

BAB 14

TRANSACTION

14.1. Tujuan Pembelajaran

1. Membuat transaction
2. Mengetahui cara kerja transaction
3. Mengetahui kegunaan transaction

14.2. Dasar Teori

Pengenalan Transaction

Untuk memahami apa itu transaksi di MySQL, mari kita lihat contoh penambahan pesanan penjualan baru di database yang sudah pernah dibuat. Langkah-langkah penambahan sales order dijelaskan sebagai berikut:

1. Pertama, kueri nomor pesanan penjualan terbaru dari tabel pesanan(orders) menggunakan nomor pesanan penjualan berikutnya sebagai nomor pesanan penjualan baru.
2. Selanjutnya, masukkan pesanan penjualan baru ke dalam tabel pesanan(orders).
3. Kemudian, dapatkan nomor pesanan penjualan yang baru dimasukkan
4. Setelah itu, masukkan item pesanan penjualan baru ke dalam tabel detail pesanan(detailsorders) dengan nomor pesanan penjualan
5. Terakhir, pilih data dari tabel order dan orderdetails untuk mengonfirmasi perubahan

Sekarang, bayangkan apa yang akan terjadi pada data pesanan penjualan jika satu atau lebih langkah di atas gagal karena beberapa alasan seperti table locking? Misalnya, jika langkah menambahkan item pesanan ke tabel detail pesanan gagal, kalian akan memiliki pesanan penjualan kosong.

Itulah mengapa pemrosesan transaksi datang untuk menyelamatkan. Transaksi MySQL memungkinkan kalian untuk menjalankan serangkaian operasi MySQL untuk memastikan bahwa database tidak pernah berisi hasil operasi parsial. Dalam satu set operasi, jika salah satunya gagal, rollback terjadi untuk memulihkan database ke keadaan semula. Jika tidak ada kesalahan yang terjadi, seluruh rangkaian pernyataan dikomit ke database.

MySQL Transaction Statement

MySQL memberi kami pernyataan penting berikut untuk mengontrol transaksi:

1. Untuk memulai transaksi, kalian menggunakan pernyataan MULAI TRANSAKSI. BEGIN atau BEGIN WORK adalah alias dari MULAI TRANSAKSI.
2. Untuk melakukan transaksi saat ini dan membuatnya permanen, kalian menggunakan pernyataan COMMIT.

3. Untuk memutar kembali transaksi saat ini dan membatalkan perubahannya, kalian menggunakan pernyataan ROLLBACK.
4. Untuk menonaktifkan atau mengaktifkan mode komitmen otomatis untuk transaksi saat ini, kalian menggunakan pernyataan SET komitmen otomatis.

Secara default, MySQL secara otomatis melakukan perubahan secara permanen ke database. Untuk memaksa MySQL agar tidak melakukan perubahan secara otomatis, kalian gunakan pernyataan berikut:

```
SET autocommit = 0;
```

Gambar 14.1 Mengatur Autocommit Secara Otomatis

Kalian menggunakan pernyataan berikut untuk mengaktifkan mode autocommit secara eksplisit:

```
SET autocommit = 1;
```

Gambar 14.2 Mengatur Autocommit Secara Eksplisit

14.3. Software

- XAMPP
- MySQL

14.4. Transaction

14.4.1 Latihan membuat transaction

Kita akan menggunakan tabel order dan orderDetails dari database yang sudah kita buat sebelumnya.

Contoh Commit

Untuk menggunakan transaksi, pertama-tama Anda harus memecah pernyataan SQL menjadi bagian logis dan menentukan kapan data harus di commit atau dibatalkan.

Berikut gambaran langkah pembuatan pesanan penjualan baru:

1. Pertama, mulai transaksi dengan menggunakan pernyataan MULAI TRANSAKSI.
2. Selanjutnya, pilih nomor pesanan penjualan terbaru dari tabel pesanan dan gunakan nomor pesanan penjualan berikutnya sebagai nomor pesanan penjualan yang baru.
3. Kemudian, masukkan pesanan penjualan baru ke dalam tabel pesanan.
4. Setelah itu, masukkan item pesanan penjualan ke dalam tabel detail pesanan.
5. Terakhir, lakukan transaksi menggunakan pernyataan COMMIT.

Secara opsional, Anda dapat memilih data dari tabel pesanan dan detail pesanan untuk memeriksa pesanan penjualan baru.

Berikut ini adalah script yang melakukan langkah-langkah di atas:

```
START TRANSACTION;
-- 2. Get the latest order number
SELECT
    @orderNumber:=MAX(orderNumber)+2
FROM
    orders;
-- 3. insert a new order for customer
INSERT INTO orders(
    orderNumber,
    customerNumber,
    orderDate,
    comment,
    status
)
VALUES(@orderNumber,38475, "2022-09-10", "Memuaskan", "terkirim");
-- 4. Insert order line items
INSERT INTO orderdetails(
    orderNumber,
    productCode,
    quantityOrdered,
    priceEach)
VALUES(@orderNumber,'amd02', 30, 100000);
-- 5. commit changes
COMMIT;
```

Gambar 14.3 Memulai Transaction dan Memberi Nilai

Outputnya akan menampilkan:

@orderNumber:=MAX(orderNumber)+2
93847

Gambar 14.4 Menampilkan Nilai Terbesar

Untuk mendapatkan pesanan penjualan yang baru dibuat, Anda menggunakan kueri berikut:

```
SELECT
    orderNumber,
    orderDate,
    status,
    comment,
    customerNumber
FROM
    orders a
    INNER JOIN
    orderdetails b USING (orderNumber)
WHERE
    a.orderNumber = 93847;
```

Gambar 14.5 Menampilkan Table Order digabung dengan Tabel OrderDetails

Outputnya akan menampilkan:

orderNumber	orderDate	status	comment	customerNumber
93847	2022-09-10	terkirim	Memuaskan	38475

Gambar 14.6 Tampilan Output Ketika Sudah Digabungkan

Contoh Rollback

Pertama, masuk ke server database MySQL dan hapus data dari tabel orders:

```
START TRANSACTION;
DELETE FROM orders;
```

Gambar 14.7 Menghapus Data Tabel Orders

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,1199 detik.)

START TRANSACTION;

[Edit kotak] [Ubah] [Buat kode PHP]

✓ 9 baris terpengaruh. (Pencarian dilakukan dalam 0,1199 detik.)

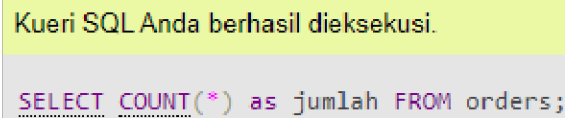
DELETE FROM orders;

[Edit kotak] [Ubah] [Buat kode PHP]

Gambar 14.8 Query Berhasil Menghapus Data Tabel Orders

Seperti yang Anda lihat dari output, MySQL mengonfirmasi bahwa semua baris dari tabel pesanan telah dihapus.

Kedua, masuk ke server database MySQL dalam sesi terpisah dan kueri data dari tabel orders.



```
Kueri SQL Anda berhasil dieksekusi.  
  
SELECT COUNT(*) as jumlah FROM orders;
```

Gambar 14.9 Menampilkan Penjumlahan Tabel Orders

Outputnya akan menampilkan:



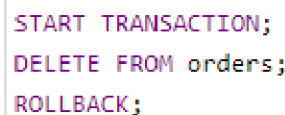
jumlah
9

Gambar 14.10 Output Penjumlahan Tabel Orders

Artinya ada 9 record atau data di dalam tabel orders.

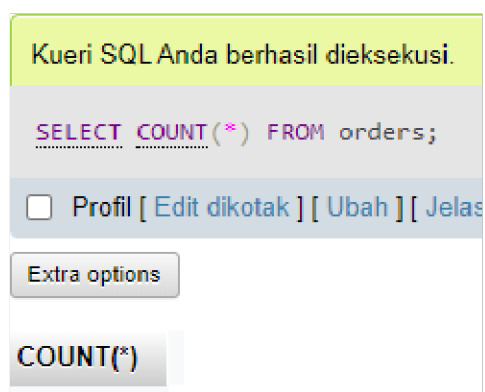
Pada sesi kedua ini, kita masih bisa melihat data dari tabel order. Kita telah membuat perubahan di sesi pertama. Namun, perubahan tersebut tidak permanen. Di sesi pertama, kita dapat melakukan atau membatalkan perubahan.

Menggunakan Transaction



```
START TRANSACTION;  
DELETE FROM orders;  
ROLLBACK;
```

Gambar 14.11 Query Memulai Transaction dan Menghapus Tabel Orders



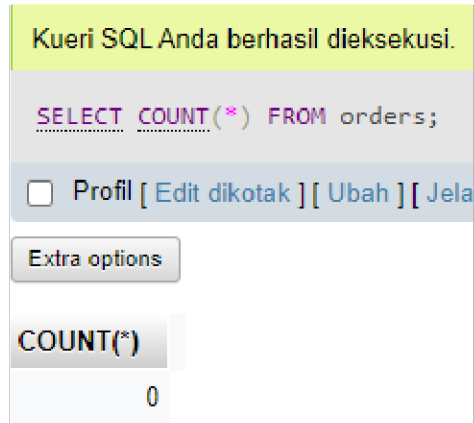
```
Kueri SQL Anda berhasil dieksekusi.  
  
SELECT COUNT(*) FROM orders;  
  
COUNT(*)
```

Gambar 14.12 Tampilan Setelah Query Dieksekusi Menggunakan Transaction

Tidak Menggunakan Transaction

```
DELETE FROM orders;
SELECT COUNT(*) FROM orders;
```

Gambar 14.13 Query Menghapus Tabel Orders

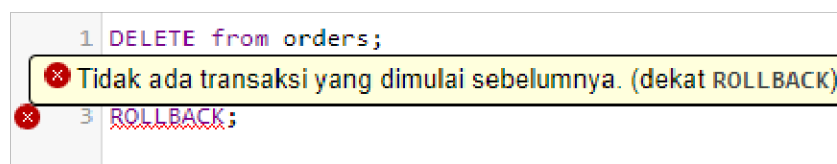


Gambar 14.14 Tampilan Setelah Query Dieksekusi Tanpa Transaction

Penjelasan dari hasil query diatas yaitu, saat menggunakan transaction kita membuat perubahan sementara, yang dimana kita menghapus data atau record di tabel orders (hanya sementara).

Lalu saat kita menggunakan transaction, kita masih bisa melihat data tersebut masih tersimpan karena belum kita commit. Jika data tersebut terhapus, kita bisa menggunakan rollback untuk mengembalikan data ke keadaan semula.

Berbeda dengan case kedua, saat kita hapus data tanpa menggunakan keyword transaction, maka data akan secara permanen terhapus, dan saat kita ingin melakukan rollback maka SQL akan menampilkan error karena kita belum mendefinisikan sebagai transaction. (jika kalian langsung menghapus tabel orders maka akan menampilkan seperti ini)



Gambar 14.15 Error Transaction

SQL memberitahu kita kalau kita tidak bisa menggunakan rollback tanpa start transaction terlebih dahulu, jika kita nekat maka data kita akan terhapus. (contoh seperti gambar dibawahini)

```
DELETE from orders;  
SELECT * from orders;
```

Gambar 14.16 Menghapus dan Menampilkan Tabel Orders

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0003 detik.)

`SELECT * from orders;`

☐ Profil [[Edit kotak](#)] [[Ubah](#)] [[Jelaskan SQL](#)] [[Buat kode PHP](#)] [[Segarkan](#)]

orderNumber	customerNumber	orderDate	comment	status
-------------	----------------	-----------	---------	--------

Operasi hasil kueri

Gambar 14.17 Tampilan Tabel Orders

Data didalam tabel orders akan menghilang secara permanen.

14.5. Latihan

1. Jelaskan menurut kalian perbedaan menggunakan transaction dan tidak menggunakan transaction!
2. Simpulkan kapan waktu yang tepat menggunakan transaction!