CHAPTER

14

SQL Operasi DML

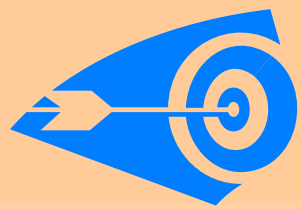
Arif Basofi, S.Kom
Information Technology, PENS - ITS

Objectives

Tujuan:

Mengenal operasi perintah SQL dalam:

- **DML (Data Manipulation Language)** [1]
- DDL (Data Definition Language) [2]



S Q L – DML

- **DML** (**Data Manipulation Language**) adalah inti dari SQL.
- **DML** adalah suatu operasi dalam SQL yang sering digunakan dalam manipulasi data.
- **DML** sering digunakan oleh DBA Administratora atau programmer.
- Statement **DML** sering di eksekusi saat melakukan operasi:
  - **QUERY (Pemanggilan Data)**
  - **INSERT (Penambahan Data)**
  - **UPDATE (Pengubahan Data)**
  - **DELETE (Penghapusan Data)**
- Suatu **TRANSAKSI** terdiri atas kumpulan statement DML yang membentuk suatu **unit logic kerja**.

S Q L – DML (INSERT)

INSERT / Menambahkan Record Baru pada Table

70	Public Relations	100	1700

New row

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

…insert a new row into the DEPARMENTS table…

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700
70	Public Relations	100	1700

S Q L – DML (INSERT)

Syntax Statement INSERT:

- Tambahkan row baru pada table menggunakan statement **INSERT**.

```
INSERT INTO table [(column [, column...])]
VALUES      (value [, value...]);
```

- Hanya **satu baris / row** yang ter-insert secara serentak dengan syntax ini.

S Q L – DML (INSERT)

Insert Baris Baru

- Saat Insert baris/row baru, **mengandung nilai-nilai** untuk tiap kolom.
- **Urutan nilai** yang dimasukkan harus **sesuai** dengan urutan kolom table dan harus sesuai dengan **tipe datanya**.
- Contoh:

```
INSERT INTO departments(department_id, department_name,
                        manager_id, location_id)
VALUES      (70, 'Public Relations', 100, 1700);
1 row created.
```

- Masukkan untuk tiap nilai bertipe **character** dan **date** di dalam **tanda petik satu (single quotation marks) (‘…’).**

S Q L – DML (INSERT)

Insert Baris (rows) dengan Nilai Null

- Ada kalanya untuk mengisi/insert data hanya pada beberapa kolom saja. Artinya kolom yang yang tidak didefinisikan akan bernilai null.
- Contoh:

```
INSERT INTO   departments (department_id,
                           department_name )
VALUES        (30, 'Purchasing');
1 row created.
```

- Jika ingin menspesifikkan nilai bernilai Null, maka keyword NULL dapat dimasukkan dalam klausa VALUES.

```
INSERT INTO   departments
VALUES        (100, 'Finance', NULL, NULL);
1 row created.
```

S Q L – DML (INSERT)

Meng-Copy Baris (rows) dari Table Lain

- Operasi Insert dalam memasukkan data baru, dapat dilakukan **insert data dari table lain** dengan menggunakan operasi **sub-query**. Jadi seakan-akan meng-copy dari table lain.
- Contoh:

```
INSERT INTO sales_reps(id, name, salary, commission_pct)
  SELECT employee_id, last_name, salary, commission_pct
  FROM   employees
  WHERE  job_id LIKE '%REP%';

4 rows created.
```

- Jangan gunakan klausa **VALUES !!!**
- Dalam sub-query, **nilai kolom** harus **sesuai** dengan **definisi kolom** yang ditentukan.

S Q L – DML (UPDATE)

UPDATE / Merubah Data dalam Table

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSION_P
100	Steven	King	SKING	17-JUN-87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	9000	60	
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	6000	60	
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	4200	60	
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800	50	

Update rows in the EMPLOYEES table.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSIO
100	Steven	King	SKING	17-JUN-87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	9000	30	
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	6000	30	
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	4200	30	
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800	50	

S Q L – DML (UPDATE)

Syntax Statement Update

- Modifikasi data row yang telah ada dengan statement **UPDATE**.

```
UPDATE      table
SET         column = value [, column = value, ...]
[WHERE      condition];
```

- Update pada lebih dari satu row baris kadang juga diperlukan.
- GUNAKAN:
- **COMMIT**: untuk menyelesaikan operasi DML tersimpan dalam database.
- **ROLLBACK**: untuk mengembalikan data di awal sebelum transaksi (DML) terakhir kali.

S Q L – DML (UPDATE)

Update Baris/Row dalam Tabel

- Jika ingin meng-update pada **spesifik** row, maka klausa **WHERE** harus di **ikut sertakan** sebagai filter/pembatas.
- Contoh:

```
UPDATE employees
SET    department_id = 70
WHERE  employee_id = 113;
1 row updated.
```

- Jika ingin meng-update **keseluruhan** row dalam table, maka klausa **WHERE** dapat **dihilangkan**.
- Contoh:

```
UPDATE   copy_emp
SET      department_id = 110;
22 rows updated.
```

S Q L – DML (UPDATE)

Update Dua Kolom dengan Sub-query

- Update juga dapat dilakukan pada **lebih dari dua kolom** dengan menggunakan **sub-query**.
- Contoh:

  Update pada kolom **job_id** dan **salary** agar nilainya sesuai dengan **employee_id no 205,** khusus untuk **employee_id = 114**.

```
UPDATE   employees
SET      job_id  = (SELECT  job_id
                    FROM    employees
                    WHERE   employee_id = 205),
         salary  = (SELECT  salary
                    FROM    employees
                    WHERE   employee_id = 205)
WHERE    employee_id    =  114;
1 row updated.
```

S Q L – DML (DELETE)

DELETE / Menghapus Record pada Table

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing		
100	Finance		
50	Shipping	124	1500
60	IT	103	1400

Delete a row from the DEPARTMENTS table.

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing		
50	Shipping	124	1500
60	IT	103	1400

S Q L – DML (DELETE)

Syntax Statement Delete

- Gunakan statement **DELETE** untuk menghapus data row yang ada dalam table.

```
DELETE [FROM] table
[WHERE        condition];
```

S Q L – DML (DELETE)

Delete Baris/Row dalam Tabel

- Jika ingin men-delete pada **spesifik** row, maka klausa **WHERE** harus di **ikut sertakan** sebagai filter/pembatas.
- Contoh:

```
 DELETE FROM departments
 WHERE  department_name = 'Finance';
1 row deleted.
```

- Jika ingin men-delete **keseluruhan** row dalam table, maka klausa **WHERE dapat dihilangkan**.
- Contoh:

```
DELETE FROM  copy_emp;
22 rows deleted.
```

S Q L – DML (DELETE)

Delete Rows Berdasarkan Table yang Lain

- Untuk menghapus/delete data row dengan nilai yang berasal dari table lain, dapat dilakukan dengan menggunakan **sub-query**.
- Contoh:

```
DELETE FROM employees
WHERE  department_id =
                       (SELECT department_id
                        FROM   departments
                        WHERE  department_name LIKE '%Public%');

1 row deleted.
```

S Q L – DML (WARNING)

DML Warning

- Yang perlu diperhatikan dalam operasi DML, baik Insert, Update maupun Delete, adalah kondisi **constraint** dari table.
- Jika terdapat constraint antar table (adanya **Primary Key** dan **Foreign Key** / Parent dan Child), misal. saat Delete data, maka akan terdapat **error**.
- Contoh:

```
DELETE FROM departments
WHERE       department_id = 60;
```

```
DELETE FROM departments
            *
ERROR at line 1:
ORA-02292: integrity constraint (HR.EMP_DEPT_FK)
violated - child record found
```

Anda tidak dapat men-**delete** row yang mengandung **primary key** yang digunakan sebagai **foreign key** pada table yang lain.

S Q L – DML (WARNING)

DML Warning

- Misal pada Update.
- Contoh: Tidak terdapat adanya department_id = 55 sebagai Primary Key dari parent table tersebut.

```
UPDATE employees
SET    department_id = 55
WHERE  department_id = 110;
```

```
UPDATE employees
       *
ERROR at line 1:
ORA-02291: integrity constraint (HR.EMP_DEPT_FK)
violated - parent key not found
```

Department number 55 does not exist

S Q L – Excercise

1. Ubahlah data nama belakang pegawai pada table employees, yang memiliki nama depan ‘Julia’ menjadi ‘Robert’. Lakukan perintah commit setelah melakukan perubahan tersebut.
2. Hapus data tabel departments khusus pada no department 50. Apa yang terjadi? Lalu bagaimana solusinya?
3. Ubahlah gaji pegawai menjadi $800, untuk pegawai yang bekerja pada department “IT Dept” atau “Finance”.
4. Masukkan data baru pada table mahasiswa, sesuai table berikut:

NRP	NAMA	ALAMAT	NO_TELP
7404030040	Lydia	ITS 40	5931040
7404030041	Olive	ITS 41	5931041
7404030042	Valentino	ITS 42	5931042
7404030043	Emyra	ITS 43	5931043

4. Ubah nama mahasiswa yang bernama ‘Valentino’ menjadi ‘Rossi’.
5. Hapus data mahasiswa yang beralamat ITS 42 dan ITS 43.

S Q L – Excercise