

# ***PERTEMUAN 10***

## **SQL : Data Manipulation (*Chap. 6 – Conoly*)**

## TUJUAN DAN PENTINGNYA SQL

**SQL adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional.**

- SQL adalah sebuah bahasa permintaan database yang terstruktur. Bahasa SQL dibuat sebagai bahasa yang dapat merelasikan beberapa tabel dalam database maupun merelasikan antar database
- Idealnya, suatu bahasa database harus memungkinkan user untuk :
  - membuat struktur database dan hubungan (relasi)
  - melakukan tugas dasar mengelola data, seperti penyisipan (insert), perubahan (modify), dan penghapusan (delete) data dari hubungan
  - melakukan pertanyaan sederhana dan rumit

## QUERY DALAM SQL

Query adalah bahasa perintah dalam SQL

SQL memiliki 2 komponen utama :

1. *Data Definition Language (DDL)* untuk mendefinisikan struktur database dan mengendalikan akses ke data

- CREATE : membuat tabel atau database
- DROP : menghapus tabel atau database
- ALTER : mengubah struktur tabel, seperti  
menambah Field (Add), mengganti  
nama Field (change) atau rename

## QUERY DALAM SQL

2. *Data Manipulation Language (DML)* untuk mengambil dan memperbarui data

- INSERT : menginput/memasukkan data pada tabel
- UPDATE : memperbaharui data
- DELETE : menghapus data pada Tabel

## SEJARAH SQL

- Tahun 1970  
EF Codd → publikasi paper sejarah model relasional
- Tahun 1974  
D. Chamberlin → *Structured English Query Language (SEQUEL)*
- Tahun 1976  
D. Chamberlin → edisi revisi *SEQUEL/2*, kemudian berubah menjadi SQL untuk alasan hukum  
banyak orang masih mengucapkan SQL sebagai "See-Quel", meskipun lafal resminya adalah "S-Q-L".

## SEJARAH SQL

- 1976  
IBM menghasilkan prototipe DBMS → Sistem *R*  
akar dari SQL ada di bahasa *SQUARE* (*Specifying  
Queries As Relational Expressions*)
- 1981-1983  
RDBMS komersial pertama → SQL/DS, untuk  
DOS/VSE dan VM/CMS, kemudian sebagai DB2 untuk  
MVS
- 1984  
ANSI & ISO → *Relational Database Language (RDL)*
- 1992  
ISO → SQL2 atau SQL-92 (ISO, 1992)

# Sejarah SQL

- 1999  
ISO → SQL:1999 (ISO, 1999a)
- 2003  
ISO → SQL: 2003
- 2008  
ISO → SQL: 2008

## MENULIS PERINTAH SQL

- Sebuah pernyataan SQL terdiri dari :
  - **Reserved words** adalah bagian tetap dari bahasa SQL dan memiliki makna tetap.
  - **User-defined words** dibuat oleh pengguna (sesuai dengan aturan sintaks tertentu) dan mewakili nama-nama berbagai objek database seperti tabel, kolom, tampilan, indeks, dan sebagainya.

Kebanyakan komponen pernyataan SQL bersifat **case-insensitive**, yang berarti peka terhadap penggunaan huruf besar dan huruf kecil.

Contoh : jika kita menyimpan nama seseorang sebagai "SMITH" dan kemudian mencari dengan menggunakan string "Smith," baris tidak akan ditemukan.



## MENULIS PERINTAH SQL

- notasi *Backus Nur Form (BNF)* untuk mendefinisikan pernyataan SQL :
  - huruf besar digunakan untuk mewakili *reserved words* dan harus dieja persis seperti yang ditampilkan;
  - huruf kecil digunakan untuk mewakili *user-defined words*;
  - sebuah bar vertikal ( | ) menunjukkan **pilihan** di antara alternatif, misalnya, a b | C;
  - kurung kurawal menunjukkan **elemen yang diperlukan**, misalnya, {a};
  - tanda kurung persegi untuk menunjukkan **elemen opsional**, misalnya, [a];
  - ellipsis (...) Digunakan untuk menunjukkan **pengulangan** opsional item nol atau lebih.

## MANIPULASI DATA

- pernyataan *Data Manipulation Language (DML)* SQL:
  - SELECT - untuk query data dalam database;
  - INSERT - untuk memasukkan data ke dalam tabel;
  - UPDATE - untuk memperbarui data dalam tabel;
  - DELETE - menghapus data dari tabel.
- Semua nilai data *nonnumeric* harus diapit tanda kutip tunggal, semua nilai data *numerik* tidak harus diapit tanda kutip tunggal.

Contoh :

**INSERT INTO** PropertyForRent(propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo, branchNo)

**VALUES** ('PA14', '16 Holhead', 'Aberdeen', 'AB7 5SU', 'House', 6, 650.00, 'CO46', 'SA9', 'B007');

## QUERY SEDERHANA

- Tujuan dari pernyataan SELECT adalah untuk mengambil dan menampilkan data dari satu atau lebih tabel database.
- SELECT adalah perintah SQL yang paling sering digunakan dan memiliki bentuk umum sebagai berikut :

```
SELECT          [DISTINCT | ALL] { * | [columnExpression  
                  [AS newName]] [, . . .])  
FROM           TableName [alias] [, . . .]  
[WHERE         condition]  
[GROUP BY     columnList] [HAVING condition]  
[ORDER BY     columnList]
```

## QUERY SEDERHANA

- Urutan pengolahan dalam sebuah pernyataan SELECT adalah :  
FROM                spesifik tabel atau tabel yang akan digunakan  
WHERE              filter baris subjek pada beberapa kondisi  
GROUP BY          bentuk kelompok baris dengan nilai kolom yang sama  
HAVING             filter kelompok subjek pada beberapa kondisi  
SELECT             menentukan kolom mana yang muncul dalam output  
ORDER BY          menentukan urutan output

## CONTOH QUERY SEDERHANA

cara cepat untuk mengungkapkan "semua kolom" di SQL, dengan menggunakan tanda bintang (\*)

- **CONTOH : Mengambil semua kolom, semua baris**

*Daftar lengkap rincian semua staf.*

```
SELECT staffNo, fName, lName, position, sex, DOB, salary, branchNo  
FROM Staff;
```

- Cara cepat :

```
SELECT *  
FROM Staff;
```

## CONTOH QUERY SEDERHANA

- Tabel Hasil

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	01-Okt-45	30000.00	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000.00	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000.00	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000.00	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000.00	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000.00	B005



## CONTOH QUERY SEDERHANA

- **CONTOH 6.2 Mengambil kolom tertentu, semua baris**  
*Menghasilkan daftar gaji untuk semua staf hanya menampilkan jumlah staf, nama pertama dan terakhir, dan rincian gaji.*

**SELECT** staffNo, fName, lName, salary  
**FROM** Staff;

**Tabel Hasil**

staffNo	fName	lName	salary
SL21	John	White	30000.00
SG37	Ann	Beech	12000.00
SG14	David	Ford	18000.00
SA9	Mary	Howe	9000.00
SG5	Susan	Brand	24000.00
SL41	Julie	Lee	9000.00

## MENGGUNAKAN FUNGSI AGREGAT SQL

Standar ISO mendefinisikan lima fungsi agregat :

- COUNT — menghasilkan jumlah nilai dalam sebuah kolom tertentu
- SUM — menghasilkan jumlah nilai dalam kolom tertentu
- AVG — menghasilkan rata-rata nilai dalam sebuah kolom tertentu
- MIN — menghasilkan nilai terkecil pada kolom yang ditentukan
- MAX — menghasilkan nilai terbesar dalam satu kolom tertentu



## MENGGUNAKAN COUNT(\*)

- **CONTOH Menggunakan COUNT(\*)**

*Berapa banyak biaya properti lebih dari £350 per bulan untuk menyewa?*

```
SELECT COUNT(*) AS myCount  
FROM PropertyForRent  
WHERE rent >350;
```

- **Contoh Menggunakan COUNT(DISTINCT)**

*Berapa banyak properti yang berbeda ditampilkan bulan Mei 2008?*

```
SELECT COUNT(DISTINCT propertyNo) AS myCount  
FROM Viewing  
WHERE viewDate BETWEEN '1-May-08' AND '31-May-08';
```

## MENGGUNAKAN COUNT DAN SUM

- Contoh Menggunakan COUNT dan SUM

*Mencari jumlah Manajer dan jumlah gaji mereka.*

```
SELECT COUNT(staffNo) AS myCount, SUM(salary) AS mySum  
FROM Staff  
WHERE position = 'Manager';
```

**Tabel Hasil**

myCount	mySum
2	54000.00

## MENGGUNAKAN MIN, MAX, AVG

- **CONTOH Menggunakan MIN, MAX, AVG**

*Mencari minimum, maksimum, dan rata-rata gaji staf.*

```
SELECT    MIN(salary) AS myMin, MAX(salary) AS myMax,  
AVG(salary) AS myAvg  
FROM Staff;
```

- **TABEL Hasil**

myMin	myMax	myAvg
9000.00	30000.00	17000.00

## MEMPERBAHARUI DATABASE

Tiga pernyataan SQL yang tersedia untuk memodifikasi isi dari tabel dalam database :

- INSERT : menambah baris baru dari data ke tabel
- UPDATE : memodifikasi data yang ada dalam tabel
- DELETE : menghapus baris data dari tabel

## MENAMBAHKAN DATA KE DALAM DATABASE (INSERT)

- Ada dua bentuk pernyataan INSERT.
- Yang pertama memungkinkan satu baris untuk dimasukkan ke tabel bernama dan memiliki format berikut :

**INSERT INTO** TableName [(columnList)]  
**VALUES** (dataValueList)

### CONTOH INSERT... VALUES

*Menyisipkan baris baru ke dalam tabel Staff untuk memasok data untuk semua kolom.*

**INSERT INTO** Staff  
**VALUES** ('SG16', 'Alan', 'Brown', 'Assistant', 'M', **DATE** '1957-05-25',  
8300, 'B003');

## MENAMBAHKAN DATA KE DALAM DATABASE (INSERT)

- Bentuk kedua dari pernyataan INSERT memungkinkan beberapa baris untuk disalin dari satu atau lebih tabel yang lain, dan memiliki format berikut :
- **INSERT INTO** TableName [(columnList)]  
**SELECT**...

### CONTOH INSERT... SELECT

```
INSERT INTO StaffPropCount
(SELECT s.staffNo, fName, IName, COUNT(*)
  FROM Staff s, PropertyForRent p
 WHERE s.staffNo = p.staffNo
 GROUP BY s.staffNo, fName, IName)
UNION
(SELECT staffNo, fName, IName, 0
  FROM Staff s
 WHERE NOT EXISTS (SELECT *
                   FROM PropertyForRent p
                   WHERE p.staffNo = s.staffNo));
```

## MODIFIKASI DATA PADA DATABASE (UPDATE)

Pernyataan UPDATE memungkinkan isi baris yang ada di tabel bernama diubah. Format, perintah adalah :

**UPDATE** TableName

**SET** columnName1 = dataValue1 [,columnName2 = dataValue2 . . . ]

**[WHERE** searchCondition]

Jika klausa WHERE diterapkan, hanya baris yang memenuhi *searchCondition* yang diperbarui. Para *dataValue* baru harus sesuai dengan tipe data untuk kolom yang sesuai.

## CONTOH UPDATE semua baris

*Berikan semua staf kenaikan gaji 3%.*

**UPDATE** Staff

**SET** salary = salary\*1.03;

## ◎ CONTOH UPDATE baris spesifik

*Berikan Manajer minyak kenaikan gaji 5%.*

**UPDATE** Staff

**SET** salary = salary\*1.05

**WHERE** position = 'Manager';

## ◎ CONTOH UPDATE banyak kolom

*Promosikan David Ford (staffNo = 'SGI4') untuk Manajer dan mengubah gajinya menjadi £18.000.*

**UPDATE** Staff

**SET** position = 'Manager', salary = 18000

**WHERE** staffNo = 'SGI4';



## Menghapus Data dari Database (DELETE)

- *Pernyataan DELETE memungkinkan baris yang akan dihapus dari tabel bernama. Format perintahnya adalah :*

**DELETE FROM** TableName  
[**WHERE** searchCondition]

- **CONTOH DELETE baris yang spesifik**

*Hapus semua tampilan yang berhubungan dengan properti PG4.*

**DELETE FROM** Viewing  
**WHERE** propertyNo = 'PG4';

- **CONTOH DELETE semua baris**

*Hapus semua baris dari tabel Viewing.*

**DELETE FROM** Viewing;