kondisi3 …;*

**SQL ORDER BY:**

Perintah ORDER BY merupakan perintah dasar SQL yang di gunakan untuk mengurutkan result-set dalam pengurutan ‘ascending’ atau ‘descending’. Secara default perintah ORDER BY menampilkan record dalam pengurutan ‘ascending’ (‘ASC’). Untuk mengurutkan ‘descending’, gunakan kata kunci ‘DESC’. Contoh:

*SELECT kolom1, kolom2, … FROM nama\_tabel ORDER BY column DESC;*

*SELECT nis, nama FROM siswa ORDER BY tahun\_lahir DESC;*

**SQL INSERT INTO:**

Dalam SQL, perintah INSERT INTO merupakan perintah dasar SQL bagian dari perintah untuk DML (Data Manipulation Language) Saya asumsikan Anda telah faham perbedaan DDL, DCL, dan DML. Perintah INSERT INTO dapat di gunakan untuk menambahkan record baru ke dalam tabel.

Contoh:

*INSERT INTO nama\_tabel VALUES (nilai1, nilai2, nilai3, …);*  
*INSERT INTO nama\_tabel (kolom1, kolom2) VALUES (nilai1, nilai2);*

**SQL NULL VALUES:**

Null adalah suatu nilai pada suatu kolom yang berarti tidak mempunyai nilai. Null tidak sama dengan 0. Sebab angka 0 mempunyai nilai yaitu bernilai 0.null juga tidak sama dengan text kosong. null adalah suatu nilai yang tidak bernilai sama sekali. Suatu kolom yang sifatnya optional dapat mempunyai nilai null.Kolom optional artinya datanya bisa diisi bisa tidak.Apabila tidak diisi maka otomatis nilainya null.

Contoh:

SELECT CustomerName, ContactName, Address

FROM Customers

WHERE Address IS NULL;

**SQL UPDATE SQL DELETE:**

Perintah UPDATE merupakan perintah dasar SQL yang di gunakan untuk memperbarui atau mengubah nilai suatu record berdasarkan kriteria tertentu.

Contoh:

*UPDATE nama\_tabel SET kolom1 = nilai1, kolom2 = nilai2, … WHERE kondisi;*

Hampir sama dengan perintah UPDATE, perintah DELETE juga merupakan perintah dasar SQL yang di gunakan untuk menghapus nilai suatu record berdasarkan kriteria tertentu. Contoh:

*DELETE FROM table\_name WHERE condition;*

**SQL SELECT TOP:**

Klausa SELECT TOP digunakan untuk menentukan jumlah record yang akan ditampilkan. Klausa SELECT TOP berguna pada tabel dengan row data yang banyak dengan ribuan record data.Menampilkan sejumlah besar record data dapat berdampak pada kinerja database.

Contoh:

SELECT column\_name(s)

FROM table\_name

WHERE condition

LIMIT number;

**SQL MIN AND MAX:**

Fungsi MIN() merupakan perintah dasar SQL yang di gunakan untuk mendapatkan nilai terkecil dari suatu kolom, Anda dapat menerapkannya pada kolom ‘harga’, ‘nilai’, ‘qty’ atau kolom yang semisal dengan itu, berbeda dengan perintah ORDEY BY, fungsi MIN() hanya menampilkan satu record saja yang memenuhi kriteria yang Anda tentukan.

Contoh:

*SELECT MIN(nama\_kolom) FROM nama\_tabel WHERE kondisi;*

Fungsi MAX() merupakan perintah dasar SQL yang di gunakan untuk mendapatkan nilai terbesar dari suatu kolom, seperti halnya fungsi MIN() Anda dapat menerapkannya pada kolom ‘harga’, ‘nilai’, ‘qty’ atau kolom yang semisal dengan itu.

Contoh:

*SELECT MAX(nama\_kolom) FROM nama\_tabel WHERE kondisi;*

**SQL COUNT, AVG, SUM:**

Fungsi COUNT() merupakan perintah dasar SQL yang di gunakan untuk mendapatkan jumlah hitungan record yang memenuhi suatu kriteria.

Contoh:

*SELECT COUNT(nama\_kolom) FROM nama\_tabel WHERE kondisi;*

Fungsi AVG() merupakan perintah dasar SQL yang di gunakan untuk mendapatkan rata-rata record yang memenuhi suatu kriteria, tentunya nilai pada kolom harus numerik. Contoh:

*SELECT AVG(nama\_kolom) FROM nama\_tabel WHERE kondisi;*

Fungsi SUM() merupakan perintah dasar SQL yang di gunakan untuk mendapatkan jumlah record yang memenuhi suatu kriteria, tentunya nilai pada kolom harus numerik.Contoh:

*SELECT SUM(nama\_kolom) FROM nama\_tabel WHERE kondisi;*

**SQL LIKE:**

MySQL Like digunakan untuk mencari data tertentu pada sebuah kolom data.Pencarian dilakukan dengan memasukkan bentuk pattern like yang setelah Where.Bentuk Penulisan Perintah MySQL Like

contoh:

SELECT \* FROM siswa WHERE nama LIKE ‘a%’

**SQL WILDCARDS:**

Wildcard adalah operator yang digunakan untuk mensubstitusikan berbagai karakter lain dalam suatu string. Wildcard SQL digunakan juga untuk mencari data dengan menggunakan tabel pada SQL.

**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 (‘Yogyakarta’,’Jakarta’)

**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

**SQL ALIASES:**

Alias ​​SQL biasa digunakan untuk memberi tabel, atau kolom dalam tabel, nama sementara.Alias ​​sering digunakan untuk membuat nama kolom lebih mudah dibaca. Biasanya menggunakan [AS](https://dosenit.com/php/keyword-as-php) pada syntaxnya.Alias ​​hanya ada selama kueri.

Contoh:

SELECT IdCustomer AS ID, NamaCustomer AS Customer

FROM Customer;

**SQL JOINS:**

Join digunakan untuk menggabungkan baris dari dua atau lebih tabel berdasarkan kolom yang terkait di antara tabel-table tersebut, atau sering kita sebut dengan *relational database.* Joinmemungkinkan kita untuk memperoleh data dari beberapa tabel dalam satu *query*. Jika hanya menggunakan satu buah tabel, data yang kita simpan atau peroleh akan terbatas. Kita bisa saja menyimpan banyak data dari sebuah tabel, tapi tabel tersebut akan menjadi kurang baik.

Contoh:

*SELECT columns  
FROM TableA  
INNER JOIN TableB  
ON A.columnName = B.columnName;*

**SQL INNER JOIN:**

Digunakan untuk menghasilkan baris data dengan caramenggabungkan 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

**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

**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

**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 bukutamu.nama,bukutamu.email,order.no\_order  
FROM bukutamu  
FULL JOIN order  
ON bukutamu.id=order.id  
ORDER BY bukutamu.nama

**SQL SELF JOIN:**

Self JOIN adalah gabungan reguler, tetapi tabel bergabung dengan dirinya sendiri.

Contoh:

SELECT A.NamaCustomer AS NamaCustomer1, B.NamaCustomer AS NamaCustomer2, A.Kota

FROM Customers A, Customers B

WHERE A.IdCustomer <> B.IdCustomer

AND A.Kota = B.Kota

ORDER BY A.Kota;

**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

**SQL GROUP BY:**

**Group By** adalah fungsi untuk mengelompokkan data dalam sebuah kolom yang ditunjuk. Fungsi ini akan menghasilkan kelompok data dengan menghilangkan data yang sama dalam satu tabel. Maka apabila dalam satu kolom terdapat beberapa data yang sama maka data yang akan ditampilkan hanya salah satu. Contoh:

create table mahasiswa ( nik char(10)notnull primary key, nama varchar(50)notnull, jk char(1), tanggal\_lhr date, jurusan char(2), umur int);

**SQL HAVING:**

Having pada sql digunakan sebagai pengganti where dimana where tidak bisa menjalankan fungsi contohnya having count. dari sini dapat dipahami having adalah fungsi untuk memfilter data dengan kondisi khusus menggunakan fungsi lainnya.

Contoh:

SELECT alamat, COUNT(id\_siswa) FROM siswa WHERE jenis\_kelamin=’Perempuan’ GROUP BY alamat HAVING COUNT(id\_siswa)>2

**SQL EXISTS:**

Operator EXISTS digunakan untuk menguji keberadaan rekaman apa pun di subkueri.Operator EXISTS dapat mengembalikan nilai true jika subkueri mengembalikan satu atau lebih rekaman.

Contoh:

SELECT NamaSupplier

FROM Suppliers

WHERE EXISTS (SELECT NamaProduk FROM Produk WHERE Produk.IdSupplier = Suppliers.IdSupplier AND Harga < 30);

**SQL ANY, ALL:**

Operator ANY akan mengembalikan TRUE jika salah satu nilai subkueri memenuhi ketentuan.Pernyataan SQL berikut mengembalikan TRUE dan mencantumkan nama produk jika menemukan catatan ANY di tabel OrderDetails bahwa Quantity = 12:

Contoh:

SELECT NamaProduk

FROM Products

WHERE IdProduct = ANY (SELECT IdProduct FROM OrderDetails WHERE Quantity = 12);.

Pernyataan SQL berikut mengembalikan TRUE dan mencantumkan nama produk jika menemukan rekaman ANY di tabel OrderDetails bahwa kuantitas > 99:

SELECT NamaProduk

FROM Products

WHERE IdProduct = ANY (SELECT IdProduct FROM OrderDetails WHERE Quantity > 99);

Operator ALL akan mengembalikan TRUE jika semua nilai subkueri memenuhi kondisi.

Pernyataan SQL berikut mengembalikan TRUE dan mencantumkan nama produk jika ALL rekaman dalam tabel OrderDetails memiliki quantitas = 10 (jadi, contoh ini akan mengembalikan FALSE, karena tidak SEMUA rekaman dalam tabel OrderDetails memiliki quantitas= 10):

Contoh

SELECT NamaProduk

FROM Products

WHERE IdProduct = ALL (SELECT IdProduct FROM OrderDetails WHERE Quantity = 10);

**.SQL SELECT INTO:**

SELECT INTO untuk menyalin data dari tabel, dan kemudian memasukkan data ke dalam tabel baru yang lain.

*SELECT \*  
INTO*newtable*[IN*externaldb*]  
FROM*table1;

**SQL INSERT INTO SELECT:**

INSERT INTO SELECT untuk menyalin data dari meja dan kemudian memasukkan data ke tabel yang ada.Menargetkan meja setiap baris yang sudah ada tidak terpengaruh.

INSERT INTO table2  
(column\_name(s))  
SELECT column\_name(s)  
FROM table1;

**SQL CASE:**

Case statement dalam SQL mengembalikan nilai pada kondisi tertentu. Kita dapat menggunakan Case statement dalam kueri pemilihan bersama dengan klausa Where, Order By, dan Group By. Statement ini dapat digunakan dalam insert statement juga.

CASE<Case\_Expression>

WHEN Value\_1 THEN Statement\_1

WHEN Value\_2 THEN Statement\_2

WHEN Value\_N THEN Statement\_N

[ELSE Statement\_Else]

ENDAS [ALIAS\_NAME]

**SQL NULL FUNCTIONS:**

Fungsi MySQL IFNULL() dapat digunakan untuk mengembalikan nilai alternatif jika ekspresi adalah NULL:

SELECT NamaProduk, HargaPerUnit \* (UnitTersedia + IFNULL(UnitsOnOrder, 0))

FROM Produk;

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

SELECT NamaProduk, HargaPerUnit \* (UnitTersedia + COALESCE(UnitsOnOrder, 0))

FROM Produk;

**SQL STORED PROCEDURES:**

Stored Procedure adalah sekumpulan perintah SQL yang disusun dalam sebuah procedure (mirip dengan di pemrograman biasa) yang memiliki nama dan fungsi tertentu

Contoh:

CREATE PROCEDURE SelectAllCustomers

AS

SELECT \* FROM Customers

GO;

**SQL COMMENTS:**

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

Contoh:

--Select all:

SELECT \* FROM Customers;

**SQL OPERATORS:**

Operator adalah kata atau karakter khusus yang digunakan pada SQL terutama dalam klausa WHERE dimana statement 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

**SQL DATABASE:**

Structured Query Language atau SQL adalah bahasa kueri yang dirancang untuk membantu dalam pengambilan dan mengelola informasi pada sebuah database.Untuk yang masih pemula dalam dunia IT, biasanya diartikan sebagai bahasa yang digunakan untuk mengakses sebuah data dalam basis relasional.

Contoh:

CREATE DATABASE namadatabase;

**SQL CREATE DB:**

Create Database adalah pernyataan yang digunakan untuk membuat database baru.

Contoh:

CREATE DATABASE databasename;

**SQL DROB DB:**

Pernyataan DROP DATABASE dapat digunakan untuk menjatuhkan database SQL yang ada.

Contoh:

DROP DATABASE testDB;

**SQL CREATE TABLE:**

Create Table adalah pernyataan yang digunakan untuk membuat tabel baru pada basis data yang sedang aktif. Secara umum, perintah ini memiliki formulasi sebagai berikut.

Contoh:

CREATE TABLE ['nama\_tabel']

(

nama\_field1 tipe\_data [constraints][,

nama\_field2 tipe\_data,

...]

)

**SQL ALTER TABLE:**

ALTER TABLE adalah fungsi atau Query SQL yang digunakan untuk mengubah struktur table pada suatu database.Pernyataan ALTER TABLE digunakan untuk menambahkan, menghapus, atau memodifikasi kolom dalam tabel yang ada.Pernyataan ALTER TABLE juga digunakan untuk menambah dan menghilangkan berbagai kendala pada tabel yang ada.

Contoh:

*ALTER TABLE <nama\_table>*  
*ADD <nama\_kolom><tipe\_data>;*

**SQL CONSTRAINTS**

Constraints SQL digunakan untuk menentukan aturan yang memperbolehkan atau membatasi nilai / data apa yang akan disimpan di tabel. Mereka menyediakan metode yang sesuai untuk memastikan keakuratan dan integritas data di dalam tabel. Hal ini juga membantu untuk membatasi jenis data yang akan disisipkan di dalam tabel. Jika ada gangguan yang terjadi antara batasan dan tindakan data, tindakan tersebut gagal.

CREATETABLE Siswa (

Id INT PRIMARY\_KEY,

Nama VARCHAR (255) NOT\_NULL,

Umur INT NOT\_NULL,

....

);

**SQL NOT NULL:**

Operator IS NOT NULL digunakan untuk menguji nilai yang tidak kosong (nilai NOT NULL).

Contoh:

SELECT NamaPelanggan, NamaKontak, Alamat

FROM Pelanggan

WHERE Alamat IS NOT NULL;

**SQL UNIQUE:**

Constraint UNIQUE digunakan untuk dapat memastikan bahwa semua nilai dalam kolom unik.

Contoh:

CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Usia int,

CONSTRAINT UC\_Person UNIQUE (ID,LastName)

);

**SQL PRIMARY KEY:**

Primary key dapat diartikan sebagai kolom yang berisi nilai unik, berfungsi sebagai identitas untuk membedakan setiap record yang ada pada tabel. primary key berfungsi sebagai identitas untuk membedakan antar record pada tabel.

Contoh:

create table pembeli ( id\_pembeli intnotnull auto\_increment primary key, nama\_pembeli varchar(30)notnull, jk char(1), no\_telp char(14), alamat varchar(50));

**SQL FOREIGN KEY**

*Foreign key*yaitu satu atribute (atau satu set atribute) yang melengkapi satu relationship (hubungan) yang menunjukkan ke induknya. *Foreign key* berguna untuk mendefinisikan kolom-kolom pada suatu tabel yang nilainya mengacu ke tabel lain,jadi kolom foreign key nilainya harus diambil dari nilai kolom pada tabel lain.

Contoh:

Pada tabel “DataMhs”, NIM merupakan Primary key.Sedangkan pada Tabel “DataNilaiMhs” NIM merupakan Foreign key.Sebab, “NIM” mendefinisikan nilai satu tabel dengan tabel lainnya.

|  |  |
| --- | --- |
| DataMhs | |
| NIM | Nama |
| 2015220102 | Udin Keling |

|  |  |  |
| --- | --- | --- |
| DataNilaiMhs | | |
| NIM | Matkul | Nilai |
| 2015220102 | B\_Ing2 | C+ |

**SQLCHECK:**

Check constraint digunakan untuk memvalidasi data yang masuk apakah sesuai dengan kondisi yang telah ditentukan atau tidak. Check constraint selalu bernilai **true** atau **false**. Jika bernilai true artinya data boleh dimasukkan, sedangkan bila bernilai false data ditolak.

Contoh:

SQL> CREATE TABLE pegawai(

2 kode\_pegawai VARCHAR2(5) NOT NULL,

3 nama\_pegawai VARCHAR2(40) NOT NULL,

4 gender CHAR(1) NOT NULL CHECK(gender IN('L','P')),

5 department VARCHAR2(40) NOT NULL CHECK(department = 'IT'));

Table created.

**SQL DEFAULT:**

Constraint DEFAULT dapat digunakan untuk memberikan nilai default untuk kolom.Nilai default akan ditambahkan ke semua rekaman baru JIKA tidak ada nilai lain yang ditentukan.

Contoh:

CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Usia int,

Kota varchar(255) DEFAULT 'Sandnes'

);

**SQL INDEQS:**

Sebuah indeks database adalah sebuah struktur data yang meningkatkan kecepatan operasi dalam sebuah tabel.Indeks dapat dibuat menggunakan satu atau lebih kolom, menyediakan dasar untuk kedua pencarian acak yang cepat dan efisien pemesanan akses ke catatan.

Sementara menciptakan indeks itu harus dipertimbangkan bahwa apa yang kolom yang akan digunakan untuk membuat query SQL dan membuat satu atau lebih indeks pada kolom tersebut.

Contoh:

mysql> ALTER TABLE testalter\_tbl ADD INDEX (c);

**SQL AUTO INCREMENT:**

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.

Contoh:

CREATE TABLE contohTutorial (

ID int NOT NULL AUTO\_INCREMENT,

nama varchar(255) NOT NULL,

alamat varchar(255),

PRIMARY KEY (ID)

);

**SQL DATES:**

Hal yang tidak mudah saat menggunakan fungsi Date pada SQL adalah memastikan bahwa [format](https://dosenit.com/software/dbms/mysql/fungsi-string-format-pada-mysql) tanggal yand dimasukkan bisa cocok dengan format kolom tanggal pada database.

mysql> CREATETABLEcontoh\_date (dat DATE, tim TIME, dattim DATETIME,

timestam TIMESTAMP, yea YEAR);

Query OK, 0 rowsaffected (0.06 sec)

mysql> DESCRIBE contoh\_date;

+----------+-----------+------+-----+-------------------+-----------------------------+

| Field    | Type      | Null| Key| Default           | Extra                       |

+----------+-----------+------+-----+-------------------+-----------------------------+

| dat      | date      | YES  |     | NULL              |                             |

| tim      | time      | YES  |     | NULL              |                             |

| dattim   | datetime  | YES  |     | NULL              |                             |

| timestam | timestamp| NO   |     | CURRENT\_TIMESTAMP| onupdateCURRENT\_TIMESTAMP|

| yea      | year(4)   | YES  |     | NULL              |                             |

+----------+-----------+------+-----+-------------------+-----------------------------+

5 rowsinset(0.01 sec)

**SQL VIEWS:**

SQL View adalah tabel virtual (bukan tabel sebenarnya) yang dibuat dari beberapa tabel atau view lain. SQL View tidak memiliki data sendiri, tetapi data-datanya berasal dari tabel-tabel atau view lain yang sudah dibentuk dari kueri SQL yang telah ditentukan sebelumnya.

SQL View dibuat atau dibangun dengan SELECT statement. Di dalam query SELECT tersebut kita bisa menambahkan perintah ORDER BY untuk mengurutkan data atau tidak sama sekali. SQL View dapat digunakan untuk memudahkan atau menyederhanakan data yang ditampilkan, mencegah menampilkan data yang penting dengan menyembunyikan nama kolomnya.

Contoh:

CREATE VIEW view\_name AS

SELECT column1, column2.....

FROM table\_name

WHERE [condition];

**SQL INJECTION:**

SQL injection atau biasa yang dikenal dengan sebutan **SQLi adalah** suatu teknik penyerangan web dengan menggunakan kode SQL (Structured Query Language) yang berbahaya untuk memanipulasi database. Tindakan ini termasuk kedalam kategori hacking, dimana penyerang akan mencari celah keamanan dari suatu website.

Contoh:

SELECT ItemName, ItemDescription

FROM Items

WHERE ItemNumber = 999; DROP TABLE USERS

**SQL HOSTING:**

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

**SQL DATA TYPES:**

Secara umum tipe data pada MySQL dibagi menjadi 4 yaitu:

Numeric Values yaitu angka atau bilangan seperti 10; 123; 100.50; -10; 1.2E+17; 2.7e-11; dan sebagainya.Tipe data ini dibagi menjadi dua yaitu Bilangan Bulat (Integer) dan Bilangan Pecahan (Floating-point).Bilangan bulat adalah bilangan tanpa tanda desimal sedangkan bilangan pecahan adalah bilangan dengan tanda desimal.Kedua jenis bilangan ini dapat bernilai positif (+) dan juga negatif (-). Jika bilangan tersebut menggunakan tanda positif (+) atau (-), maka disebut SIGNED. Sebaliknya bila tanpa tanda apapun disebut UNSIGNED. Karena tanda positif (+) dapat diabaikan penulisannya maka pada bilangan yang bernilai positif disebut UNSIGNED.

String/Character Values adalah semua karakter (atau teks) yang penulisannya selalu diapit oleh tanda kutip baik kutip tunggal (‘) maupun kutip ganda (“). Hal ini tidak hanya berlaku pada huruf alfabet saja tetapi angka yang ditulis dengan tanda kutip pun akan menjadi karakter atau string.

Date and Time Values yaitu tanggal dan waktu. Untuk jenis data tanggal dan waktu format standar (default) penulisan tanggalnya adalah “tahun-bulan-tanggal”, Misalnya untuk 22 Januari 2001 dituliskan “2002-01-22”. Untuk penulisan waktu, formatnya adalah “jam-menit-detik”.Contoh, “13:55:07”.Data tanggal dan waktu bisa digabung penulisannya menjadi “2002-01-22 13:55:07”.

NULL. NULL sebenarnya bukan data, tapi dia mewakili sesuatu yang “tidak pasti”, “tidak diketahui” atau “belum ada nilainya”. Sebagai contoh dalam kehidupan sehari-hari Anda melakukan suatu survei berapa jumlah pengguna sistem operasi LINUX dan Windows di Indonesia.Selama survei belum tuntas maka data pastinya belum dapat diketahui.Oleh sebab itu, data tersebut bisa diwakili dengan NULL, alias belum diketahui.

**SQL REFERENCES:**

Integeritas referensial merupakan suatu upaya untuk memnjaga agar Foreign Key (FK) dan Primary Key (PK) milik tabel lain selalu konsisten. Hal ini bertujuan untuk menjamin nilai dalam suatu tabel yang merujuk ke tabel lain benar-benar nilai yang terdapat pada tabel yang dirujuk. Ada beberapa syarat yang harus terpenuhi seperti, antara tabel yang ingin dihubungkan harus memiliki kolom dengan tipe data yang sama. Misalnya, terdapat tabel barang dan

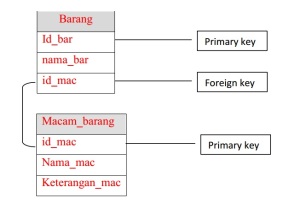
macam\_barang. Berikut query kedua tabel tersebut :

create table macam\_barang (id\_mac integer not null, nama\_mac varchar(15), guna\_mac varchar(40), primary key (id\_mac));

create table barang (id\_bar integer not null, nama\_bar varchar(15), id\_mac integer not null references macam\_barang, primary key (id\_bar, id\_mac));

Atau ketika kita lupa memberikan references pada tabel macam barang kita dapat memberikan query sebagai berikut :

alter table barang add constraint relasi\_barang foreign key (id\_mac) references macam\_barang (id\_mac) on update no action on delete no action;

[](https://pipeh0009.files.wordpress.com/2014/10/contoh-fk.jpg)

**SQLKEYWORDS:**

Perintah SET digunakan dengan [UPDATE](https://dosenit.com/sql/statement-update-pada-sql-syntax-dan-contohnya) untuk menentukan kolom dan nilai mana yang harus diperbarui dalam tabel.

Contoh:

UPDATE Customers

SET NamaKontak = 'Alfred Schmidt', Kota= 'Frankfurt'

WHERE CustomerID = 1;

**MYSQL FUNCTIONS:**

Digunakan untuk mengkalkulasikan data pada tabel-tabel didalam database.

Ada 2 Bagian dari SQl Function yaitu;

1. SQL Aggregate Functions

Digunakan untuk mengkalkulasikan nilai-nilai didalam kolom. Fungsi Aggregate terdiri dari fungsi:

1. AVG() – menghitung rata-rata
2. COUNT() – menghitung jumlah row
3. FIRST() – nilai pertama
4. LAST() – nilai terakhir
5. MAX() – nilai maksimal
6. MIN() – nilai minimal
7. SUM() – menjumlahkan
8. AVG

Syntax AVG ( )

SELECT AVG(column\_name) FROM table\_name

Contoh:

SELECT AVG(harga\_order) FROM Order

untuk menampilkan kostumer  yang mempunyai nilai diatas rata-rata

SELECT kustomer FROM Order  
WHERE harga\_order>(SELECT AVG(Harga\_Order) FROM Order)

COUNT

Syntax Count

SELECT COUNT(column\_name) FROM table\_name

contoh:

SELECT COUNT(kustomer) AS kustomerAyu FROM Order  
WHERE Kustomer=’Ayu’

Syntax SQL COUNT(\*)

SELECT COUNT(\*) FROM table\_name

contoh:

SELECT COUNT(\*) AS BanyaknyaOrder FROM Order

Syntax SQL COUNT(DISTINCT column\_name)

SELECT COUNT(DISTINCT column\_name) FROM table\_name

contoh:

SELECT COUNT(DISTINCT kustomer) AS banyaknya\_Kustomer FROM Order

FIRST

Syntax SQL FIRST

SELECT FIRST(column\_name) FROM table\_name

Contoh:

SELECT FIRST(harga\_order) AS HargaOrderPertama FROM Order

LAST

**SQL SERVER FUNCTIONS:**

Fungsi agregat adalah fungsi yang digunakan untuk mengkoleksi sejumlah data yangtersimpan pada suatu field dan menampilkannya dalam bentuk nilai tunggal. Fungsi agregat yang ada dalam SQL adalah SUM, MAX, MIN, AVG dan COUNT. Fungsi agregat ini bisa digunakan pada perintah SELECT untuk menampilkan data dengan memunculkan nilai tunggal. Kegunaan masing-masing fungsi agregat antara lain :

·         
         SUM digunakan untuk menghitung total jumlah data yang tersimpan pada sebuah field. Bentuk umum penggunakan fungsi SUM adalah :

            SUM(nama\_field)

·            
     MAX digunakan untuk menentukan nilai tertinggi dari data yang tersimpan pada sebuah field. Bentuk umum penggunaan fungsi MAX adalah :

            MAX(nama\_field)

·            
     MIN digunakan untuk menentukan nilai terendah dari data yang tersimpan pada sebuah field. Bentuk umum penggunaan fungsi MAX adalah :

 MIN(nama\_field)

  AVG digunakan untuk mencari nilai rata-rata dari data yang tersimpan pada sebuah field. Bentuk umum penggunaan fungsi AVG adalah :

 AVG(nama\_field)

COUNT digunakan untuk menghitung jumlah record dalam sebuah tabel. Bentuk umum penggunaan fungsi COUNT adalah :

COUNT(nama\_field)

.Fungsi Scalar SQL Server  
SQL Scalar yaitu fungsi yang digunakan untuk mendapatkan nilai tunggal berdasarkan nilai inputan, contohnya :

ROUND() untuk membulatkan nilai decimal  
LEN() mendapatkan panjang karakter dari sebuah field text  
NOW() mendapatkan data tanggal dan waktu saat ini.  
UCASE() merubah field text menjadi upper case atau huruf besar  
LCASE() merubah field text menjadi lower case atau huruf kecil

b.Fungsi Table Value SQL Server

 Jika Scalar function dapat mengembalikan sebuah nilai dengan tipe data tertentu, maka table value function dapat mengembalikan nilai berupa table. Table valued function ini dibedakan menjadi dua jenis yaitu Inline table valued function dan multistatement table valued function. Inline valued function dapat digunakan jika hasil yang akan dikembalikan merupakan hasil dari perintah SELECT. Bagian dari deklarasi inline function tidak didalam blok BEGIN END. Fungsi Inline table valued function ini mirip seperti view, perbedaanya adalah jika pada view tidak bisa menggunakan parameter, sedangkan pada function diijinkan mengggunaan parameter.

*CREATE FUNCTION [dbo].[test] ( @X INT, @Y INT )  
RETURNS INT  
AS  
BEGIN  
RETURN @X + @Y  
END  
GO*

c.Fungsi System SQL Server

Kelompok fungsi sistem adalah kelompok fungsi yang memberikan informasi tentang pemakaian server database oleh pemakai. Kelompok fungsi ini terdiri atas :

§  CURRENT\_USER ( )

menghasilkan kombinasi nama user dan nama host pada sesi saat ini yang terautentifikasi. Hasil output dapat berbeda dari nilai USER ().

§  DATABASE ( )

 menampilkan nama database yang sedang digunakan.

§  VERSION ( )

menampilkan versi MySQL yang sedang aktif.

§  LAST\_INSERT\_ID( )

digunakan untuk menampilkan informasi data yang telah dihasilkan (di-generate) oleh MySQL pada kolom yang menggunakan tipe data AUTOINCREMENT.

§  SESSION\_USER( )

digunakan untuk menampilkan informasi pemakain yangsedang melakukan akses ke dalam server MySQL saat ini. Fungsi ini kadang digunakan oleh seorang pemakai., untuk mengetahui siapa saja yang sedang login ke dalam MySQL pada suatu terminal yang sedang ditinggalkan oleh pemakai lain, atau kadang juga kita ingin tahu kita sedang mengakses MySQL dengan nama pemakai yang mana, akrena kita memiliki banyak nama pemakai yang berbeda-beda privilegenya demi keamanan

§  SYSTEM USER( )

fungsi ini sama dengan fungsi SESSION\_USER( ).

§  USER( )

fungsi ini sama dengan fungsi SYSTEM\_USER( ) dan  SESSION\_USER( ).

§  BENCHMARK(count, ekspresi)

mengerjakan perintah pada ekspresi sejumlah count kali. Nilai yang dihasilkan selalu 0. Hal penting dari fungsi ini adalah nilai waktu yang digunakan (elapsed time) yang diberikan pada bagian akhir, memungkinkan kita menilai berpaa cepat server mengevaluasi query. Berikut ini contoh tampilan fungsi benchmark untuk mengetahui kecepatan 10 kali perintah select \* from pinjaman.

§  LOAD\_FILE(namafile)

digunakan untuk membuka file dan memberikan isinya sebagai string. File harus ada di dalam server, dan pemakai untuk menggunakan fungsi ini harus memiliki privilege FILE

**MS ACCESS FUNCTIONS:**

Pengertian Microsoft Access atau biasa disebut Ms.Access adalah sebuah program atau aplikasi basis data (database) yang dicanangkan oleh Microsoft dalam sebuah paket bernama Microsoft Office.[Database](https://idmanajemen.com/definisi-basis-data/) sendiri merupakan sekumpulan informasi yang sistematis dan tersimpan dalam komputer sehingga sewaktu waktu dapat diperiksa oleh software yang terinstall di komputer guna memperoleh informasi dari basis data tersebut.

Fungsi utama dari Microsoft Access adalah untuk menangani proses manipulasi data dan berperan dalam pembuatan sebuah sistem. Dengan hadirnya microsoft access ini diharapkan dapat memberikan manfaat besar yang berkaitan dengan media untuk membuat aplikasi web dasar. Fungsi lain dari program microsoft access diantaranya :

1. Untuk membuat program aplikasi berkaitan dengan jumlah peserta didik.
2. Untuk membuat laporan keuangan.
3. Untuk membuat program aplikasi gaji karyawan.
4. Untuk membuat form dan sub form.
5. Untuk membuat basis data (database).
6. Untuk membuat program aplikasi penyimpanan buku perpustakaan.
7. Untuk membuat program aplikasi daftar hadir.
8. Untuk membuat program aplikasi persediaan barang (inventory), dan juga fungsi-fungsi tambahan lainnya sesuai dengan tujuan penggunaan microsoft access.

Program microsoft access mempunyai banyak sekali fungsi yang membantu kinerja seseorang dalam mengolah data sebagaimana telah disebutkan sebelumnya.Dari sekian banyak fungsi yang bisa diperankan oleh Microsoft Access, tentunya program ini mempunyai manfaat yang cukup signifikan dalam menambah tingkat kinerja seseorang. Selain menghemat waktu dan tenaga dalam mengolah database, tentunya microsoft access dapat mempermudah pelaksanaan pekerjaan yang tadinya terasa berat dan menjemukan.

Kelebihan program Microsoft access adalah mudah digunakan, praktis, serta user friendly atau mudah digunakan. Meskipun begitu, program ini juga mempunyai kekurangan berupa keharusan bagi penggunanya untuk membayar jika ingin menggunakan fitur-fitur yang ada dalam microsoft access. Biasanya program ini sudah di-bundling dengan program besutan Microsoft Office lainnya.

**SQL QUICK REF:**

|  |  |
| --- | --- |
| **SQL Statement** | **Syntax** |
| AND / OR | SELECT column\_name(s) FROM table\_name WHERE condition AND|OR condition |
| ALTER TABLE | ALTER TABLE table\_name ADD column\_name datatype  or  ALTER TABLE table\_name DROP COLUMN column\_name |
| AS (alias) | SELECT column\_name AS column\_alias FROM table\_name  or  SELECT column\_name FROM table\_name  AS table\_alias |
| BETWEEN | SELECT column\_name(s) FROM table\_name WHERE column\_name BETWEEN value1 AND value2 |
| CREATE DATABASE | CREATE DATABASE database\_name |
| CREATE TABLE | CREATE TABLE table\_name ( column\_name1 data\_type, column\_name2 data\_type, column\_name3 data\_type, ... ) |
| CREATE INDEX | CREATE INDEX index\_name ON table\_name (column\_name)  or  CREATE UNIQUE INDEX index\_name ON table\_name (column\_name) |
| CREATE VIEW | CREATE VIEW view\_name AS SELECT column\_name(s) FROM table\_name WHERE condition |
| DELETE | DELETE FROM table\_name WHERE some\_column=some\_value  or  DELETE FROM table\_name (**Note:**Deletes the entire table!!)  DELETE \* FROM table\_name (**Note:**Deletes the entire table!!) |
| DROP DATABASE | DROP DATABASE database\_name |
| DROP INDEX | DROP INDEX table\_name.index\_name (SQL Server) DROP INDEX index\_name ON table\_name (MS Access) DROP INDEX index\_name (DB2/Oracle) ALTER TABLE table\_name DROP INDEX index\_name (MySQL) |
| DROP TABLE | DROP TABLE table\_name |
| EXISTS | IF EXISTS (SELECT \* FROM table\_name WHERE id = ?) BEGIN --do what needs to be done if exists END ELSE BEGIN --do what needs to be done if not END |
| GROUP BY | SELECT column\_name, aggregate\_function(column\_name) FROM table\_name WHERE column\_name operator value GROUP BY column\_name |
| HAVING | 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 |
| IN | SELECT column\_name(s) FROM table\_name WHERE column\_name IN (value1,value2,..) |
| INSERT INTO | INSERT INTO table\_name VALUES (value1, value2, value3,....)  *or*  INSERT INTO table\_name (column1, column2, column3,...) VALUES (value1, value2, value3,....) |
| INNER JOIN | SELECT column\_name(s) FROM table\_name1 INNER JOIN table\_name2 ON table\_name1.column\_name=table\_name2.column\_name |
| LEFT JOIN | SELECT column\_name(s) FROM table\_name1 LEFT JOIN table\_name2 ON table\_name1.column\_name=table\_name2.column\_name |
| RIGHT JOIN | SELECT column\_name(s) FROM table\_name1 RIGHT JOIN table\_name2 ON table\_name1.column\_name=table\_name2.column\_name |
| FULL JOIN | SELECT column\_name(s) FROM table\_name1 FULL JOIN table\_name2 ON table\_name1.column\_name=table\_name2.column\_name |
| LIKE | SELECT column\_name(s) FROM table\_name WHERE column\_name LIKE pattern |
| ORDER BY | SELECT column\_name(s) FROM table\_name ORDER BY column\_name [ASC|DESC] |
| SELECT | SELECT column\_name(s) FROM table\_name |
| SELECT \* | SELECT \* FROM table\_name |
| SELECT DISTINCT | SELECT DISTINCT column\_name(s) FROM table\_name |
| SELECT INTO | SELECT \* INTO new\_table\_name [IN externaldatabase] FROM old\_table\_name  *or*  SELECT column\_name(s) INTO new\_table\_name [IN externaldatabase] FROM old\_table\_name |
| SELECT TOP | SELECT TOP number|percent column\_name(s) FROM table\_name |
| TRUNCATE TABLE | TRUNCATE TABLE table\_name |
| UNION | SELECT column\_name(s) FROM table\_name1 UNION SELECT column\_name(s) FROM table\_name2 |
| UNION ALL | SELECT column\_name(s) FROM table\_name1 UNION ALL SELECT column\_name(s) FROM table\_name2 |
| UPDATE | UPDATE table\_name SET column1=value, column2=value,... WHERE some\_column=some\_value |
| WHERE | SELECT column\_name(s) FROM table\_name WHERE column\_name operator value |