

SISTEM BASIS DATA

Dosen Pengampu : Musliadi KH, S.Kom., M.Kom

Nama : Nurul Annisa Adawia

Nim : 200250502052

Prodi : Teknik Informatika (B)

➤ SQL Tutorial Meliputi:

- **Structured Query Language (SQL)** yang dapat dikatakan sebagai bahasanya database. Hampir semua aplikasi database relational, seperti [MySQL](#), [Oracle](#), dan [Microsoft SQL Server](#) menggunakan *SQL* untuk menangani perintah-perintah dalam operasional database.

- Pengertian Structured Query Language (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'.

- Sejarah SQL

Bersamaan dengan paper Dr. Edgar F. Codd pada tahun 1969 tentang [Teori Database Relational](#), ia pun mengajukan sebuah bahasa yang disebut *DSL/Alpha* untuk manajemen data dalam relational database. Berdasarkan ide Dr.Codd ini, beberapa saat setelah itu IBM mencoba merancang bahasa prototipe sederhana *DSL/Alpha* yang disebut *SQUARE*. Pada tahun 1970, team yang beranggotakan peneliti IBM Donald D. Chamberlin dan Raymond F. Boyce, mengembangkan *SQUARE* lebih lanjut menjadi *SEQUEL (Structured English Query Language)*. *SEQUEL* digunakan untuk mengoperasikan prototipe RDBMS pertama IBM, *System R*. Di kemudian hari, *SEQUEL* berubah nama menjadi **SQL** karena permasalahan merk dagang (*trademark*) dengan sebuah perusahaan pesawat di inggris yang terlebih dahulu telah memakai nama *SEQUEL*. Pada akhir 1970an, perusahaan **Relational Software, Inc.** (sekarang [Oracle Corporation](#)) melihat potensi bahasa SQL dan mengembangkan sendiri versi SQL untuk RDBMS

mereka. **Oracle V2 (versi 2)** yang dirilis Juni 1979 adalah RDBMS komersial pertama yang mengimplementasikan SQL. Dengan kemudahan yang ditawarkan, SQL mulai diimplementasikan oleh berbagai RDBMS dengan versi SQL mereka masing-masing. Namun hal ini menimbulkan permasalahan karena perbedaan penerapan SQL dari satu aplikasi dengan aplikasi database lainnya yang tidak seragam. Sehingga pada tahun 1986, badan standar amerika, **ANSI (American National Standards Institute)** merancang sebuah standar untuk SQL. Satu tahun setelahnya, **ISO (International Organization for Standardization)** juga mengeluarkan standar untuk SQL. Versi terakhir standar SQL dirilis pada 2011, yang dinamakan SQL 2011. Dengan standar ini diharapkan ada keseragaman SQL antar aplikasi RDBMS. Akan tetapi walaupun sudah ada standar tentang SQL, banyak perusahaan RDBMS yang menambahkan 'fitur' SQL selain standar yang ada. MySQL juga memiliki SQL yang tidak standar, yang tidak ada pada Oracle, begitu juga sebaliknya. Namun setidaknya bahasa SQL hampir sama untuk perintah-perintah dasar antar RDBMS. Perintah SQL untuk membuat tabel misalnya, dapat digunakan baik di Oracle maupun MySQL.

➤ Jenis-jenis perintah SQL

Perintah atau instruksi SQL dapat dikelompokkan berdasarkan jenis dan fungsinya. Terdapat 3 jenis perintah dasar SQL : *Data Definition Language*, *Data Manipulation Language* dan *Data Control Language*.

- **Data Definition Language (DDL)** *adalah jenis instruksi SQL yang berkaitan dengan pembuatan struktur tabel maupun database.* Termasuk diantaranya : CREATE, DROP, ALTER, dan RENAME.
- **Data Manipulation Language (DML)** *adalah jenis instruksi SQL yang berkaitan dengan data yang ada dalam tabel, tentang bagaimana menginput, menghapus, memperbaharui serta membaca data yang tersimpan di dalam database.* Contoh perintah SQL untuk DML : SELECT, INSERT, DELETE, dan UPDATE.
- **Data Control Language (DCL)** *adalah jenis instruksi SQL yang berkaitan dengan manajemen hak akses dan pengguna (user) yang dapat mengakses database maupun tabel.* Termasuk diantaranya : GRANT dan REVOKE.

Selain ketiga jenis perintah SQL, terdapat juga 2 jenis SQL tambahan : *Transaction Control Language*, dan *Programmatic SQL*.

- **Transaction Control Language (TCL)** *adalah perintah SQL untuk proses transaksi.* Proses transaksi ini digunakan untuk perintah yang lebih dari 1, namun harus berjalan semua, atau tidak sama sekali. Misalnya untuk aplikasi critical seperti transfer uang dalam sistem database perbankan. Setidaknya akan ada 2 perintah, yaitu mengurangi uang nasabah A, dan menambah uang nasabah B. Namun jika terjadi kesalahan sistem, kedua transaksi ini harus dibatalkan. Tidak

bisa hanya satu perintah saja. Termasuk ke dalam TCL adalah perintah : COMMIT, ROLLBACK, dan SET TRANSACTION.

- **Programmatic SQL** berkaitan dengan sub program (stored procedure) maupun penjelasan mengenai struktur database. Contoh perintah seperti : DECLARE, EXPLAIN, PREPARE, dan DESCRIBE.

- SQL Home
 - SQL Tutorial mengajarkan Anda cara menggunakan SQL di: MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres, dan sistem database lainnya.

- SQL Intro
 - Apa itu SQL?
 - SQL adalah singkatan dari Structured Query Language.
 - SQL memungkinkan Anda mengakses dan memanipulasi database.
 - SQL menjadi standar American National Standards Institute (ANSI) pada 1986, dan International Organization for Standardization (ISO) pada 1987.

 - Apa yang Dapat SQL lakukan?
 - SQL dapat mengeksekusi query terhadap database
 - SQL dapat mengambil data dari database
 - SQL dapat menyisipkan catatan dalam database
 - SQL dapat memperbarui catatan dalam database
 - SQL dapat menghapus catatan dari database
 - SQL dapat membuat database baru
 - SQL dapat membuat tabel baru dalam database
 - SQL dapat membuat prosedur tersimpan dalam database
 - SQL dapat membuat tampilan dalam database
 - SQL dapat mengatur izin pada tabel, prosedur, dan tampilan

- SQL Syntax
 - **Syntax** berarti seperangkat aturan grammar pada kata atau klausa dalam membentuk kalimat. Biasanya, dalam kalimat tertulis maupun percakapan, kita melihat elemen **syntax** pada kata atau klausa yang biasa kita sebut sebagai subject, verb, dan object.

 - Contoh Syntax dalam perintah sederhana SQL :
Syntax :

```
SELECT * FROM (nama table);
```

Contoh :

```
SELECT * FROM Transaksi;
```

➤ SQL Select

- Select merupakan perintah dasar SQL yang di gunakan untuk memilih data dari database. Data yang di kembalikan di simpan dalam table yang di sebut result-set.

- Contoh Select :

```
SELECT kolom1, kolom2, ...FROM nama_tabel;
```

```
SELECT * FROM nama_tabel;
```

➤ SQL Select Distinct

- 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 :

```
SELECT DISTINCT kolom1, kolom2, ... FROM nama_tabel;
```

➤ SQL WhereSQL And, Or, Not

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

-Contoh WhereSQL :

```
SELECT kolom1, kolom2, ... FROM nama_tabel WHERE kondisi;
```

- 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.

- Contoh And, Or, Not:

```
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

-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 Order By:

```
SELECT kolom1, kolom2, ... FROM nama_tabel ORDER BY column DESC;  
SELECT nis, nama FROM siswa ORDER BY tahun_lahir DESC;
```

➤ SQL Insert Into

- 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:

```
INSERT INTO nama_tabel VALUES (nilai1, nilai2, nilai3, ...);  
INSERT INTO nama_tabel (kolom1, kolom2) VALUES (nilai1, nilai2);
```

➤ SQL Null Values

- 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.

-Contoh Null Values:

```
SQL> SELECT ID, NAME, AGE, ADDRESS, SALARY  
FROM CUSTOMERS  
WHERE SALARY IS NULL;
```

➤ SQL UpdateSQL Delete

-Update Digunakan untuk mengubah/memperbarui data di tabel database.

- Contoh Update:

Syntax dasar:

UPDATE table_name

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

WHERE some_column=some_value

Contoh :

UPDATE bukutamu

SET email='arini@yahoo.com', kota='Jakarta'

WHERE

-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.

-Contoh Delete:

Syntax dasar :

DELETE FROM table_name

WHERE some_column=some_value

Contoh:

DELETE FROM bukutamu

WHERE id=1



Select Top

- SQL SELECT TOP digunakan untuk mengambil record dari satu atau lebih tabel dalam database dan membatasi jumlah record yang dikembalikan berdasarkan nilai atau persentase tetap.

- Contoh Select Top:

SELECT TOP (top_value) [PERCENT]

expressions

FROM tables

[WHERE conditions]

[ORDER BY expression [ASC | DESC]];

➤ SQL Min and Max

- 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 ORDER BY, fungsi MIN() hanya menampilkan satu record saja yang memenuhi kriteria yang Anda tentukan.

-Contoh Min:

SELECT MIN(nama_kolom) FROM nama_tabel WHERE kondisi;

- 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 Max:

SELECT MAX(nama_kolom) FROM nama_tabel WHERE kondisi;

➤ SQL Count, Avg, Sum

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

-Contoh Count:

SELECT COUNT(nama_kolom) FROM nama_tabel WHERE kondisi;

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

-Contoh Avg:

SELECT AVG(nama_kolom) FROM nama_tabel WHERE kondisi;

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

-Contoh Sum:

SELECT SUM(nama_kolom) FROM nama_tabel WHERE kondisi;

➤ SQL Like

- Digunakan bersama dengan perintah Where, untuk proses pencarian data dengan spesifikasi tertentu.

-Contoh Like :

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".

➤ SQL Wildcards

- Dalam SQL, SQL LIKE operator dengan karakter wildcard digunakan bersama-sama.

- Contoh SQL Wildcards:

```
SELECT * FROM Website
MANA url LIKE 'https%';
```

➤ SQL In

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

- Contoh In:

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.

- Contoh SQL Between:

Syntax dasar:

```
SELECT * FROM Employees
WHERE JoiningDate BETWEEN '01-01-2015' AND `01-01-2020`;
```


➤ SQL Aliases

- untuk nama tabel atau kolom nama dengan menggunakan SQL. Pada dasarnya, membuat alias untuk memungkinkan nama kolom lebih mudah dibaca.

-Contoh Aliases:

Aliases kolom sintaks SQL

```
SELECT column_name AS alias_name
```

```
FROM table_name;
```

SQL sintaks tabel alias

```
SELECT column_name(s)
```

```
FROM table_name AS alias_name;
```

➤ SQL Joins

- SQL klausa JOIN ke baris dari dua atau lebih tabel bersama-sama, berdasarkan bidang umum antara meja.

-Contoh Joins:

```
SELECT Websites.id, Websites.name, access_log.count, access_log.date
```

```
DARI Website
```

```
INNER JOIN access_log
```

```
ON Websites.id = access_log.site_id;
```

➤ 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.

-Contoh Inner Join:

```
SELECT Orders.ID, Customers.Name
```

```
FROM Orders
```

```
INNER JOIN Customers ON Orders.ID = Customers.ID;
```

➤ 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).

-Contoh Left Join:

```
SELECT Customers.CustomerName, Orders.OrderID
```

```
FROM Customers
```

```
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID
ORDER BY Customers.CustomerName;
```

➤ SQL Right Join

- Digunakan untuk menghasilkan baris data dari tabel kanan (nama tabel kedua) yang tidak ada pasangannya pada tabel kiri (nama tabel pertama).

-Contoh Right Join:

```
SELECT Orders.OrderID, Employees.LastName
FROM Orders
RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
ORDER BY Orders.OrderID;
```

➤ SQL Full Join

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

-Contoh SQL Full Join:

```
SELECT Customers.Name, CustomerOrders.ID
FROM Customers
FULL OUTER JOIN Orders ON Customers.ID = CustomerOrders.customerID
ORDER BY Customers.Name;
```

➤ SQL Self Join

- Gabung diri adalah gabungan biasa, tetapi tabel bergabung dengan dirinya sendiri.

-Contoh Self Join:

```
SELECT column_name(s)
FROM table1 T1, table1 T2
WHERE condition;
```

➤ SQL Union

- Digunakan untuk menggabungkan hasil dari 2 atau lebih perintah Select.

-Contoh SQL Union:

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
```

➤ SQL Group B

- Digunakan untuk mengelompokkan data dengan kriteria tertentu.
- Contoh SQL Group By:

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
```

➤ SQL Having

- Digunakan untuk memfilter data dengan fungsi tertentu.
- Contoh SQL Having:

```
SELECT COUNT(Department), Department  
FROM Employees  
GROUP BY Department  
HAVING COUNT(Department) > 2;
```

➤ SQL Extract

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

- Contoh SQL Extract:

Syntax dasar :

```
Extract(unit FROM date)
```

Keterangan :

Parameter unit dapat berupa :

MICROSECOND

SECOND

MINUTE

HOURL

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
```

➤ SQL Any, All

- Operator ANY dan ALL memungkinkan Anda melakukan perbandingan antara nilai kolom tunggal dan rentang nilai lainnya.

-Contoh Any, All:

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

-Contoh All:

➤ SQL Select Into

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

- Contoh Select Into :

```
SELECT *  
INTO WebsitesBackup2016  
FROM Websites;
```

➤ Insert Into Select.

- INSERT INTO SELECT untuk menyalin data dari meja dan kemudian memasukkan data ke tabel yang ada.

- Contoh Insert Into Select.

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

➤ SQL Case

- 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
```

- 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.
```

➤ SQL Null Functions

-jika ada "UnitsOnOrder" nilai adalah NULL, maka hasilnya adalah NULL. Fungsi Microsoft ISNULL () digunakan untuk menentukan bagaimana menangani nilai NULL. NVL (), IFNULL () dan Bersatu () fungsi juga dapat mencapai hasil yang sama. Di sini, kami berharap nilai NULL adalah 0.

Sekarang, jika "UnitsOnOrder" adalah NULL, itu tidak akan mempengaruhi perhitungan, karena jika nilai NULL, ISNULL () mengembalikan 0:

-Contoh Null Functions

```
SELECT ProductName,UnitPrice*(UnitsInStock+UnitsOnOrder)  
FROM Products
```

➤ SQL Stored Procedures

- Prosedur tersimpan adalah kode SQL yang disiapkan yang dapat Anda simpan, sehingga kode tersebut dapat digunakan kembali berulang kali.

Jadi, jika Anda memiliki kueri SQL yang Anda tulis berulang kali, simpan sebagai prosedur tersimpan, lalu panggil saja untuk menjalankannya.

-Contoh SQL Stored Procedures:

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

➤ SQL Comments

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

- Contoh SQL Comments:

--Select all:

```
SELECT * FROM Customers;
```

➤ SQL Operators

- Operator dalam SQL adalah simbol yang digunakan untuk menginstruksi program untuk melakukan sesuatu. Akan berbeda definisi ketika kita membicarakan operator dalam kehidupan sehari-hari. Banyak sekali operator yang bisa digunakan ketika kita ingin menuliskan query.

Disini kita akan membahas satu persatu operator dasar yang bisa kita gunakan dalam menuliskan query.

➤ SQL Database

- DATABASE digunakan untuk membuat database SQL baru.

- Contoh SQL Database:

```
CREATE DATABASE databasename;
```

➤ SQL Create DB

- DROP DATABASE digunakan untuk menjatuhkan database SQL yang ada.

- Contoh SQL Create DB:

```
DROP DATABASE databasename;
```

➤ SQL Drop DB

-DROP DATABASE digunakan untuk menjatuhkan database SQL yang ada.

- Contoh SQL Drop DB:

```
DROP DATABASE databasename
```

➤ SQL Backup DB

- CADANGAN DATABASE digunakan di SQL Server untuk membuat cadangan penuh dari database SQL yang ada.

-Contoh SQL Backup DB:
BACKUP DATABASE *dbname*
TO DISK = '*filepath*';

➤ SQL Create Table

- CREATE TABLE digunakan untuk membuat tabel baru dalam database.
- Contoh SQL Create Table:

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

➤ SQL Drop Table

- DROP TABLE digunakan untuk menjatuhkan tabel yang ada dalam database.
- Contoh SQL Drop Table:

```
DROP TABLE table_name;
```

➤ SQL Alter Table

- ALTER TABLE digunakan untuk menambah, menghapus, atau memodifikasi kolom dalam tabel yang sudah ada.

-Contoh SQL Alter Table:

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

➤ SQL Constraints

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

- Contoh SQL Constraints:

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

➤ SQL Not Null

- Secara default, kolom dapat menampung nilai NULL.

Batasan NOT NULL memaksa kolom untuk NOT menerima nilai NULL. Ini memaksa bidang untuk selalu berisi nilai, yang berarti Anda tidak dapat menyisipkan catatan baru, atau memperbarui catatan tanpa menambahkan nilai ke bidang ini.

-Contoh SQL Not Null:

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

➤ SQL Unique

- Batasan 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 SQL Unique:

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

➤ SQL Primary Key

- Batasan 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; dan dalam tabel- CREATE TABLE Persons (

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


➤ SQL Foreign Key

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

-Contoh SQL Foreign Key

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

➤ SQL Check

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

-Contoh SQL Check:

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

➤ SQL Default

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

- Contoh SQL Default:

```
ALTER TABLE Persons  
ALTER COLUMN City DROP DEFAULT;
```

➤ SQL Index

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

-Contoh SQL Index:

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

➤ SQL Auto Increment

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

-Contoh SQL Auto Increment:

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

➤ SQL Dates

- Bagian tersulit saat bekerja dengan tanggal adalah memastikan bahwa format tanggal yang Anda coba masukkan, cocok dengan format kolom tanggal dalam database.

-Contoh SQL Dates :

```
SELECT * FROM Orders WHERE OrderDate='2008-11-11'
```

➤ SQL Views

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

- Contoh SQL Views

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

➤ SQL Injection

-Injeksi SQL adalah teknik injeksi kode yang mungkin 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 SQL Injection:

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

- **SQL Hosting**
 - Jika Anda ingin situs web Anda dapat menyimpan dan mengambil data dari database, server web Anda harus memiliki akses ke sistem database yang menggunakan bahasa SQL. Jika server web Anda di-host oleh Penyedia Layanan Internet (ISP), Anda harus mencari paket hosting SQL. Database hosting SQL yang paling umum adalah MS SQL Server, Oracle, MySQL, dan MS Access.
- **SQL Data Types**
 - Setiap kolom dalam tabel database harus memiliki nama dan tipe data. Pengembang SQL harus memutuskan jenis data apa yang akan disimpan di dalam setiap kolom saat membuat tabel. Tipe data adalah pedoman bagi SQL untuk memahami tipe data apa yang diharapkan di dalam setiap kolom, dan juga mengidentifikasi bagaimana SQL akan berinteraksi dengan data yang disimpan.

SQL References

- **SQL Keywords**
 - Referensi kata kunci SQL ini berisi kata-kata yang dicadangkan dalam SQL.
- **MySQL Functions**
 - MySQL memiliki banyak fungsi bawaan. Referensi ini berisi string, numerik, tanggal, dan beberapa fungsi lanjutan di MySQL.
- **SQL Server Functions**
 - SQL Server memiliki banyak fungsi bawaan. Referensi ini berisi string, numerik, tanggal, konversi, dan beberapa fungsi lanjutan di SQL Server.
- **MS Access Functions**
 - MS Access memiliki banyak fungsi bawaan. Referensi ini berisi fungsi string, numerik, dan tanggal di MS Access.

Kesimpulan

- SQL adalah bahasa kueri yang digunakan untuk merancang dan membuat struktur basis data terkait dengan kebutuhan pengembangan aplikasi.
- Terdapat beberapa perintah dasar dari penerapan *Structured Query Language*, yaitu DDL, DML, dan DCL.
- Sebenarnya, setiap programmer khususnya dalam bidang *backend* diharuskan untuk menguasai SQL, namun terdapat beberapa pekerjaan khusus berkaitan dengan penggunaan bahasa ini, yaitu database administrator, data scientist, dan big data architect.