



Topik

Data Manipulation Language (DML) pada DBMS MySQL

Tujuan

Mahasiswa diharapkan dapat:

1. Memahami penggunaan SQL statement INSERT.
2. Memahami penggunaan SQL statement UPDATE.
3. Memahami penggunaan SQL statement DELETE.

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.

Pada DML terdapat dua jenis bahasa, yaitu :

1. **High-Level(Non_procedural) DML.**
 - Digunakan secara interaktif (interpreter)
 - Dapat dijadikan satu dengan general purpose programming language (embedded)High-Level DML yang biasa digunakan secara interaktif disebut "Query Language".
2. **Low-Level(Proedural) DML.**
Digunakan secara embedded dalam suatu general purpose programming language

Bilamana kedua jenis DML diatas digunakan secara "embedded", maka : bahasa pemrograman yang digunakan disebut sebagai "Host Language" dan "DML-nya disebut "Sub Language"

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.

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.

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.
3. 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.

Praktikum – Bagian 1: Membuat Database untuk Percobaan

Langkah	Keterangan
1	Perhatikan skema/model relasional/EER diagram dari database berikut.
2	Skema tersebut adalah sekema database pada sebuah sistem informasi yang digunakan oleh suatu perusahaan yang bergerak di bidang pembuatan perangkat lunak. Pertama-tama, buatlah database tersebut dengan mengeksekusi baris-baris kode DDL berikut.

	<pre> -- ----- -- Create schema -- ----- DROP DATABASE IF EXISTS software_house; CREATE DATABASE software_house; USE software_house; -- ----- -- Table departemen -- ----- CREATE TABLE departemen (kode VARCHAR(3) NOT NULL, nama VARCHAR(45) UNIQUE NOT NULL, nik_manajer VARCHAR(10) NULL, PRIMARY KEY (kode)); </pre>
	<pre> -- ----- -- Table karyawan -- ----- CREATE TABLE karyawan (nik VARCHAR(10) NOT NULL, nama VARCHAR(100) NOT NULL, tanggal_lahir DATE NULL, jenis_kelamin ENUM('PRIA', 'WANITA') NULL, alamat VARCHAR(255) NULL, nomor_telepon VARCHAR(20) NOT NULL, email VARCHAR(100) UNIQUE NOT NULL, gaji DECIMAL(10,2) UNIQUE NULL, departemen_kode VARCHAR(3) NOT NULL, PRIMARY KEY (nik), FOREIGN KEY fk_karyawan_departemen_kode (departemen_kode) REFERENCES departemen (kode) ON DELETE CASCADE ON UPDATE CASCADE); </pre>

```

-- Table departemen FK nik_manajer
-----

ALTER TABLE departemen ADD
    FOREIGN KEY fk_departemen_karyawan_nik(nik_manajer)
        REFERENCES karyawan (nik)
        ON DELETE CASCADE
        ON UPDATE CASCADE;

-----

-- Table klien
-----

CREATE TABLE klien (
    nomor INT NOT NULL,
    nama VARCHAR(100) NOT NULL,
    alamat VARCHAR(255) NULL,
    jenis ENUM('INDIVIDU', 'INSTANSI') NOT NULL,
    nomor_telepon VARCHAR(20) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    PRIMARY KEY (nomor)
);

```

```

-- Table proyek
-----

CREATE TABLE proyek (
    nomor INT NOT NULL,
    nomor_klien INT NOT NULL,
    nama VARCHAR(100) NOT NULL,
    deskripsi TEXT NULL,
    tanggal_mulai DATE NULL,
    tanggal_selesai DATE NULL,
    PRIMARY KEY (nomor),
    FOREIGN KEY fk_proyek_klien_nomor (nomor_klien)
        REFERENCES klien (nomor)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

```

	<pre>-- Table penugasan CREATE TABLE penugasan (nomor_proyek INT NOT NULL, nik VARCHAR(10) NOT NULL, total_jam INT(3) NULL, PRIMARY KEY (nomor_proyek, nik), FOREIGN KEY fk_penugasan_proyek_nomor (nomor_proyek) REFERENCES proyek (nomor) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY fk_penugasan_karyawan_nik (nik) REFERENCES karyawan (nik) ON DELETE CASCADE ON UPDATE CASCADE);</pre>
3	Cek database Anda dengan perintah 'SHOW TABLES' untuk memastikan bahwa semua tabel sudah dibuat.
	<pre>[mysql> SHOW TABLES; +-----+ Tables_in_software_house +-----+ departemen karyawan klien penugasan proyek +-----+ 5 rows in set (0.00 sec)</pre>
4	Buat 1 tabel backup untuk tabel departemen bernama departemen_backup . Tabel ini akan bermanfaat ketika bagian INSERT dan DELETE kita kerjakan.
	<pre>CREATE TABLE departemen_backup (kode VARCHAR(3) PRIMARY KEY, nama VARCHAR (45), nik_manajer VARCHAR(10));</pre>
5	Cek strukturnya.

	<pre>[mysql> SHOW COLUMNS FROM departemen_backup;</pre> <table><tr><th>Field</th><th>Type</th><th>Null</th><th>Key</th><th>Default</th><th>Extra</th></tr><tr><td>kode</td><td>varchar(3)</td><td>NO</td><td>PRI</td><td>NULL</td><td></td></tr><tr><td>nama</td><td>varchar(45)</td><td>YES</td><td></td><td>NULL</td><td></td></tr><tr><td>nik_manajer</td><td>varchar(10)</td><td>YES</td><td></td><td>NULL</td><td></td></tr></table> <pre>3 rows in set (0.00 sec)</pre>	Field	Type	Null	Key	Default	Extra	kode	varchar(3)	NO	PRI	NULL		nama	varchar(45)	YES		NULL		nik_manajer	varchar(10)	YES		NULL	
Field	Type	Null	Key	Default	Extra																				
kode	varchar(3)	NO	PRI	NULL																					
nama	varchar(45)	YES		NULL																					
nik_manajer	varchar(10)	YES		NULL																					
6	Setelah selesai membuat database diatas, lanjutkan ke <u>Praktikum – Bagian 2</u> .																								

Praktikum – Bagian 2: Percobaan Statement INSERT

Langkah	Keterangan
1	Untuk menambahkan data (mengisi) suatu tabel, digunakan statement (pernyataan) INSERT . Eksekusi SQL berikut untuk menambahkan 1 baris (record) baru pada tabel departemen .
	<pre>INSERT INTO departemen (kode, nama, nik_manajer) VALUES ('MKK', 'Manajemen dan Kontrol Kualitas', NULL);</pre>
2	Statement SQL tersebut menambahkan 1 baris baru ke tabel departemen pada kolom yang dinyatakan di dalam tanda kurung () pertama. Untuk melihat hasil SQL yang kita eksekusi tersebut, gunakan statement SELECT seperti berikut.
	<pre>SELECT * FROM departemen;</pre>
3	Pembahasan lebih lengkap mengenai SELECT dijadwalkan untuk disampaikan pada pertemuan berikutnya, namun secara umum, statement SELECT digunakan untuk menyajikan record-record yang ada pada suatu tabel. Karakter * akan menampilkan isi dari semua kolom yang ada pada tabel. Statement SELECT tersebut menghasilkan:
	<pre>[mysql> SELECT * FROM departemen; +-----+-----+-----+ kode nama nik_manajer +-----+-----+-----+ MKK Manajemen dan Kontrol Kualitas NULL +-----+-----+-----+ 1 row in set (0.00 sec)</pre>
4	Apabila data di-insert-kan pada semua kolom tabel, maka kita dapat langsung menggunakan klausa VALUES tanpa harus menuliskan nama-nama kolom dahulu.
	<pre>INSERT INTO departemen VALUES ('RND', 'Riset dan Pengembangan', NULL);</pre>

	<pre>[mysql> SELECT * FROM departemen;</pre> <table><tr><th>kode</th><th>nama</th><th>nik_manajer</th></tr><tr><td>MKK</td><td>Manajemen dan Kontrol Kualitas</td><td>NULL</td></tr><tr><td>RND</td><td>Riset dan Pengembangan</td><td>NULL</td></tr></table> <pre>2 rows in set (0.00 sec)</pre>	kode	nama	nik_manajer	MKK	Manajemen dan Kontrol Kualitas	NULL	RND	Riset dan Pengembangan	NULL																		
kode	nama	nik_manajer																										
MKK	Manajemen dan Kontrol Kualitas	NULL																										
RND	Riset dan Pengembangan	NULL																										
5	Untuk menambahkan beberapa kolom sekaligus dalam 1 statement digunakan statement dengan format seperti berikut.																											
	<pre>INSERT INTO departemen VALUES ('DUX', 'Desain dan User Experience', NULL), ('MKT', 'Marketing', NULL), ('CSR', 'Customer Service and Relation', NULL);</pre>																											
6	Dan seperti berikut, jika hanya kolom tertentu saja yang akan diberi nilai.																											
	<pre>INSERT INTO departemen (kode, nama) VALUES ('PRD', 'Produksi'), ('HRD', 'Human Resource Development'), ('MGT', 'Top Level Management')</pre>																											
	<pre>[mysql> SELECT * FROM departemen;</pre> <table><tr><th>kode</th><th>nama</th><th>nik_manajer</th></tr><tr><td>CSR</td><td>Customer Service and Relation</td><td>NULL</td></tr><tr><td>DUX</td><td>Desain dan User Experience</td><td>NULL</td></tr><tr><td>HRD</td><td>Human Resource Development</td><td>NULL</td></tr><tr><td>MGT</td><td>Top Level Management</td><td>NULL</td></tr><tr><td>MKK</td><td>Manajemen dan Kontrol Kualitas</td><td>NULL</td></tr><tr><td>MKT</td><td>Marketing</td><td>NULL</td></tr><tr><td>PRD</td><td>Produksi</td><td>NULL</td></tr><tr><td>RND</td><td>Riset dan Pengembangan</td><td>NULL</td></tr></table> <pre>8 rows in set (0.00 sec)</pre>	kode	nama	nik_manajer	CSR	Customer Service and Relation	NULL	DUX	Desain dan User Experience	NULL	HRD	Human Resource Development	NULL	MGT	Top Level Management	NULL	MKK	Manajemen dan Kontrol Kualitas	NULL	MKT	Marketing	NULL	PRD	Produksi	NULL	RND	Riset dan Pengembangan	NULL
kode	nama	nik_manajer																										
CSR	Customer Service and Relation	NULL																										
DUX	Desain dan User Experience	NULL																										
HRD	Human Resource Development	NULL																										
MGT	Top Level Management	NULL																										
MKK	Manajemen dan Kontrol Kualitas	NULL																										
MKT	Marketing	NULL																										
PRD	Produksi	NULL																										
RND	Riset dan Pengembangan	NULL																										
7	Statement INSERT juga dapat dieksekusi dengan menggunakan klausa SET alih-alih VALUES.																											
	<pre>INSERT INTO departemen SET kode = 'PKU', nama = 'Pekerjaan Umum';</pre>																											

	<pre>[mysql> SELECT * FROM departemen;</pre> <table><tr><th>kode</th><th>nama</th><th>nik_manajer</th></tr><tr><td>CSR</td><td>Customer Service and Relation</td><td>NULL</td></tr><tr><td>DUX</td><td>Desain dan User Experience</td><td>NULL</td></tr><tr><td>HRD</td><td>Human Resource Development</td><td>NULL</td></tr><tr><td>MGT</td><td>Top Level Management</td><td>NULL</td></tr><tr><td>MKK</td><td>Manajemen dan Kontrol Kualitas</td><td>NULL</td></tr><tr><td>MKT</td><td>Marketing</td><td>NULL</td></tr><tr><td>PKU</td><td>Pekerjaan Umum</td><td>NULL</td></tr><tr><td>PRD</td><td>Produksi</td><td>NULL</td></tr><tr><td>RND</td><td>Riset dan Pengembangan</td><td>NULL</td></tr></table> <pre>9 rows in set (0.00 sec)</pre>	kode	nama	nik_manajer	CSR	Customer Service and Relation	NULL	DUX	Desain dan User Experience	NULL	HRD	Human Resource Development	NULL	MGT	Top Level Management	NULL	MKK	Manajemen dan Kontrol Kualitas	NULL	MKT	Marketing	NULL	PKU	Pekerjaan Umum	NULL	PRD	Produksi	NULL	RND	Riset dan Pengembangan	NULL
kode	nama	nik_manajer																													
CSR	Customer Service and Relation	NULL																													
DUX	Desain dan User Experience	NULL																													
HRD	Human Resource Development	NULL																													
MGT	Top Level Management	NULL																													
MKK	Manajemen dan Kontrol Kualitas	NULL																													
MKT	Marketing	NULL																													
PKU	Pekerjaan Umum	NULL																													
PRD	Produksi	NULL																													
RND	Riset dan Pengembangan	NULL																													
8	<p>Pada statement INSERT juga dapat digunakan klausa SELECT. Misalnya kita ingin menyalin semua baris pada tabel departemen ke tabel departemen_backup, maka kita SQL berikut dapat digunakan.</p>																														
	<pre>INSERT INTO departemen_backup SELECT * FROM departemen;</pre>																														
	<pre>[mysql> SELECT * FROM departemen_backup;</pre> <table><tr><th>kode</th><th>nama</th><th>nik_manajer</th></tr><tr><td>CSR</td><td>Customer Service and Relation</td><td>NULL</td></tr><tr><td>DUX</td><td>Desain dan User Experience</td><td>NULL</td></tr><tr><td>HRD</td><td>Human Resource Development</td><td>NULL</td></tr><tr><td>MGT</td><td>Top Level Management</td><td>NULL</td></tr><tr><td>MKK</td><td>Manajemen dan Kontrol Kualitas</td><td>NULL</td></tr><tr><td>MKT</td><td>Marketing</td><td>NULL</td></tr><tr><td>PKU</td><td>Pekerjaan Umum</td><td>NULL</td></tr><tr><td>PRD</td><td>Produksi</td><td>NULL</td></tr><tr><td>RND</td><td>Riset dan Pengembangan</td><td>NULL</td></tr></table> <pre>9 rows in set (0.00 sec)</pre>	kode	nama	nik_manajer	CSR	Customer Service and Relation	NULL	DUX	Desain dan User Experience	NULL	HRD	Human Resource Development	NULL	MGT	Top Level Management	NULL	MKK	Manajemen dan Kontrol Kualitas	NULL	MKT	Marketing	NULL	PKU	Pekerjaan Umum	NULL	PRD	Produksi	NULL	RND	Riset dan Pengembangan	NULL
kode	nama	nik_manajer																													
CSR	Customer Service and Relation	NULL																													
DUX	Desain dan User Experience	NULL																													
HRD	Human Resource Development	NULL																													
MGT	Top Level Management	NULL																													
MKK	Manajemen dan Kontrol Kualitas	NULL																													
MKT	Marketing	NULL																													
PKU	Pekerjaan Umum	NULL																													
PRD	Produksi	NULL																													
RND	Riset dan Pengembangan	NULL																													
9	<p>Sekarang penambahan data pada tabel karyawan dapat dilakukan karena data departemen sudah siap. Eksekusi SQL berikut ini untuk menambahkan beberapa data pegawai.</p>																														
	<pre>INSERT INTO karyawan VALUES ('741104013', 'Ahmad Husain', '1988-07-12', 'PRIA', 'Jl. Ikan Tombro No. 8', '0858992878', 'a.husain88@gmail.com', 100000, 'DUX'), ('741104019', 'Brian Anggriawan', '1987-12-9', 'PRIA', 'Jl. Bandung No. 2', '0857646577', 'brian.anggriawan@live.com', 89000, 'PKU');</pre>																														

	<pre>[mysql> SELECT * FROM karyawan; +-----+-----+-----+-----+-----+ nik nama tanggal_lahir jenis_kelamin alamat nomor_telepon email gaji departemen_kode +-----+-----+-----+-----+-----+ 741104013 Ahmad Husain 1988-07-12 PRIA Jl. Ikan Tombro No. 8 0858992878 a.husain88@gmail.com 100000.00 DUX 741104019 Brian Anggriawan 1987-12-09 PRIA Jl. Bandung No. 2 0857646577 brian.anggriawan@live.com 89000.00 PKU +-----+-----+-----+-----+-----+ 2 rows in set (0.00 sec)</pre>
10	Setelah berhasil mengeksekusi SQL tersebut, lanjutkan ke Praktikum - Bagian 3 .

Praktikum - Bagian 3: Percobaan Statement UPDATE

Langkah	Keterangan
1	UPDATE digunakan untuk mengubah nilai suatu baris pada sebuah tabel. Format dasar statement ini adalah sebagai berikut:
	UPDATE departemen SET nik_manajer = '741104013';
2	Statement tersebut mengubah nilai SEMUA baris dari tabel departemen pada kolom nik_manajer dengan nilai 741104013 . Apabila kita tampilkan isi tabel, maka sekarang semua manajer akan dikepalai oleh karyawan dengan nik tersebut.
	<pre>[mysql> SELECT * FROM departemen; +-----+-----+-----+ kode nama nik_manajer +-----+-----+-----+ CSR Customer Service and Relation 741104013 DUX Desain dan User Experience 741104013 HRD Human Resource Development 741104013 MGT Top Level Management 741104013 MKK Manajemen dan Kontrol Kualitas 741104013 MKT Marketing 741104013 PKU Pekerjaan Umum 741104013 PRD Produksi 741104013 RND Riset dan Pengembangan 741104013 +-----+-----+-----+ 9 rows in set (0.01 sec)</pre>
3	Untuk mengubah nilai pada baris tertentu saja, kita tambahkan klausa WHERE pada statement UPDATE. Misalkan kita akan menjadikan karyawan dengan nip 7411040719 sebagai manajer pada departemen PKU, maka dapat digunakan SQL sebagai berikut:

```
UPDATE departemen
SET nik_manajer = '741104019'
WHERE kode = 'PKU';
```

```
[mysql> SELECT * FROM departemen;
```

kode	nama	nik_manajer
CSR	Customer Service and Relation	741104013
DUX	Desain dan User Experience	741104013
HRD	Human Resource Development	741104013
MGT	Top Level Management	741104013
MKK	Manajemen dan Kontrol Kualitas	741104013
MKT	Marketing	741104013
PKU	Pekerjaan Umum	741104019
PRD	Produksi	741104013
RND	Riset dan Pengembangan	741104013

```
9 rows in set (0.00 sec)
```

4

Klause WHERE **tidak selalu** hanya membatasi UPDATE pada 1 baris saja, ia juga bisa memberlakukan UPDATE pada banyak baris sekaligus. Semuanya tergantung pada kondisi yang kita tentukan. Statement berikut ini akan mengosongkan **nik_manajer** untuk semua **departemen** yang dikepalai oleh karyawan dengan **nik 741104013**.

```
UPDATE departemen
SET nik_manajer = NULL
WHERE nik_manajer = '741104013';
```

```
[mysql> SELECT * FROM departemen;
```

kode	nama	nik_manajer
CSR	Customer Service and Relation	NULL
DUX	Desain dan User Experience	NULL
HRD	Human Resource Development	NULL
MGT	Top Level Management	NULL
MKK	Manajemen dan Kontrol Kualitas	NULL
MKT	Marketing	NULL
PKU	Pekerjaan Umum	741104019
PRD	Produksi	NULL
RND	Riset dan Pengembangan	NULL

```
9 rows in set (0.00 sec)
```

5

Untuk mengubah **beberapa kolom sekaligus** dalam satu kali eksekusi statement UPDATE, dapat digunakan format berikut.

	<pre>UPDATE departemen SET nama = 'Desain dan Pengalaman Pengguna', nik_manajer = '741104013' WHERE kode = 'DUX';</pre>
	<pre>[mysql> SELECT * FROM departemen; +-----+-----+-----+ kode nama nik_manajer +-----+-----+-----+ CSR Customer Service and Relation NULL DUX Desain dan Pengalaman Pengguna 741104013 HRD Human Resource Development NULL MGT Top Level Management NULL MKK Manajemen dan Kontrol Kualitas NULL MKT Marketing NULL PKU Pekerjaan Umum 741104019 PRD Produksi NULL RND Riset dan Pengembangan NULL +-----+-----+-----+ 9 rows in set (0.00 sec)</pre>
6	<p>Kita juga dapat menggunakan statement UPDATE dengan SELECT. Misalkan kita ingin mengeset nik_manajer departemen 'RND' dengan nik karyawan yang bernama 'Ahmad Husain', maka dapat digunakan SQL dengan format berikut.</p>
	<pre>UPDATE departemen SET nik_manajer = (SELECT nik FROM karyawan WHERE nama = 'Ahmad Husain') WHERE kode = 'RND';</pre>
	<pre>[mysql> SELECT * FROM departemen; +-----+-----+-----+ kode nama nik_manajer +-----+-----+-----+ CSR Customer Service and Relation NULL DUX Desain dan Pengalaman Pengguna 741104013 HRD Human Resource Development NULL MGT Top Level Management NULL MKK Manajemen dan Kontrol Kualitas NULL MKT Marketing NULL PKU Pekerjaan Umum 741104019 PRD Produksi NULL RND Riset dan Pengembangan 741104013 +-----+-----+-----+ 9 rows in set (0.00 sec)</pre>
7	<p>Setelah berhasil mengeksekusi SQL tersebut, lanjutkan ke Praktikum - Bagian 4.</p>

Praktikum - Bagian 4: Percobaan Statement DELETE

Langkah	Keterangan																					
1	DELETE digunakan untuk menghapus satu atau lebih baris dari sebuah tabel. Misalkan kita ingin menghapus departemen yang memiliki nilai pada kolom nik_manajernya, maka format dasar statement seperti berikut dapat kita gunakan:																					
	<code>DELETE FROM departemen WHERE nik_manajer IS NOT NULL;</code>																					
	<pre>[mysql> SELECT * FROM departemen;</pre> <table><tr><th>kode</th><th>nama</th><th>nik_manajer</th></tr><tr><td>CSR</td><td>Customer Service and Relation</td><td>NULL</td></tr><tr><td>HRD</td><td>Human Resource Development</td><td>NULL</td></tr><tr><td>MGT</td><td>Top Level Management</td><td>NULL</td></tr><tr><td>MKK</td><td>Manajemen dan Kontrol Kualitas</td><td>NULL</td></tr><tr><td>MKT</td><td>Marketing</td><td>NULL</td></tr><tr><td>PRD</td><td>Produksi</td><td>NULL</td></tr></table> <pre>6 rows in set (0.00 sec)</pre>	kode	nama	nik_manajer	CSR	Customer Service and Relation	NULL	HRD	Human Resource Development	NULL	MGT	Top Level Management	NULL	MKK	Manajemen dan Kontrol Kualitas	NULL	MKT	Marketing	NULL	PRD	Produksi	NULL
kode	nama	nik_manajer																				
CSR	Customer Service and Relation	NULL																				
HRD	Human Resource Development	NULL																				
MGT	Top Level Management	NULL																				
MKK	Manajemen dan Kontrol Kualitas	NULL																				
MKT	Marketing	NULL																				
PRD	Produksi	NULL																				
2	HATI-HATI apabila kita menggunakan statement DELETE tanpa WHERE! Cobalah eksekusi syntax SQL berikut:																					
	<code>DELETE FROM departemen;</code>																					
3	Semua data dalam satu tabel departemen akan hilang!																					
	<pre>[mysql> SELECT * FROM departemen;</pre> <p>Empty set (0.00 sec)</p>																					
4	Lanjutkan ke bagian Tugas!																					

Tugas

- Pastikan isi pada tabel-tabel Anda sesuai dengan kondisi akhir pada bagian **Praktikum**.
- Himpunlah *screenshot* sintaksis SQL berikut hasil SELECT yang Anda dapat di komputer Anda pada bagian **Praktikum** dalam sebuah laporan dengan format **PDF**!
Untuk nomor 2 dan seterusnya, buatlah juga screenshot sintaksis SQL berikut hasilnya (SELECT).
- Kembalikan lagi data yang hilang pada tabel **departemen** dengan cara mengisi data yang sudah ada pada tabel **departemen_backup**!
- Tambahkan sedikitnya 10 (sepuluh) data karyawan dengan ketentuan:
 - 4 orang berjenis kelamin PRIA.
 - 6 orang lainnya WANITA.
 - Masing-masing departemen ditempati oleh TEPAT 1 orang Karyawan **KECUALI**:
 - Departemen dengan kode CSR dimana pada departemen ini harus ada TEPAT 3 orang karyawan berjenis kelamin WANITA
 - Departemen HRD dimana pada departemen ini anggotanya TEPAT 2 orang berjenis kelamin 1 PRIA dan 1 WANITA.
- Buatlah SEMUA departemen sedemikian sehingga masing-masing memiliki manajer dimana manajernya haruslah karyawan yang bekerja pada departemen tersebut!

6. Tambahkan sedikitnya 5 **proyek** baru dengan 5 **klien** berbeda dimana pada masing-masing proyek minimal terdapat 3 **penugasan**.
7. Backup tabel departemen dengan terlebih dahulu menghapus semua nilai pada tabel **departemen_backup**, kemudian salin semua isi tabel **departemen** kedalam tabel tersebut.
8. Cobalah untuk menghapus semua **departemen** yang pada namanya ada kata '**dan**' hanya dengan 1 statement DELETE.

-- Selamat Mengerjakan --

Daftar Pustaka

- Dwi Puspitasari, S.Kom, "**Buku Ajar Dasar Basis Data**", *Program Studi Manajemen Informatika Politeknik Negeri Malang*, 2012.