

Laporan Praktikum Modul 4

Pertama kita membuat sebuah database bernama *toko_kain* sebagai berikut :

```
1 CREATE DATABASE toko_kain;
2 USE toko_kain;
3
4 CREATE TABLE supplier (
5     id_supplier INT PRIMARY KEY,
6     nama_supplier VARCHAR(50),
7     alamat_supplier VARCHAR(100)
8 );
9
10 CREATE TABLE produk (
11     id_produk INT PRIMARY KEY,
12     nama_produk VARCHAR(50),
13     harga_produk DECIMAL(10, 2)
14 );
15
16 CREATE TABLE kategori_produk (
17     id_kategori INT PRIMARY KEY,
18     nama_kategori VARCHAR(50)
19 );
20
21 CREATE TABLE stok_produk (
22     id_produk INT,
23     stok_masuk INT,
24     stok_keluar INT,
25     FOREIGN KEY (id_produk) REFERENCES produk(id_produk)
26 );
27
28 CREATE TABLE penjualan (
29     id_penjualan INT PRIMARY KEY,
30     tanggal_penjualan DATE,
```

```
31  id_produk INT,  
32  jumlah INT,  
33  total_harga DECIMAL(10, 2),  
34  FOREIGN KEY (id_produk) REFERENCES produk(id_produk)  
35 );  
36  
37 CREATE TABLE pembelian (  
38  id_pembelian INT PRIMARY KEY,  
39  tanggal_pembelian DATE,  
40  id_supplier INT,  
41  id_produk INT,  
42  jumlah INT,  
43  total_harga DECIMAL(10, 2),  
44  FOREIGN KEY (id_supplier) REFERENCES supplier(id_supplier),  
45  FOREIGN KEY (id_produk) REFERENCES produk(id_produk)  
46 );  
47  
48 CREATE TABLE pelanggan (  
49  id_pelanggan INT PRIMARY KEY,  
50  nama_pelanggan VARCHAR(50),  
51  alamat VARCHAR(100)  
52 );  
53  
54 INSERT INTO supplier (id_supplier, nama_supplier, alamat_supplier)  
55 VALUES (1, 'Supplier A', 'Jl. Supplier A No. 123'),  
56         (2, 'Supplier B', 'Jl. Supplier B No. 456'),  
57         (3, 'Supplier C', 'Jl. Supplier C No. 789'),  
58         (4, 'Supplier D', 'Jl. Supplier D No. 012'),  
59         (5, 'Supplier E', 'Jl. Supplier E No. 345'),  
60         (6, 'Supplier F', 'Jl. Supplier F No. 678'),  
61         (7, 'Supplier G', 'Jl. Supplier G No. 901'),  
62         (8, 'Supplier H', 'Jl. Supplier H No. 234'),  
63         (9, 'Supplier I', 'Jl. Supplier I No. 567'),  
64         (10, 'Supplier J', 'Jl. Supplier J No. 890');  
65
```

```
66 INSERT INTO produk (id_produk, nama_produk, harga_produk)
67 VALUES (1, 'Kain Katun', 50.00),
68         (2, 'Kain Sutra', 100.00),
69         (3, 'Kain Denim', 75.00),
70         (4, 'Kain Linen', 90.00),
71         (5, 'Kain Wol', 120.00),
72         (6, 'Kain Satin', 80.00),
73         (7, 'Kain Flanel', 60.00),
74         (8, 'Kain Batik', 70.00),
75         (9, 'Kain Tenun', 110.00),
76         (10, 'Kain Rayon', 65.00);
77
78 INSERT INTO kategori_produk (id_kategori, nama_kategori)
79 VALUES (1, 'Kategori 1'),
80         (2, 'Kategori 2'),
81         (3, 'Kategori 3'),
82         (4, 'Kategori 4'),
83         (5, 'Kategori 5'),
84         (6, 'Kategori 6'),
85         (7, 'Kategori 7'),
86         (8, 'Kategori 8'),
87         (9, 'Kategori 9'),
88         (10, 'Kategori 10');
89
90 INSERT INTO stok_produk (id_produk, stok_masuk, stok_keluar)
91 VALUES (1, 100, 50),
92         (2, 200, 100),
93         (3, 150, 80),
94         (4, 120, 70),
95         (5, 180, 90),
96         (6, 140, 60),
97         (7, 110, 40),
98         (8, 130, 55),
99         (9, 160, 75),
100        (10, 90, 30);
101
```

```

102 INSERT INTO penjualan (id_penjualan, tanggal_penjualan, id_produk, jumlah, total_harga)
103 VALUES (1, '2023-06-01', 1, 10, 500.00),
104          (2, '2023-06-02', 2, 5, 500.00),
105          (3, '2023-06-03', 3, 8, 600.00),
106          (4, '2023-06-04', 4, 7, 630.00),
107          (5, '2023-06-05', 5, 12, 1440.00),
108          (6, '2023-06-06', 6, 6, 480.00),
109          (7, '2023-06-07', 7, 9, 540.00),
110          (8, '2023-06-08', 8, 4, 280.00),
111          (9, '2023-06-09', 9, 11, 1210.00),
112          (10, '2023-06-10', 10, 3, 195.00);
113
114 INSERT INTO pembelian (id_pembelian, tanggal_pembelian, id_supplier, id_produk, jumlah, total_harga)
115 VALUES (1, '2023-06-01', 1, 1, 20, 1000.00),
116          (2, '2023-06-02', 2, 2, 10, 1000.00),
117          (3, '2023-06-03', 3, 3, 15, 1125.00),
118          (4, '2023-06-04', 4, 4, 8, 720.00),
119          (5, '2023-06-05', 5, 5, 25, 3000.00),
120          (6, '2023-06-06', 6, 6, 12, 960.00),
121          (7, '2023-06-07', 7, 7, 18, 1080.00),
122          (8, '2023-06-08', 8, 8, 6, 420.00),
123          (9, '2023-06-09', 9, 9, 22, 2420.00),
124          (10, '2023-06-10', 10, 10, 5, 325.00);
125
126 INSERT INTO pelanggan (id_pelanggan, nama_pelanggan, alamat)
127 VALUES (1, 'Pelanggan A', 'Jl. Pelanggan A No. 123'),
128          (2, 'Pelanggan B', 'Jl. Pelanggan B No. 456'),
129          (3, 'Pelanggan C', 'Jl. Pelanggan C No. 789'),
130          (4, 'Pelanggan D', 'Jl. Pelanggan D No. 012'),
131          (5, 'Pelanggan E', 'Jl. Pelanggan E No. 345'),
132          (6, 'Pelanggan F', 'Jl. Pelanggan F No. 678'),
133          (7, 'Pelanggan G', 'Jl. Pelanggan G No. 901'),
134          (8, 'Pelanggan H', 'Jl. Pelanggan H No. 234'),
135          (9, 'Pelanggan I', 'Jl. Pelanggan I No. 567'),
136          (10, 'Pelanggan J', 'Jl. Pelanggan J No. 890');

```

Soal dengan npm (genap) :

1. TRIGGER AFTER DELETE

```

1 -- Membuat tabel log untuk menyimpan entri log
2 CREATE TABLE log (
3     id INT AUTO_INCREMENT PRIMARY KEY,
4     event VARCHAR(100)
5 );
6
7 -- Membuat TRIGGER AFTER DELETE
8 DELIMITER //
9 CREATE TRIGGER produk_after_delete
10 AFTER DELETE ON produk
11 FOR EACH ROW
12 BEGIN
13     INSERT INTO log (event) VALUES (CONCAT('Produk dihapus: ', OLD.nama_produk));
14 END //
15 DELIMITER ;

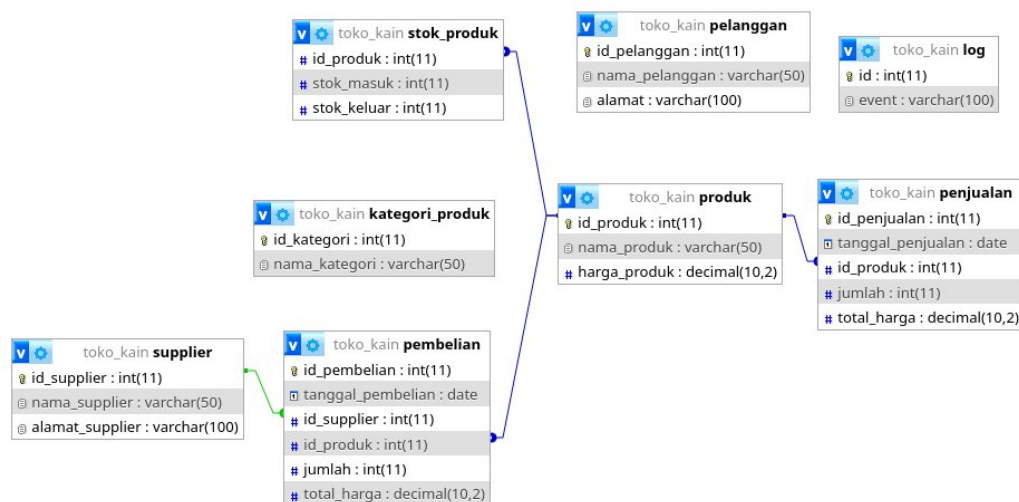
```

Pada saat melakukan *DELETE* terjadi sebuah error

```
sudo /opt/lampp/bin/mysql -u root

MariaDB [toko_kain]> DELETE FROM produk WHERE id_produk = 1;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`toko_kain`.`pembelian`, CONSTRAINT `pembelian_ibfk_2` FOREIGN KEY (`id_produk`) REFERENCES `produk` (`id_produk`))
MariaDB [toko_kain]>
```

Hal itu terjadi karena pada *TABLE* produk terdapat item yang menjadi *FOREIGN KEY* pada tabel lain



Cara untuk menangani hal tersebut yaitu menghapus data yang menjadi *FOREIGN KEY* pada setiap tabel-nya

```
sudo /opt/lampp/bin/mysql -u root

MariaDB [toko_kain]> DELETE FROM pembelian WHERE id_produk = 1;
Query OK, 1 row affected (0,010 sec)

MariaDB [toko_kain]> DELETE FROM stok_produk WHERE id_produk = 1;
Query OK, 1 row affected (0,010 sec)

MariaDB [toko_kain]> DELETE FROM penjualan WHERE id_produk = 1;
Query OK, 1 row affected (0,008 sec)

MariaDB [toko_kain]> DELETE FROM produk WHERE id_produk = 1;
Query OK, 1 row affected (0,009 sec)

MariaDB [toko_kain]>
```


2. TRIGGER BEFORE UPDATE

```
1 -- membuat TRIGGER BEFORE UPDATE:
2 DELIMITER //
3 CREATE TRIGGER produk_before_update
4 BEFORE UPDATE ON produk
5 FOR EACH ROW
6 BEGIN
7     IF NEW.harga_produk = 0 THEN
8         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Harga produk tidak dapat diubah menjadi 0.';
9     END IF;
10 END //
11 DELIMITER ;
```

Pada *TRIGGER* diatas menggunakan *if statment* untuk menguji apakah nilai `harga_produk == 0`, jika iya maka `SQLSTATE` akan memberi `SIGNAL` berupa message yang bertuliskan “Harga produk tidak dapat diubah menjadi 0”.

Pesan kesalahan `ERROR 1644 (45000)` pada MySQL menunjukkan adanya kesalahan atau pengecualian yang dikendalikan oleh pengguna (*user-defined error*). Kode status `'45000'` pada pesan kesalahan tersebut menunjukkan bahwa ini adalah jenis pengecualian umum.

Output :

```
sudo /opt/lampp/bin/mysql -u root

MariaDB [toko_kain]> UPDATE produk SET harga_produk = 0 WHERE id_produk = 2;

ERROR 1644 (45000): Harga produk tidak dapat diubah menjadi 0.
MariaDB [toko_kain]>
```