

MODUL PEMBELAJARAN APLIKASI BASIS DATA
STRUCTURED QUERY LANGUAGE (SQL)
EDISI KE - 1

Disusun Oleh:

AGUNG BAITUL HIKMAH
MIFTAH FARID ADIWISASTRA

MANAJEMEN INFORMATIKA
AMIK BSI TASIKMALAYA

2016

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, yang telah memberikan rahmat dan hidayahnya sehingga modul ini dapat terselesaikan dengan baik. Selanjutnya modul ini disusun untuk memberikan gambaran bagi Mahasiswa yang mempelajari Aplikasi Basis Data dengan menggunakan Aplikasi Basis Data Structured Query Language (SQL). Dan mudah-mudahan dapat mempermudah dalam mempelajari SQL karena modul ini disertai dengan contoh kasus dan penggunaan SQL dalam database.

Akhir kata penulis berharap semoga modul Aplikasi Basis Data Structured Query Language (SQL) ini dapat dipergunakan sebaik-baiknya dan dapat dijadikan referensi untuk mahasiswa umum yang ingin mempelajari dan memperdalam tentang database. Tak lupa penulis mengucapkan banyak terima kasih kepada semua pihak yang telah membantu dengan tenaga dan pikirannya, terima kasih juga kepada rekan-rekan instruktur, dosen dan semuanya yang tidak bisa disebutkan satu persatu, yang selalu mendukung penulis sehingga modul ini sehingga dapat selesai sesuai yang diinginkan.

Penulis menyadari masih banyak kekurangan dalam penyusunan modul ini. Untuk itu saran dan kritik yang membangun sangat penulis harapkan guna perbaikan dan pengembangan modul ini kedepan.

Tasikmalaya, Januari 2016

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
PERTEMUAN 1 PENDAHULUAN	5
A. MySQL	5
B. Keunggulan	6
C. Kekurangan	6
PERTEMUAN 2 STRUCTURED QUERY LANGUAGE(SQL)	7
PERTEMUAN 3 DATA DEFINITION LANGUAGE (DCL)	8
A. Create	8
B. Alter	9
C. Drop	9
PERTEMUAN 4 DATA MANIPULATION LANGUAGE (DDL).....	10
A. Insert	10
B. Select	11
C. Update	12
D. Delete	13
PERTEMUAN 5 DATA CONTROL LANGUAGE (DCL).....	15
A. Grant	15
B. Revoke	15
PERTEMUAN 6 EKSPLORASI PERINTAH SQL	16
A. Where	16
B. Operator Logika And, Or, dan Not	17
C. Between dan Not Between	19
D. Like dan Not Like	19
E. In dan Not In	22
F. Order By	23
G. Select Distinct	24
H. Group By	25
I. Fungsi Agreragat	26

J. Alter dan Drop Table	29
K. Operator Aritmatika	30
L. Aliases	31
M. Exists dan Not Exists	33
N. Having	34
O. Union	35
P. View	36
Q. Any Dan All	37
R. Constraint	37
PERTEMUAN 7 OPERASI JOIN	40
A. Inner Join	40
B. Left Outer Join	41
C. Right Outer Join	43
D. Full Join	44
DAFTAR PUSTAKA	46

PERTEMUAN 1

PENDAHULUAN

A. MySQL

MySQL merupakan DBMS yang pertama kali mulai dikembangkan tahun 1994 oleh sebuah perusahaan software bernama TcX Data Konsult AB yang dikemudian hari berganti label menjadi MySQL-AB. “My” pada kata MySQL sebenarnya bukan berarti MY dalam bahasa Inggris, tetapi konon merupakan nama putri dari Michel Widenius seorang perintis dari MySQL. Versi lain menyebutkan “My” adalah kependekan dari “Monty”, yang merupakan julukan untuk Michel Widenius.

MySQL adalah Sebuah program database server yang mampu menerima dan mengirimkan datanya sangat cepat, multi user serta menggunakan perintah dasar SQL (Structured Query Language).MySQL merupakan dua bentuk lisensi, yaitu FreeSoftware dan Shareware. MySQL yang biasa kita gunakan adalah MySQL FreeSoftware yang berada dibawah Lisensi GNU/GPL (General Public License). MySQL Merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya . Selain database server, MySQL juga merupakan program yang dapat mengakses suatu database MySQL yang berposisi sebagai Server, yang berarti program kita berposisi sebagai Client. Jadi MySQL adalah sebuah database yang dapat digunakan sebagai Client maupun server. Database MySQL merupakan suatu perangkat lunak database yang berbentuk database relasional atau disebut Relational Database Management System (RDBMS) yang menggunakan suatu bahasa permintaan yang bernama SQL (Structured Query Language).

Sebagaimana database sistem yang lain, dalam SQL juga dikenal hierarki server dengan database-database.Tiap-tiap database memiliki tabel-tabel. Tiap-tiap tabel memiliki field-field. Umumnya informasi tersimpan dalam tabel – tabel yang secara logic merupakan struktur 2 dimensi terdiri atas baris dan kolom.Field-field tersebut dapat berupa data seperti int , realm char, date, time dan lainnya. SQL tidak memiliki fasilitas pemrograman yang lengkap,tidak ada looping ataupun percabangan,misalnya.Sehingga untuk menutupi kelemahan ini perlu digabung dengan bahasa pemrograman semisal C.

B. Keunggulan

Database MySQL memiliki beberapa keunggulan disbanding database lain, diantaranya:

1. MySQL merupakan Database Management System (DBMS)
2. MySQL sebagai Relation Database Management System (RDBMS) atau disebut dengan database Relational
3. MySQL Merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya dan bisa di download di www.mysql.com.
5. MySQL merupakan program yang multi-threaded, sehingga dapat dipasang pada server yang memiliki multi-CPU.
6. Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, TCL APIs dls.
7. Bekerja pada berbagai platform.(tersedia berbagai versi untuk berbagai sistem operasi).
8. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem database.
9. Memiliki sistem sekuriti yang cukup baik dengan verifikasi host.
10. Mendukung ODBC untuk sistem operasi Microsoft Windows.
11. Mendukung record yang memiliki kolom dengan panjang tetap atau panjang bervariasi. dan masih banyak keunggulan lainnya (lihat manual mysql_3.22)
12. MySQL dan PHP saling terintegrasi. Maksudnya adalah pembuatan database dengan menggunakan sintak PHP dapat di buat. Sedangkan input yang di masukkan melalui aplikasi web yang menggunakan script server-side seperti PHP dapat langsung dimasukkan ke database MySQL yang ada di server dan tentunya web tersebut berada di sebuah web server.

C. Kekurangan

Penggunaan database MySQL memiliki kekurangan sebagai berikut:

1. Banyak mengklaim kurang support terhadap pemrograman Visual/Desktop, sehingga sedikit yang menggunakan untuk aplikasi visual.
2. Karena berlisensi GPL sehingga sulit mendapatkan update untuk problem yang urgent, sehingga perusahaan skala menengah keatas lebih memilih RDBMS berlisensi dan disupport seperti Oracle dan MS SQL Server
3. Sangat diragukan dalam menangani data skala besar, karena ada beberapa opini yang pro dan kontra terhadap kemampuan MySQL terhadap pengolahan data yang besar.

PERTEMUAN 2

STRUCTURED QUERY LANGUAGE (SQL)

Structured Query Language (SQL) merupakan komponen bahasa relational database system. SQL merupakan bahasa baku (ANSI/SQL), non procedural, dan berorientasi himpunan (set-oriented language). SQL dapat digunakan baik secara interaktif atau ditempelkan (embedded) pada sebuah program aplikasi. SQL dapat digunakan untuk mendefinisikan struktur data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (security), hingga pemeliharaan kinerja basis data.

Secara umum, bahasa SQL memiliki beberapa bagian penting, yaitu:

1. Data Definition Language (DDL)

DDL menyediakan perintah-perintah untuk mendefinisikan skema relasi, menghapus relasi, serta memodifikasi skema relasi.

2. Data Manipulation Language (DML)

DML mencakup bahasa SQL untuk menyisipkan rekaman pada relasi, menghapus rekaman pada relasi, serta memodifikasi rekaman pada relasi.

3. Data Control Language (DCL)

DCL adalah sub bahasa SQL yang berfungsi untuk melakukan pengontrolan data dan server databasenya, seperti manipulasi user dan hak akses (priviledges). Yang termasuk perintah dalam DCL ada dua, yaitu GRANT dan REVOKE.

PERTEMUAN 3

DATA DEFINITION LANGUAGE (DDL)

Data Definition Language (DDL) merupakan perintah SQL yang digunakan untuk melakukan definisi awal suatu basis data dan tabel pada konsep RDBMS. Secara sederhana, penulisan perintah SQL pada kelompok ini terdiri dari Create, Alter, dan Drop. Berikut merupakan pembahasan untuk perintah SQL tersebut.

A. Create

Perintah SQL ini digunakan untuk membuat suatu basis data dan table pendukung didalam pembangunan basis data tersebut. Table merupakan objek yang sangat penting dalam suatu basis data.

1. Sintak untuk membuat database baru :

```
CREATE DATABASE databasename;
```

2. Sintak untuk membuat table baru:

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

3. Contoh membuat database baru:

```
CREATE DATABASE testDB;
```

4. Contoh membuat table baru:

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


B. Alter

Perintah SQL ini digunakan untuk mengubah struktur table yang terdapat didalam basis data. Hal ini dapat saja terjadi, jika ingin melakukan penambahan atau penghapusan suatu field atau atribut key (Primary/Foreign), tanpa mendefinisikan struktur baru pada table. Melalui perintah SQL ini jika sudah terisikan record pada table tersebut, maka mengenai hilangnya record yang terdapat pada suatu table tidak akan terjadi. Berikut ini merupakan aturan umum penulisan yang digunakan pada alter table.

1. Sintak menambah kolom baru:

ALTER TABLE table_name ADD column_name datatype;

2. Sintak menghapus kolom:

ALTER TABLE table_name DROP COLUMN column_name;

3. Sintak merubah kolom:

ALTER TABLE table_name ALTER COLUMN column_name datatype;

4. Contoh penerapan dan penjelasan:

Menambahkan kolom alamat pada table mahasiswa dengan tipe data varchar panjang field 25

ALTER TABLE mahasiswa ADD alamat varchar(25);

5. Menghapus kolom alamat pada table mahasiswa:

ALTER TABLE mahasiswa DROP COLUMN alamat;

C. Drop

Perintah SQL ini digunakan untuk menghapus table yang terdapat didalam basis data. Berikut ini merupakan aturan umum penulisan yang digunakan pada drop table.

- a. Sintak menghapus database:

DROP DATABASE databasename;

- b. Sintak menghapus table:

DROP TABLE table_name;

- c. Contoh menghapus database testDB:

DROP DATABASE testDB;

- d. Contoh menghapus table mahasiswa:

DROP TABLE mahasiswa;

PERTEMUAN 4

DATA MANIPULATION LANGUAGE(DML)

Data Manipulation Language (DML) merupakan perintah SQL yang digunakan untuk melakukan pengolahan record atau memanipulasi data pada table dalam suatu basis data. Secara sederhana penulisan perintah SQL pada kelompok ini terdiri dari insert, select, update, dan delete. Berikut ini adalah deskripsi mengenai kelompok perintah DML tersebut.

A. Insert

Perintah SQL ini digunakan untuk melakukan entry atau penambahan suatu record pada table dalam basis data. Berikut ini merupakan aturan umum penulisan yang digunakan pada insert suatu table.

1. Sintak pertama:

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

2. Sintak kedua:

```
INSERT INTO table_name VALUES (value1, value2, value3, ...);
```

3. Contoh penerapan sintak pertama:

- a. Menambahkan satu record pada tabel customer

CustomerName	ContactName	Address	City	PostalCode	Country
White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland

```
INSERT INTO Customers (CustomerName, ContactName, Address,  
City,  
PostalCode,  
Country)  
VALUES ('Cardinal', 'Erichsen', 'Skagen', 'Stavanger', '4006',  
'Norway');
```

Hasilnya:

CustomerName	ContactName	Address	City	PostalCode	Country
White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
Cardinal	Tom B. Erichsen	Skagen 21	Stavanger	4006	Norway

B. Select

Perintah SQL ini digunakan untuk memilih record yang akan ditampilkan berdasarkan data pada table dalam basis data. Berikut ini merupakan aturan umum dan variasi penulisan yang digunakan pada select.

1. Sintak untuk menampilkan salah satu kolom yang ada ditabel:

SELECT column1, column2, ...FROM table_name;

2. Sintak untuk menampilkan semua kolom yang ada ditabel:

SELECT * FROM table_name;

3. Contoh penerapan

- a. Menampilkan CustomerName dan City pada table customer:

SELECT CustomerName,City FROM Customers;

CustomerName	City
Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå
Blauer See Delikatessen	Mannheim
Blondel père et fils	Strasbourg
Bólido Comidas preparadas	Madrid

- b. Contoh menampilkan seluruh kolom yang ada di table Customer:

```
SELECT * FROM customer;
```

CustomerName	ContactName	Address	City	PostalCode
Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209
Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021
Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023
Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP
Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22
Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306
Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000

C. Update

Perintah SQL ini untuk mengubah data dalam suatu table pada field tertentu dengan record baru berdasarkan suatu field sebagai kriteria pengubahan record-nya.

1. Sintak perintah update:

```
UPDATE table_name SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

2. Contoh penerapan

- a. Merubah ContactName= 'Maria Anders' menjadi 'Alfred Schmidt' dan City= 'Frankfurt'

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

```
UPDATE Customers SET ContactName = 'Alfred Schmidt',  
City= 'Frankfurt' WHERE CustomerID = 1;
```

Hasilnya:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Alfred Schmidt	Obere Str. 57	Frankfurt	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

- b. Merubah dua record sekaligus yaitu merubah yang Country='Mexico' dirubah ContactName menjadi 'Juan'

```
UPDATE Customers                                SET ContactName='Juan'
WHERE Country='Mexico';
```

Hasilnya:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Alfred Schmidt	Obere Str. 57	Frankfurt	12209	Germany
2	Ana Trujillo Emparedados y helados	Juan	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Juan	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

D. Delete

Perintah SQL ini digunakan untuk menghapus data dalam suatu table, berdasarkan suatu field sebagai kriteria penghapusan record-nya. Berikut ini merupakan aturan umum penulisan yang digunakan pada delete sebagai berikut :

1. Sintak perintah untuk menghapus salah satu record:

```
DELETE FROM table_name WHERE condition;
```

2. Sintak untuk menghapus seluruh record pada tabel:

```
DELETE FROM table_name;
```

3. Contoh penerapan

- a. Menghapus CustomerName= 'Alfreds Futterkiste' dari tabel customer

```
DELETE FROM Customers      WHERE CustomerName='Alfreds  
Futterkiste';
```

- b. Menghapus seluruh record yang ada di tabel customer

```
DELETE FROM mahasiswa;
```

PERTEMUAN 5

DATA CONTROL LANGUAGE(DCL)

Data Control Language (DCL) merupakan perintah SQL yang digunakan untuk melakukan pengaturan hak akses suatu objek data para pengguna basis data. Secara sederhana, penulisan perintah SQL pada kelompok ini terdiri dari grant, revoke.

A. Grant

Perintah SQL ini digunakan oleh seorang administrator basis data untuk memberikan hak aksesnya kepada pengguna tertentu, agar dapat mengakses suatu table dalam basis data. Hak akses tersebut adalah insert, delete, update, dan select. Berikut ini merupakan aturan umum penulisan yang digunakan pada Grant.

1. Sintak :

GRANT *hak_akses* ON *nama_tabel* TO *pengguna_tertentu*;

2. Contoh penerapan dan penjelasan:

- Pemberian hak akses INSERT dan DELETE pada tabel mahasiswa kepada Budi

GRANT *INSERT, DELETE* ON *mahasiswa* TO *Budi*;

- Pemberian hak akses UPDATE pada tabel mahasiswa kepada Raisa

GRANT *UPDATE* ON *mahasiswa* TO *Raisa*;

B. Revoke

Perintah SQL ini digunakan oleh seorang administrator basis data, untuk membatalkan / menghentikan hak akses yang telah diberikan kepada pengguna tertentu, agar tidak dapat mengakses table dalam basis data. Berikut ini merupakan aturan umum penulisan yang digunakan pada revoke.

1. Sintak :

REVOKE *hak_akses* ON *nama_tabel* FROM *pengguna_tertentu*;

2. Contoh penerapan dan penjelasan:

- Membatalkan hak akses INSERT dan DELETE pada tabel mahasiswa kepada Budi

REVOKE *INSERT, DELETE* ON *mahasiswa* FROM *Budi*;

- Membatalkan hak akses UPDATE pada tabel mahasiswa kepada Raisa

REVOKE *UPDATE* ON *mahasiswa* FROM *Raisa*;

PERTEMUAN 6

EKSPLORASI PERINTAH SQL

Setelah pada kajian sebelumnya dilakukan pembahasan untuk tiga jenis perintah SQL, yaitu DDL, DML, dan DCL kemudian pada bab ini dilakukan beberapa eksplorasi dari suatu perintah SQL dengan adanya penggabungan klausa, operator, fungsi agregat, join, dan relasi. Berikut ini merupakan hal umum untuk perintah SQL yang dilakukan saat eksplorasi tersebut.

A. Where

Where ini digunakan untuk melakukan seleksi pada record, yang sesuai dengan syarat suatu kriteria pada suatu kondisi perintah SQL. Pada table 6.1 disajikan operator matematika yang digunakan oleh klausa where.

Tabel 6.1 Operator Matematika

NO	OPERATOR	ARTI
1	=	Sama dengan
2	<>	Tidak sama dengan
3	<	Lebih kecil
4	<=	Lebih kecil sama dengan
5	>	Lebih besar
6	>=	Lebih besar sama dengan

1. Sintak perintah Where dengan Select:

SELECT *column1*, *column2*,...FROM *table_name*
WHERE *condition*;

2. Contoh penerapan

- a. Menampilkan Country='Mexico' dari tabel customer

SELECT * FROM Customers WHERE Country='Mexico';

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
13	Centro comercial Moctezuma	Francisco Chang	Sierras de Granada 9993	México D.F.	05022	Mexico
58	Pericles Comidas clásicas	Guillermo Fernández	Calle Dr. Jorge Cash 321	México D.F.	05033	Mexico

- b. Menampilkan CostumerID=1 dari tabel customer

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Alfred Schmidt	Obere Str. 57	Frankfurt	12209	Germany

Penggunaan WHERE juga bisa digunakan dalam perintah UPDATE dan DELETE.

B. Operator Logika And , Or Dan Not

Operator logika and dan or digunakan untuk menggabungkan seleksi pada record yang syaratnya lebih dari satu kondisi. Pada operator logika, not digunakan untuk negasi dari kondisi tersebut. Pada table 6.2 disajikan untuk interpretasi hubungan dengan dua kondisi (K1 dan K2).

Tabel 6.2 Operator Logika

K1	K2	NOT K1	K1 AND K2	K1 OR K2
T	T	F	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	F

Keterangan T = Jika kondisi benar (True)

F = Jika kondisi salah (False)

- Sintak operator:

- AND

```
SELECT column1, column2,...FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

- OR

```
SELECT column1, column2,...FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

- NOT

```
SELECT column1, column2,...FROM table_name
WHERE NOT condition;
```

2. Contoh penerapannya:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

a. Menampilkan customer city='Berlin' dan Country='Germany'

```
SELECT * FROM Customers WHERE Country='Germany' AND City='Berlin';
```

Hasilnya:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany

b. Menampilkan customer city='London' atau Country='Mexico'

```
SELECT * FROM Customers WHERE Country='Mexico' AND City='London';
```

Hasilnya:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK

c. Menampilkan costomer yang country nya bukan jerman

```
SELECT * FROM Customers WHERE NOT Country='Germany';
```

Hasilnya:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taqueria	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

C. Between Dan Not Between

Between digunakan untuk mengolah data suatu nilai dalam range tertentu. Not between merupakan negasinya, yaitu mengolah suatu nilai diluar dari range yang telah ditentukan.

1. Sintak perintah Between:

```
SELECT column_name(s) FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

2. Contoh penerapannya:

- a. Menampilkan semua record dari tabel produk dengan harga 10000 sampai dengan 20000

```
SELECT * FROM produk
WHERE harga BETWEEN 10000 AND 20000;
```

- b. Menampilkan semua record dari tabel produk dengan harga tidak antara 10000 sampai dengan 20000

```
SELECT * FROM produk
WHERE harga NOT BETWEEN 10000 AND 20000;
```

- c. Menampilkan semua mahasiswa dari tabel mahasiswa antara nama 'Budi' dan Raisa diurutkan berdasarkan nama:

```
SELECT * FROM mahasiswa
WHERE nama NOT BETWEEN 'Budi' AND 'Raisa'
ORDER BY nama;
```

D. Like Dan Not Like

Like digunakan untuk mencari suatu teks yang sesuai berdasarkan kata depan (prefix), kata tengah (infix), kata akhir (suffix). Not like merupakan pernyataan negasinya, yaitu mencari suatu teks yang tidak sesuai dengan kriteria like. Pada table 6.3 disajikan

mengenai pola yang digunakan oleh operator like dan not like untuk mencari kesesuaian kata. Pada operator ini terdapat penggunaan tanda persen (%) untuk mewakili satu atau lebih kecocokan karakter, dan tanda underscore (_) untuk mewakili satu saja kecocokan karakter yang dicari.

Tabel 6.3 Pola penggunaan Percent (%) dan Underscore (_)

NO	POLA	PENJELASAN
1	a%	Cara untuk mencari kesesuaian pada suatu kata yang berawalan 1 atau beberapa huruf a atau A
2	%a%	Cara untuk mencari kesesuaian pada suatu kata yang didalamnya mengandung 1 atau beberapa huruf a atau A
3	%a	Cara untuk mencari kesesuaian pada suatu kata yang berakhiran 1 atau beberapa huruf a atau A
4	a_	Cara untuk mencari kesesuaian pada suatu kata yang berawalan hanya 1 huruf a atau A yang sama
5	_a_	Cara untuk mencari kesesuaian pada suatu kata yang didalamnya mengandung hanya 1 huruf a atau A yang sama
6	_a	Cara untuk mencari kesesuaian pada suatu kata yang berakhiran hanya 1 huruf a atau A yang sama

1. Sintak perintah Like:

```
SELECT column1,column2,...FROM table_name
WHERE columnN LIKE pattern;
```

2. Contoh penerapan:

- a. Menampilkan nama costumer yang berawalan huruf a:

```
SELECT * FROM Customers WHERE CustomerName LIKE 'a%';
```

CustomerName	ContactName	Address	City
Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin
Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.
Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.
Around the Horn	Thomas Hardy	120 Hanover Sq.	London

- b. Menampilkan nama customer yang ada huruf or diawal, ditengah ataupun diakhir

```
SELECT * FROM Customers WHERE CustomerName LIKE '%or%';
```

CustomerName	ContactName	Address	City
Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.
Around the Horn	Thomas Hardy	120 Hanover Sq.	London
Hungry Coyote Import Store	Yoshi Latimer	City Center Plaza 516 Main St.	Elgin
La corne d'abondance	Daniel Tonini	67, avenue de l'Europe	Versailles
Lazy K Kountry Store	John Steel	12 Orchestra Terrace	Walla Walla
Morgenstern Gesundkost	Alexander Feuer	Heerstr. 22	Leipzig
North/South	Simon Crowther	South House 300 Queensbridge	London

- c. Menampilkan nama customer yang mempunyai huruf kedua 'r'

```
SELECT * FROM Customers WHERE CustomerName LIKE '_r%';
```

CustomerName	ContactName	Address	City
Around the Horn	Thomas Hardy	120 Hanover Sq.	London
Drachenblut Delikatessend	Sven Ottlieb	Walserweg 21	Aachen
Ernst Handel	Roland Mendel	Kirchgasse 6	Graz
Frankenversand	Peter Franken	Berliner Platz 43	München

3. Sintak Not Like:

```
SELECT column1,column2,...FROM table_name WHERE column  
NOT LIKE pattern;
```

4. Contoh penerapan:

- a. Menampilkan nama customer yang bukan berawalan huruf a:

```
SELECT * FROM Customers  
WHERE CustomerName NOT LIKE 'a%';
```

CustomerName	ContactName	Address	City
Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå
Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim
Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg
Bólido Comidas preparadas	Martín Sommer	C/ Araquil, 67	Madrid
Bon app'	Laurence Lebihans	12, rue des Bouchers	Marseille
Bottom-Dollar Marketse	Elizabeth Lincoln	23 Tsawassen Blvd.	Tsawassen

E. In Dan Not In

In digunakan untuk mencari nilai kesesuaian dengan suatu data tertentu yang ada dalam suatu daftar / list nilai. Operator Not In merupakan negasi untuk nilai yang tidak ada dalam daftar / list tersebut.

1. Sintak In:

```
SELECT column_name(s) FROM table_name
WHERE column_name IN (value1, value2, ...);
```

Atau:

```
SELECT column_name(s) FROM table_name
WHERE column_name IN (SELECT STATEMENT);
```

2. Contoh penerapan:

- a. Memilih semua customer yang berada di Jerman dan Perancis

```
SELECT * FROM Customers WHERE Country IN ('Germany',
'France');
```

CustomerName	ContactName	Address	City	PostalCode	Country
Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany
Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France
Bon app'	Laurence Lebihans	12, rue des Bouchers	Marseille	13008	France
Drachenblut Delikatessend	Sven Ottlieb	Waiseweg 21	Aachen	52066	Germany
Du monde entier	Janine Labruno	67, rue des Cinquante Otages	Nantes	44000	France

- b. Memilih semua costumer yang berasal dari negara yang sama dengan supplier

```
SELECT * FROM Customers WHERE Country IN (SELECT Country
FROM Suppliers);
```

CustomerName	ContactName	Address	City	PostalCode	Country
Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany
Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France
Bólido Comidas preparadas	Martin Sommer	C/ Araquil, 67	Madrid	28023	Spain
Bon app'	Laurence Lebihans	12, rue des Bouchers	Marseille	13008	France

3. Sintak Not In:

```
SELECT column_name(s) FROM table_name
WHERE column_name NOT IN (value1, value2,...);
```

4. Contoh penerapan:

- a. Memilih semua customer yang bukan dari Jerman dan Perancis

```
SELECT * FROM Customers WHERE Country not IN
('Germany', 'France');
```

CustomerName	ContactName	Address	City	PostalCode	Country
Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

F. Order By

Order By digunakan untuk melakukan pengurutan berdasarkan suatu field, yang merupakan hasil dari pernyataan select secara ascending atau descending. Default pengurutan klausa order by adalah ascending.

1. Sintak Order By:

```
SELECT column1, column2,... FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

2. Contoh penerapan

- a. Menampilkan seluruh record dari tabel customer dan diurutkan berdasarkan nama customer

```
SELECT * FROM Customers ORDER BY Country;
```

CustomerName	ContactName	Address	City	PostalCode
Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209
Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021
Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023
Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP
B's Beverages	Victoria Ashworth	Fauntleroy Circus	London	EC2 5NT
Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22

- b. Menampilkan seluruh record dari tabel customer dan diurutkan berdasarkan nama customer dengan urutan z – a

```
SELECT * FROM Customers ORDER BY CustomerName DESC;
```

CustomerName	ContactName	Address	City	PostalCode
Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012
Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240
White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128
Wellington Importadora	Paula Parente	Rua do Mercado, 12	Resende	08737-363
Wartian Herkku	Pirkko Koskitalo	Torikatu 38	Oulu	90110
Vins et alcools Chevalier	Paul Henriot	59 rue de l'Abbaye	Reims	51100
Victuailles en stock	Mary Saveley	2, rue du Commerce	Lyon	69004

G. Select Distinct

Distinct digunakan untuk menghilangkan duplikasi pada suatu tampilan, dengan cara menghilangkan tampilan record dengan nilai yang sama untuk hasil suatu perintah SQL, tetapi bukan menghapus record secara fisik.

1. Sintak Distinct

```
SELECT DISTINCT column1, column2,...FROM table_name;
```

2. Contoh penerapan

- a. Menampilkan country yang berbeda dari tabel customer

```
SELECT DISTINCT Country FROM Customers;
```

Country
Germany
Mexico
UK
Sweden
France
Spain

- b. Menampilkan jumlah negara yang berbeda dari tabel customer


```
SELECT COUNT(DISTINCT Country) FROM Customers;
```

COUNT(DISTINCT Country)
21

H. Group By

Group By digunakan untuk melakukan pengelompokan berdasarkan suatu field atau beberapa field.

1. Sintak group by

```
SELECT column_name(s)                FROM table_name
WHERE condition                       GROUP BY column_name(s)
ORDER BY column_name(s);
```

2. Contoh penerapan

- a. Menampilkan jumlah customer di setiap Negara

```
SELECT COUNT(CustomerID), Country FROM Customers GROUP
BY Country;
```

COUNT(CustomerID)	Country
3	Argentina
2	Austria
2	Belgium
9	Brazil
3	Canada

- b. Menampilkan jumlah customer dari setiap Negara diurutkan dari bawah z- a.

```
SELECT COUNT(CustomerID), Country FROM Customers GROUP
BY Country ORDER BY COUNT(CustomerID) DESC;
```

COUNT(CustomerID)	Country
13	USA
11	France
11	Germany
9	Brazil

I. Fungsi Agregat

Kelompok fungsi ini digunakan untuk suatu nilai, dengan kualifikasi yang merupakan hasil dari perhitungan sekelompok baris data atau table. Pada table 6.4, disajikan fungsi agregat yang memiliki kualifikasi tersendiri untuk menghasilkan suatu nilai tertentu.

Tabel 6.4 Kelompok Fungsi Agregat

NO	NAMA FUNGSI	KUALIFIKASI
1.	AVG	Menghasilkan nilai rata-rata
2	COUNT	Menghitung banyaknya data/record
3	MAX	Menghasilkan nilai terbesar
4	MIN	Menghasilkan nilai terkecil
5	SUM	Menghitung jumlah sekelompok data

1. AVG

a. Sintak

```
SELECT AVG(column_name)FROM table_name WHERE condition;
```

b. Contoh penerapan

- Mencari rata-rata jumlah beli(quantity) dari tabel orderdetail

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40

```
SELECT AVG(Quantity) FROM OrderDetails;
```

Hasilnya:

AVG(Quantity)
24.6003861003861

2. COUNT

a. Sintak

```
SELECT COUNT(column_name)FROM table_name  
WHERE condition;
```

b. Contoh penerapan

- Menghitung jumlah record dari tabel produk

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35
6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25
7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pks.	30

```
SELECT COUNT(ProductID) FROM Products;
```

Hasilnya:

COUNT(ProductID)
7

3. MAX

a. Sintak

```
SELECT MAX(column_name)FROM table_name;
```

b. Contoh penerapan

- Mencari Quantity terbesar dari tabel orderdetail

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40

```
SELECT MAX(Quantity) FROM OrderDetails;
```

Hasilnya:

MAX(Quantity)
40

4. MIN

a. Sintak

```
SELECT MAX(column_name)FROM table_name;
```

b. Contoh penerapan

- Mencari Quantity terkecil dari tabel orderdetail

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40

```
SELECT MAX(Quantity) FROM OrderDetails;
```

Hasilnya:

MIN(Quantity)
5

5. SUM

a. Sintak

```
SELECT SUM(column_name)FROM table_name;
```

b. Contoh penerapan

- Mencari jumlah (Quantity) dari tabel OrderDetail

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40

```
SELECT SUM(Quantity) FROM OrderDetails
```

Hasilnya:

SUM(quantity)
76

J. Alter Dan Drop Tabel

Klausa alter table digunakan untuk melakukan modifikasi struktur table. Umumnya, klausa ini dilakukan pada suatu kegiatan saat mengubah nama field sebuah table. Selain pengubahan struktur table, seorang administrator basis data juga melakukan penghapusan langsung pada table yang diinginkan, dalam menggunakan klausa drop table.

1. ALTER TABEL

- Sintak menambahkan kolom(field)

ALTER TABLE *table_name* ADD *column_name* *datatype*;

- Sintak menghapus kolom(field)

ALTER TABLE *table_name* DROP COLUMN *column_name*;

- Sintak modifikasi kolom(field)

ALTER TABLE *table_name* MODIFY COLUMN *column_name* *datatype*;

- Contoh penerapannya

- Menambahkan Field(kolom) “DateOfBirth” pada tabel person dengan tipe data date;

ID	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

ALTER TABLE Persons ADD DateOfBirth date;

Hasilnya:

ID	LastName	FirstName	Address	City	DateOfBirth
1	Hansen	Ola	Timoteivn 10	Sandnes	
2	Svendson	Tove	Borgvn 23	Sandnes	
3	Pettersen	Kari	Storgt 20	Stavanger	

- Menghapus Field(Kolom) DateOfBirth dari tabel person

Hasilnya:

ID	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

- Modifikasi kolom DateOfBirth tipe datanya menjadi varchar(20)

```
ALTER TABLE Persons          ALTER COLUMN DateOfBirth
varchar(20);
```

2. DROP TABEL

a. Sintak

DROP TABLE *table_name*;

b. Contoh penerapan

- Menghapus tabel Shippers

```
DROP TABLE Shippers;
```

K. Operator Aritmatika

Pada umumnya operator aritmatika untuk proses perhitungan yang digunakan dalam perintah SQL terdiri dari perkalian(*), pembagian(/), penjumlahan(+), dan pengurangan(-).

1. Operator perkalian (*)

Contoh:

```
SELECT 30 * 20;
```

Hasilnya:

30 * 20
600

2. Operator Diva atau hasil bagi ($/$)

Contoh:

```
SELECT 30 / 10;
```

Hasilnya:

30 / 10
3

3. Operator pengurangan (-)

Contoh:

```
SELECT 30 - 10;
```

Hasilnya:

30 - 20
10

4. Operator Penjumlahan (+)

Contoh:

```
SELECT 30 + 10;
```

Hasilnya:

30 + 20
50

5. Operator Mod atau Sisa bagi (%)

Contoh:

```
SELECT 17 % 5;
```

Hasilnya:

17 % 5
2

L. Aliases

Alias secara sederhana adalah "nama lain", artinya kita dapat mengganti sementara sebuah nama tabel dan nama kolom dengan nama yang sesuai dengan keinginan kita. Tujuan penggunaan alias adalah untuk mempermudah dan menyingkat deskripsi sebuah nama tabel dan nama kolom pada sebuah query.

1. Sintak

a. Mengganti nama kolom

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

b. Mengganti nama tabel

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

2. Contoh penerapan

- a. Mengganti kolom ProductID menjadi ID, dan kolom ProductName menjadi Product dalam tabel Product.

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35
6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25
7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	30

SELECT ProductID AS ID, ProductName AS Product FROM Products;

Hasilnya:

ID	Product
1	Chais
2	Chang
3	Aniseed Syrup
4	Chef Anton's Cajun Seasoning
5	Chef Anton's Gumbo Mix
6	Grandma's Boysenberry Spread
7	Uncle Bob's Organic Dried Pears

- b. Contoh penerapan alias pada tabel

SELECT Orders.OrderID, Orders.OrderDate, Customers.CustomerName FROM Customers, Orders WHERE Customers.CustomerName="Around the Horn" AND Customers.CustomerID=Orders.CustomerID;

Hasilnya:

OrderID	OrderDate	CustomerName
10355	1996-11-15	Around the Horn
10383	1996-12-16	Around the Horn

M. Exists Dan Not Exist

Perintah **Exists** biasanya digunakan untuk *check* keberadaan suatu obyek di suatu database. Operator EXISTS mengembalikan nilai true jika subquery mengembalikan satu atau beberapa record. Kata kunci EXIST dan NOT EXIST dirancang hanya untuk digunakan di subquery. Kata kunci-kata kunci ini menghasilkan nilai TRUE atau FALSE. EXIST akan mengirim nilai TRUE jika dan hanya jika terdapat sedikitnya satu baris di table hasil yang dikirim oleh subquery. EXIST mengirim nilai FALSE jika subquery mengirim table kosong. NOT EXIST kebalikan dari EXIST. Karena EXIST dan NOT EXIST hanya memeriksa keberadaan baris-baris di table hasil subquery.

1. Sintak

```
SELECT column_name(s) FROM table_name
WHERE EXISTS(SELECT column_name FROM table_name WHERE conditio
n);
```

2. Contoh penerapan

- a. Mengembalikan TRUE dan mencantumkan Suppliers dengan harga product kurang dari 20.

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35

SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country
1	Exotic Liquid	Charlotte Cooper	49 Gilbert St.	London	EC1 4SD	UK
2	New Orleans Cajun Delights	Shelley Burke	P.O. Box 78934	New Orleans	70117	USA
3	Grandma Kelly's Homestead	Regina Murphy	707 Oxford Rd.	Ann Arbor	48104	USA
4	Tokyo Traders	Yoshi Nagase	9-8 Sekimai Musashino-shi	Tokyo	100	Japan

```
SELECT SupplierName FROM Suppliers WHERE EXISTS (SELECT ProductName
FROM Products WHERE SupplierId = Suppliers.supplierId AND Price <
20);
```

Hasilnya:

SupplierName
Exotic Liquid
New Orleans Cajun Delights
Tokyo Traders
Mayumi's
Pavlova, Ltd.
Specialty Biscuits, Ltd.
PB Knäckebröd AB
Refrescos Americanas LTDA
Heli Süßwaren GmbH & Co. KG
Plutzer Lebensmittelgroßmärkte AG
Formaggi Fortini s.r.l.
Norske Meierier
Bigfoot Breweries
Svensk Sjöföda AB

N. Having

HAVING ditambahkan ke SQL karena kata kunci WHERE tidak dapat digunakan dengan fungsi agregat. HAVING digunakan untuk menentukan kondisi bagi GROUP BY. Kelompok yang memenuhi HAVING saja yang akan dihasilkan.

1. Sintak

```
SELECT column_name(s)FROM table_nameWHERE conditionGROUP BY  
column_name(s) HAVING condition ORDER BY column_name(s);
```

2. Contoh penerapan

a. Menampilkan jumlah customer di setiap negara. Hanya sertakan negara dengan lebih dari 5 customer:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

```
SELECT COUNT(CustomerID), Country FROM Customers GROUP BY  
Country  
HAVING COUNT(CustomerID) > 5;
```

Hasilnya:

COUNT(CustomerID)	Country
9	Brazil
11	France
11	Germany
7	UK
13	USA

O. Union

Operator UNION digunakan untuk menggabungkan hasil-set dari dua atau lebih pernyataan SELECT. Dengan persyaratan sebagai yaitu: Setiap pernyataan SELECT dalam UNION harus memiliki jumlah kolom yang sama, Kolom juga harus memiliki tipe data yang sama, dan Kolom di setiap pernyataan SELECT juga harus berada dalam urutan yang sama.

1. Sintak

```
SELECT column_name(s) FROM table1  
UNION SELECT column_name(s) FROM table2;
```

Atau:

```
SELECT column_name(s) FROM table1  
UNION ALL SELECT column_name(s) FROM table2;
```

2. Contoh penerapan

- Pernyataan SQL berikut memilih semua kota yang berbeda (hanya nilai yang berbeda) dari "Customers" dan "Suppliers":

```
SELECT City FROM Customers UNION SELECT City FROM Suppliers  
ORDER BY City;
```

Hasilnya:

City
Aachen
Albuquerque
Anchorage
Ann Arbor
Annecy
Barcelona
Barquisimeto
Bend
Bergamo

b. Pernyataan SQL berikut memilih semua kota (nilai duplikat juga) dari "Customers" dan "Suppliers":

```
SELECT City FROM Customers UNION ALL SELECT City FROM Suppliers ORDER BY City;
```

City
Aachen
Albuquerque
Anchorage
Ann Arbor
Annecy
Barcelona
Barquisimeto
Bend

P. View

View adalah tabel virtual berdasarkan hasil-set pernyataan SQL. Tampilan berisi baris dan kolom, seperti tabel sebenarnya. Bidang dalam tampilan adalah bidang dari satu atau lebih tabel nyata dalam database. Dalam view dapat menambahkan fungsi SQL, WHERE, dan JOIN statements untuk melihat dan menyajikan data seolah-olah data berasal dari satu tabel tunggal.

1. Sintak

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

2. Contoh penerapan

```
CREATE VIEW [Current ProductList] AS  
SELECT ProductID,ProductName FROM Products  
WHERE Discontinued = No;
```

Q. Any Dan All

Operator Any mengembalikan nilai true jika salah satu dari nilai subkueri memenuhi persyaratan. Operator ALL mengembalikan true jika semua nilai subquery memenuhi kondisi.

1. Sintak

a. Any

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

b. All

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

2. Contoh penerapan

a. Any

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

b. All

```
SELECT ProductName
FROM Products
WHERE ProductID
= ALL (SELECT ProductID FROM OrderDetails WHERE Quantity
= 10);
```

R. Constraint

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

1. Sintak

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

2. Contoh penerapan

a. Not Null

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

b. Unique

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

c. Primary Key

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

d. Foreign Key

```
CREATE TABLE Orders (  
    OrderID int NOT NULL,
```

```

        OrderNumber int NOT NULL,
        PersonID int,
        PRIMARY KEY (OrderID),
        FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)
    );

```

e. Check

```

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

```

f. Default

```

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

```

g. Index

```

CREATE INDEX idx_lastname
ON Persons (LastName);

```

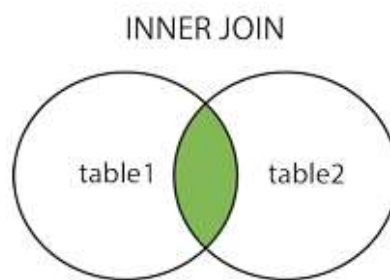
PERTEMUAN 7

OPERASI JOIN

Merupakan operasi yang digunakan untuk menghasilkan kumpulan record dari dua buah table atau lebih. Secara umum operasi join dapat dikelompokkan menjadi:

A. Inner Join

Merupakan operasi join yang digunakan untuk menghasilkan kumpulan record dari dua buah table atau lebih dengan nilai yang sama, nilai tersebut merupakan irisan yang terdapat pada tabel-tabel tersebut.



1. Sintak

```
SELECT column_name(s) FROM table1  
INNER JOIN table2 ON table1.column_name = table2.c  
olumn_name;
```

2. Contoh penerapan

a. Menggabungkan Tabel Orders dan Customer

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taqueria	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

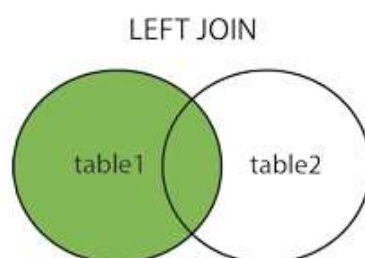
```
SELECT Orders.OrderID, Customers.CustomerName FROM Orders
INNER JOIN Customers ON Orders.CustomerID =
Customers.CustomerID;
```

Hasilnya:

OrderID	CustomerName
10248	Wilman Kala
10249	Tradição Hipermercados
10250	Hanari Carnes
10251	Victuailles en stock
10252	Suprêmes délices
10253	Hanari Carnes
10254	Chop-suey Chinese
10255	Richter Supermarkt
10256	Wellington Importadora

B. Left Outer Join

Merupakan operasi join yang digunakan untuk menghasilkan kumpulan record dari dua buah table atau lebih, dengan asumsi bahwa table sebelah kiri sebagai rujukan. Anggotanya adalah semua yang memiliki nilai sama dengan asumsi table sebelah kiri tersebut.



1. Sintak

```
SELECT column_name(s) FROM table1  
LEFT JOIN table2 ON table1.column_name = table2.column_name;
```

2. Contoh penerapan

a. Left Join tabel Orders dan Customers

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

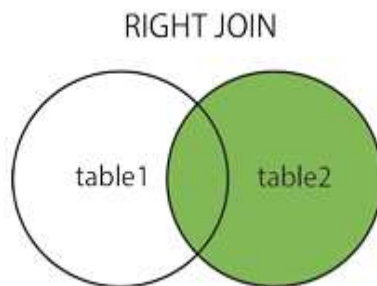
```
SELECT Customers.CustomerName, Orders.OrderID FROM Customers LEFT  
JOIN Orders ON Customers.CustomerID=Orders.CustomerID ORDER BY  
Customers.CustomerName;
```

Hasilnya:

CustomerName	OrderID
Alfreds Futterkiste	null
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365
Around the Horn	10355
Around the Horn	10383
B's Beverages	10289
Berglunds snabbköp	10278
Berglunds snabbköp	10280
Berglunds snabbköp	10384

C. Right Outer Join

Merupakan operasi join yang digunakan untuk menghasilkan kumpulan record dari dua buah table atau lebih dengan asumsi bahwa table sebelah kanan sebagai rujukan, sehingga anggotanya adalah semua yang memiliki nilai sama dengan asumsi table sebelah kanan tersebut.



1. Sintaknya

```
SELECT column_name(s) FROM table1  
RIGHT JOIN table2 ON table1.column_name = table2.c  
olumn_name;
```

2. Contoh penerapan

a. Right Join tabel Orders dan Employee

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

EmployeeID	LastName	FirstName	BirthDate	Photo
1	Davolio	Nancy	12/8/1968	EmpID1.pic
2	Fuller	Andrew	2/19/1952	EmpID2.pic
3	Leverling	Janet	8/30/1963	EmpID3.pic

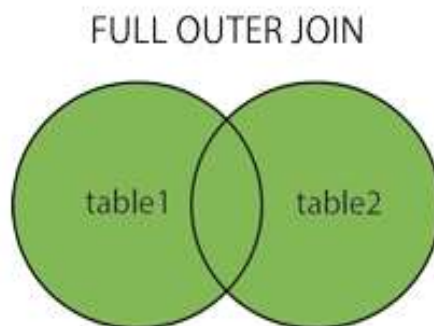
```
SELECT Orders.OrderID, Employees.LastName, Employees.FirstName  
FROM Orders RIGHT JOIN Employees ON Orders.EmployeeID =  
Employees.EmployeeID ORDER BY Orders.OrderID;
```

Hasilnya:

OrderID	LastName	FirstName
	West	Adam
10248	Buchanan	Steven
10249	Suyama	Michael
10250	Peacock	Margaret
10251	Leverling	Janet
10252	Peacock	Margaret
10253	Leverling	Janet
10254	Buchanan	Steven

D. Full Outer Join

Merupakan operasi join yang digunakan untuk menghasilkan kumpulan record dari dua buah table atau lebih anggotanya adalah semua yang memiliki nilai sama ataupun tidak sama.



1. Sintak

```
SELECT column_name(s) FROM table1  
FULL OUTER JOIN table2 ON table1.column_name = table2.column_name;
```

2. Contoh penerapan

a. Full Join tabel Orders dan Customers

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

```

SELECT Customers.CustomerName, Orders.OrderID FROM Customers
FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.Customer
ID
ORDER BY Customers.CustomerName;

```

Hasilnya:

CustomerName	OrderID
Alfreds Futterkiste	
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365
	10382
	10351

DAFTAR PUSTAKA

Abdul Kadir (2013). Pengertian MySQL. Tersedia dalam : Buku Pintar Programmer Pemula PHP. Yogyakarta. Mediakom.

Hutabarat, Bernaridho. 2005. Datawarehousing dengan SQL Server. Jakarta : PT Elex Media Komputindo

Priyadi, Yudi. 2014. Kolaborasi SQL & ERD Dalam Implementasi Database. Penerbit Andi. Yogyakarta.

Raharjo, Budi. 2011. Membuat Database Menggunakan MySql. Bandung : Informatika.

The World's Largest Web Developer Site. 2015. Learn SQL
. <https://www.w3schools.com/sql/default.asp>. Diakses pada tanggal 2 September 2015.