

BASIS DATA PERTEMUAN 5
DATA MANIPULATION LANGUAGE (DML) PADA DBMS MYSQL
Annisa Puspa Kirana, S.Kom, M.Kom

Pendahuluan

DML merupakan istilah untuk beberapa sintaksis (syntax) dari SQL yang digunakan untuk melakukan perubahan pada data (isi tabel-tabel) dalam suatu database. DML terdiri dari 3 klausa utama yaitu:

1. **INSERT** : Menambah baris baru pada sebuah tabel
2. **UPDATE** : Mengubah nilai suatu baris pada sebuah tabel.
3. **DELETE** : Menghapus suatu baris dari sebuah tabel.

Silahkan login ke SQL anda. Silahkan masuk ke CMD ketikkan **cd c:/xampp/mysql/bin.** selanjutnya ketikkan **mysql -u root -p** kemudian tekan enter. Maka akan muncul perintah untuk memasukan password, ketikkan password yang telah Anda set sebelumnya, jika tidak menggunakan password silahkan langsung langsung tekan Enter.

Sebelum melakukan latihan DDL berikut ini silahkan export file “minimart.sql” yang telah diberikan. Berikut caranya.

1. Operasi INSERT

Operasi insert bertujuan untuk menyisipkan satu tuple baru ke dalam suatu relasi R.

Klausa pembentuk:

1. **INSERT**
2. **INTO**
3. **VALUES**

Format:

1. **INSERT INTO** nama_tabel (kolom1, kolom2, ...dst.) **VALUES** (nilai_kolom1, nilai_kolom2, ...dst.);
2. **INSERT INTO** nama_tabel **VALUES** (nilai_kolom1, nilai_kolom2, ...dst.);
3. [Salah satu dari kedua format sebelumnya], (nilai_kolom_kolom_baris1), (nilai_kolom_kolom_baris2), ...dst.

Operasi ini memungkinkan untuk melanggar empat jenis constraint sebagaimana dijelaskan berikut ini :

1. **DOMAIN Constraint** dapat dilanggar jika suatu nilai attribute yang diberikan tidak ada dalam domain yang berkorespondensi dengan attribute tadi.
2. **KEY Constraint** dapat dilanggar jika nilai key dalam tuple baru t sudah ada dalam tuple lain dalam relasi r(R).
3. **ENTITY INTEGRITY** Constraint dapat dilanggar jika primary key dari tuple baru t adalah NULL
4. **REFERENTIAL INTEGRITY** Constraint dapat dilanggar jika nilai dari suatu foreign key dalam t mengacu ke suatu tuple yang tidak ada dalam relasi yang diacu.

Ada dua pilihan tindakan yang dapat dilakukan jika ada satu atau lebih constraint yang dilanggar akibat operasi insert, yaitu :

1. Menolak (reject) operasi insertion. Biasanya DBMS memberikan penjelasan mengapa proses insertion ditolak.
2. Berusaha memperbaiki alasan penolakan proses insertion. Dimana insertion akan diterima jika user melakukan perubahan nilai-nilai attribute sehingga insertion diterima.

Untuk memasukkan data / record pada tabel siswa, maka perintah yang kita gunakan adalah perintah insert, struktur perintahnya adalah :

***INSERT INTO nama_tabel (kolom1, kolom2, kolom3, kolom_n)
values ('isi 1 ', 'isi 2 ', 'isi 3 ', 'isi n ')***

Pada contoh berikut kita akan memasukkan record pada tabel siswa yang telah kita buat sebelumnya.

INSERT INTO mahasiswa (nim, nama) values (1, 'Ani Rahmawati');

```
MariaDB [akademik]> INSERT INTO mahasiswa (nim, nama ) values (1, 'Ani Rahmawati');  
Query OK, 1 row affected (0.06 sec)
```

jika perintah diatas kita jalankan, maka akan dimasukkan data siswa bernama Ani kedalam tabel mahasiswa. Jika perintah berhasil, akan dicetak pesan Query Ok, 1 row affected

INSERT INTO mahasiswa (nim, nama) values (2, 'Budi Raharjo', '08578776453');

```
MariaDB [akademik]> INSERT INTO mahasiswa values (2, 'Budi Raharjo', '08578776453');  
Query OK, 1 row affected (0.12 sec)
```

***INSERT INTO mahasiswa values
(3, 'Charlie', '08571234453'),
(4, 'Diandra', '08578771234');***

```
MariaDB [akademik]> INSERT INTO mahasiswa values  
-> (3, 'Charlie', '08571234453'),  
-> (4, 'Diandra', '08578771234');  
Query OK, 2 rows affected (0.10 sec)  
Records: 2 Duplicates: 0 Warnings: 0
```

2. Operasi DELETE

Operasi delete bertujuan untuk menghapus satu atau beberapa tuple di dalam suatu relasi R. Operasi ini hanya dapat melanggar **referential integrity**, jika tuple yang dihapus diacu oleh kunci-kunci tamu dari tuple yang lain dalam basis data.

Klausula pembentuk:

1. **DELETE**
2. **FROM**
3. **WHERE** [opsional]

Format:

1. **DELETE FROM** nama_tabel **WHERE** nama_kolom_patokan [operator_perbandingan] nilai_patokan;
2. **DELETE * FROM** nama_tabel; atau **DELETE FROM** nama_tabel;

*Operator perbandingan/comparison operator dapat berupa: =, <, >, <=, >=, <>

Ada empat pilihan tindakan yang dapat dilakukan jika suatu deletion melanggar constraint yang telah ditentukan, yaitu :

1. Menolak(reject) proses deletion.
2. Berusaha untuk melakukan “cascade deletion”, yaitu dengan menghapus sejumlah tuple yang mengacu pada tuple yang akan dihapus.
3. Melakukan modifikasi nilai attribute yang mengacu pada tuple yang dihapus, yaitu setiap nilai diset NULL atau diganti dengan nilai dari tuple lain yang valid sebagai acuan baru. Akan tetapi, bila

attribute yang mengacu yang menyebabkan pelanggaran adalah bagian dari primary key, maka ia tidak dapat diset NULL (karena melanggar entity integrity).

4. Kombinasi 2 dan 3.

Contoh

```
MariaDB [AKADEMIK]> SELECT*FROM MATAKULIAH;
+-----+-----+-----+
| kode | nama                                | sks |
+-----+-----+-----+
| ASD  | ALGORITMA DAN STRUKTUR DATA      | 3   |
| KCB  | KECERDASAN BUATAN                 | 2   |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

DELETE FROM MATAKULIAH WHERE KODE= 'BDD';

Coba perhatikan berapa baris yang berhasil di delete dengan perintah di atas

```
MariaDB [AKADEMIK]> DELETE FROM MATAKULIAH WHERE KODE= 'BDD';
Query OK, 0 rows affected (0.00 sec)

MariaDB [AKADEMIK]> DELETE FROM MATAKULIAH WHERE KODE= 'ASD';
Query OK, 1 row affected (0.13 sec)
```

3. Operasi UPDATE

Operasi update digunakan untuk merubah nilai-nilai satu atau lebih attribute dalam satu atau lebih tuple dalam sejumlah relasi R.

Klausula pembentuk:

1. **UPDATE**
2. **SET**
3. **WHERE** [opsional]

Format:

1. **UPDATE** nama_tabel **SET** nama_kolom = nilai_baru **WHERE** nama_kolom_patokan [operator_perbandingan] nilai_patokan;
2. **UPDATE** nama_tabel **SET** nama_kolom1 = nilai_baru1, nama_kolom2 = nilai_baru2, ...dst. **WHERE** nama_kolom_patokan [operator_perbandingan] nilai_patokan;

*Operator perbandingan/comparison operator dapat berupa: =, <, >, <=, >=, <>

Operasi UPDATE bisa dilakukan pada tiga jenis attribut, dengan permasalahan yang berbeda sebagaimana berikut ini :

1. Modifikasi nilai suatu foreign key, maka DBMS harus melakukan pengecekan bahwa nilai-nilai baru yang diberikan mengacu pada tuple yang ada dalam relasi-relasi yang dijadikan acuan.
2. modifikasi nilai suatu primary key serupa dengan proses deletion satu tuple dan inserting yang lain pada tempat yang sama. Akibatnya, pilihan- pilihan seperti yang dilakukan pada operasi INSERT dan DELETE dapat dipakai agar modifikasi tidak melanggar constraint.

modifikasi suatu attribute yang bukan primary key atau bukan foreign key biasanya tidak akan menimbulkan masalah. DBMS hanya perlu untuk mengecek apakah nilai-nilai baru yang diberikan mempunyai tipe data dan domain yang valid.

Contoh

```
MariaDB [AKADEMIK]> SELECT*FROM MATAKULIAH;
+-----+-----+-----+
| kode | nama                | sks |
+-----+-----+-----+
| KCB  | KECERDASAN BUATAN  | 2   |
+-----+-----+-----+
1 row in set (0.00 sec)
```

```
MariaDB [AKADEMIK]> UPDATE `matakuliah` SET nama='Matakuliah Kecerdasan Buatan',sks='4' WHERE kode='KCB';
Query OK, 1 row affected (0.15 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
MariaDB [AKADEMIK]> SELECT*FROM MATAKULIAH;
+-----+-----+-----+
| kode | nama                | sks |
+-----+-----+-----+
| KCB  | Matakuliah Kecerdasan Buatan | 4   |
+-----+-----+-----+
1 row in set (0.00 sec)
```

4. Menampilkan semua data atau record dan pada kolom tertentu

Untuk melihat data dari sebuah tabel, maka kita gunakan perintah select , yang srtuktur perintahnya adalah sebagai berikut :

SELECT kolom1, kolom2, kolom_n FROM nama_tabel

pada contoh berikut kita akan menampilkan isi dari tabel siswa, maka kita gunakan perintah berikut :

SELECT * FROM mahasiswa ;

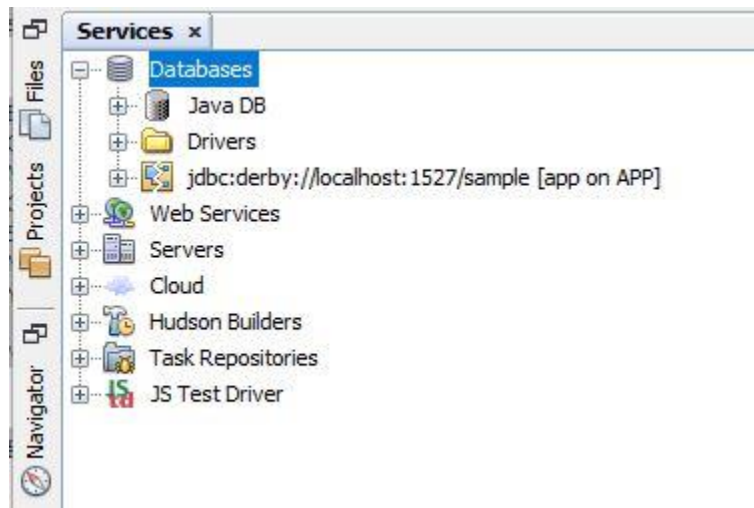
```
MariaDB [akademik]> SELECT * FROM mahasiswa ;
+-----+-----+-----+
| nim | nama                | nomor_telepon |
+-----+-----+-----+
| 1   | Ani Rahmawati      | NULL          |
| 2   | Budi Raharjo       | 08578776453   |
| 3   | Charlie            | 08571234453   |
| 4   | Diandra            | 08578771234   |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

KONEKSI MYSQL KE NETBEANS

The screenshot displays the NetBeans IDE interface. At the top, the 'SQL 1' window shows a MySQL connection string: `jdbc:mysql://localhost:3306/minimart?zeroDateTimeBehavior=convertToNull [ro...`. Below the connection string, two SQL queries are listed. The first query is `select * from agents;`. The second query, which is highlighted, is `select agent_name, working_area, phone_no from agents where agent_code='A007';`. Below the SQL editor, the 'Output - SQL 1 execution' window shows the execution results. It indicates that the query was executed successfully in 0.001 seconds. The results are displayed in a table with the following data:

#	agent_name	working_area	phone_no
1	Ramasundar	Bangalore	077-25814763

The output window also shows the message: 'Execution finished after 0.001 s, 0 error(s) occurred.'



New Connection Wizard

Locate Driver

Driver: MySQL (Connector/J driver) ▼

Driver File(s): C:\Program Files (x86)\NetBeans 8.0.2\ide\modules\ext\mysql-connector-java-5.1.21

Add... Remove

< >

< Back Next > Finish Cancel Help

New Connection Wizard

Customize Connection

Driver Name: MySQL (Connector/J driver) ▼

Host: localhost Port: 3306

Database: minimart

User Name: root

Password:

☐ Remember password

Connection Properties Test Connection

JDBC URL: jdbc:mysql://localhost:3306/minimart?zeroDateTimeBehavior=convertToNull

< Back Next > Finish Cancel Help

Connection Properties

Test Connection

JDBC URL: jdbc:mysql://localhost:3306/minimart?zeroDateTimeBehavior=convertToNull

Connection Succeeded.

New Connection Wizard

Choose name for connection

Override the default name for the connection. The name should be descriptive about the connection you are creating.

Input connection name:
jdbc:mysql://localhost:3306/minimart?zeroDateTimeBehavior=convertToNull [root on Default schema]

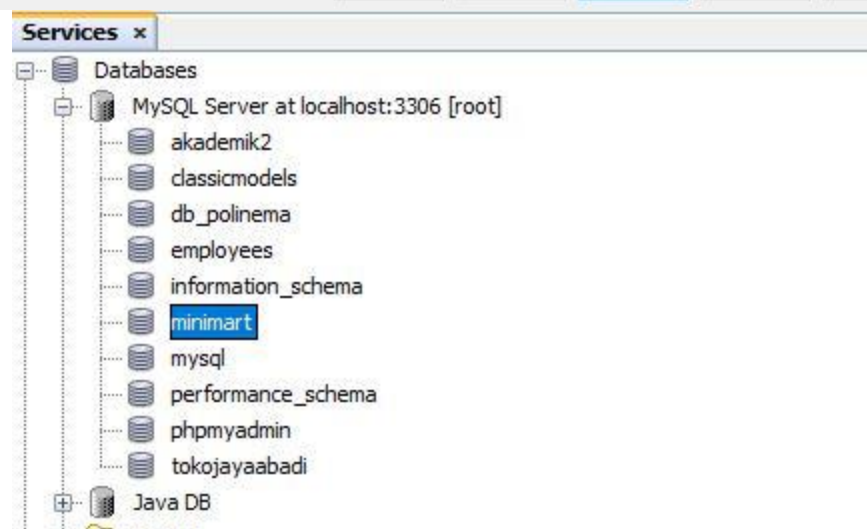
< Back

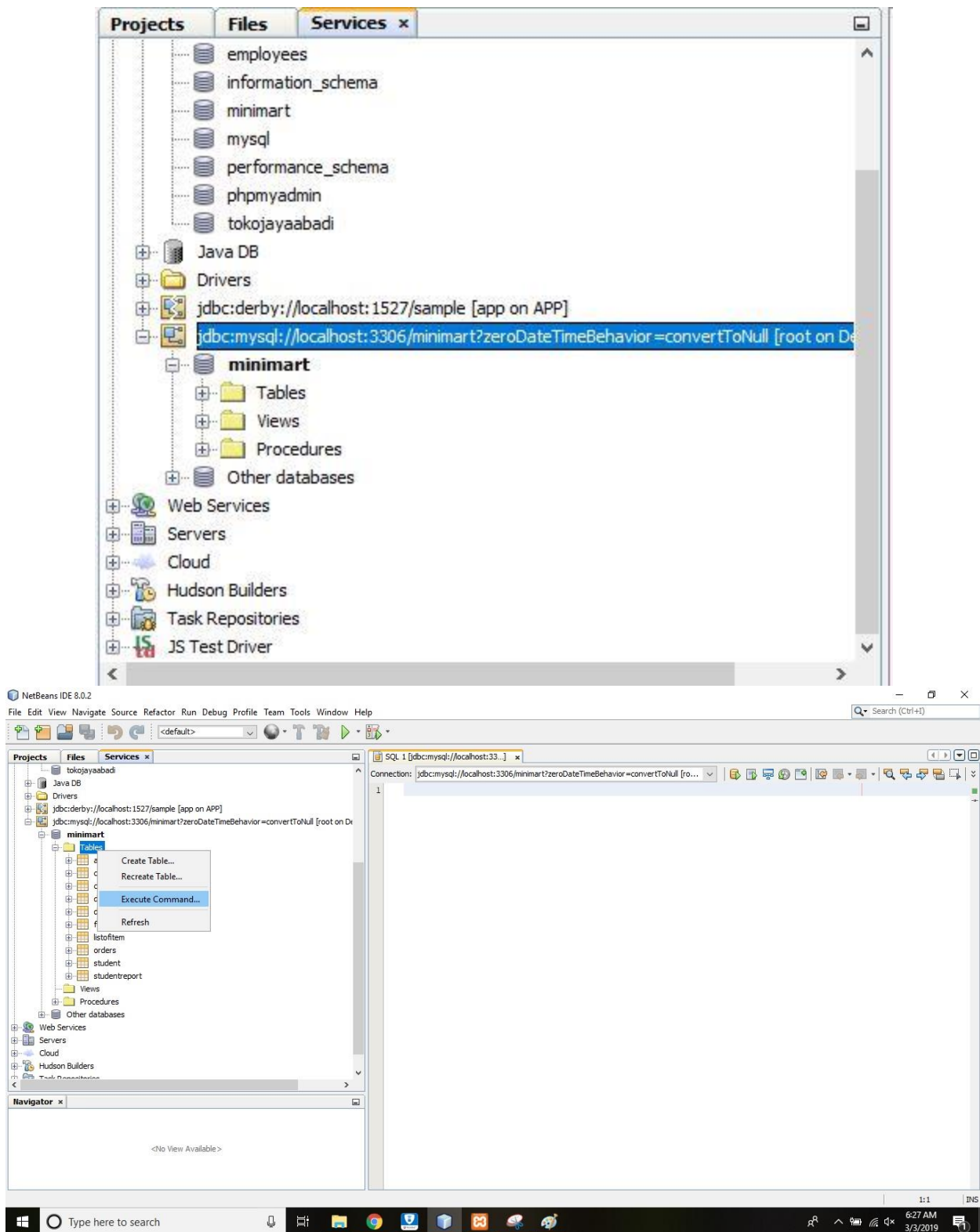
Next >

Finish

Cancel

Help





SQL 1 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/minimart?zeroDateTimeBehavior=convertToNull [ro...]

```
1 select * from agents;
```

t schema]

select * from agents x

Page Size: 20 | Total Rows: 12 | Page: 1 of 1 | Matching Rows:

#	AGENT_CODE	AGENT_NAME	WORKING_AREA	COMMISSION	PHONE_NO
1	A007	Ramasundar	Bangalore	0.15	077-25814763
2	A003	Alex	London	0.13	075-12458969
3	A008	Alford	New York	0.12	044-25874365
4	A011	Ravi Kumar	Bangalore	0.15	077-45625874
5	A010	Santakumar	Chennai	0.14	007-22388644
6	A012	Lucida	San Jose	0.12	044-52981425
7	A005	Anderson	Brisban	0.13	045-21447739
8	A001	Subbarao	Bangalore	0.14	077-12346674
9	A002	Mukesh	Mumbai	0.11	029-12358964
10	A006	McDen	London	0.15	078-22255588
11	A004	Ivan	Toronto	0.15	008-22544166
12	A009	Benjamin	Hampshair	0.11	008-22536178

Output - SQL 1 execution x

Executed successfully in 0.001 s.
Line 1, column 1

Execution finished after 0.001 s, 0 error(s) occurred.