

11/20/2024

BASIS DATA

PERTEMUAN X

NAMA: RIFANDY ARNAS

NIM/NPM: 232310001

KELAS: TI – 23 – PA (LAB 2)

Tugas Pertemuan Ke-10
TEMPORARY TABLE

Github: https://github.com/Rifandy232310001TeknologiInformasi/Basis_Data-Lab

Soal nomor 1

CREATE VIEW rata_rata_pembelian_pelanggan AS

SELECT

 pelanggan.id_pelanggan,

 pelanggan.nama,

 AVG(detail_pesanan.total_harga) AS rata_rata_pembelian

FROM

 pelanggan

JOIN

 pesanan ON pelanggan.id_pelanggan = pesanan.id_pelanggan

JOIN

 detail_pesanan ON pesanan.id_pesanan = detail_pesanan.id_pesanan

GROUP BY

 pelanggan.id_pelanggan, pelanggan.nama;

Soal nomor 2

CREATE VIEW view_pesanan_detail AS

SELECT

 detail_pesanan.id_pesanan,

 detail_pesanan.id_produk,

 detail_pesanan.jumlah,

 detail_pesanan.harga_satuan,

 (detail_pesanan.jumlah * detail_pesanan.harga_satuan) AS total_harga

FROM

detail_pesanan

JOIN

produk ON detail_pesanan.id_produk = produk.id_produk

JOIN

pesanan ON detail_pesanan.id_pesanan = pesanan.id_pesanan;

Soal nomor 3

INSERT INTO detail_pesanan (id_detail_pesanan, id_pesanan, id_produk, jumlah, harga_satuan, total_harga)

VALUES (6, 5, 5, 2, 7000000, 14000000);

Soal nomor 4

Jika data di tabel detail_pesanan diubah, misalnya ada baris data yang diperbarui atau dihapus, maka **view akan secara otomatis mencerminkan perubahan tersebut**, karena view hanyalah representasi dinamis dari tabel yang dirujuknya.

Soal nomor 5

SELECT

SUM(total_harga) AS total_pesanan

FROM

view_pesanan_detail

WHERE

id_pesanan IN (SELECT id_pesanan FROM pesanan WHERE id_pelanggan = 2);

Soal nomor 6

Penggunaan view mempermudah pengelolaan data. Berikut ini adalah keunggulan view!

1. **Abstraksi:** View menyembunyikan struktur tabel yang kompleks sehingga query menjadi lebih sederhana.
2. **Keamanan:** Data sensitif dapat dilindungi dengan memberikan akses hanya ke view tertentu.
3. **Pemeliharaan Mudah:** Perubahan pada struktur tabel cukup dilakukan di view tanpa harus memodifikasi query di berbagai tempat.

Soal nomor 7

```
CREATE TEMPORARY TABLE stok_kritis AS
```

```
SELECT
```

```
    id_produk, nama_produk, stok
```

```
FROM
```

```
    produk
```

```
WHERE
```

```
    stok < 5;
```

Soal nomor 8

Membuat dan mengisi temporary table!

```
CREATE TEMPORARY TABLE produk_baru (
```

```
    id_produk INT,
```

```
    nama_produk VARCHAR(100),
```

```
    harga_jual DECIMAL(10, 2),
```

```
    stok INT
```

```
);
```

```
INSERT INTO produk_baru (id_produk, nama_produk, harga_jual, stok)
```

```
VALUES
```

```
(101, 'Laptop Pro', 15000000, 10),
```

```
(102, 'Mouse Wireless', 250000, 50),
```

```
(103, 'Keyboard Mechanical', 1200000, 15);
```

Memindahkan data dari produk_baru ke tabel master!

```
INSERT INTO produk (id_produk, nama_produk, harga_jual, stok)
```

```
SELECT id_produk, nama_produk, harga_jual, stok
```

```
FROM produk_baru;
```

Menampilkan tabel produk sebelum dan sesudah!

```
-- Sebelum update
```

```
SELECT * FROM produk;
```

```
-- Sesudah update
```

```
INSERT INTO produk SELECT * FROM produk_baru;
```

```
SELECT * FROM produk;
```

Soal nomor 9

```
CREATE TEMPORARY TABLE temp_pelanggan_baru (
```

```
id_pelanggan INT,
```

```
nama_depan VARCHAR(50),
```

```
nama_belakang VARCHAR(50),
```

```
email VARCHAR(100),
```

```
nomor_telepon VARCHAR(15)
```

);

```
INSERT INTO temp_pelanggan_baru (id_pelanggan, nama_depan, nama_belakang, email,  
nomor_telepon)
```

```
VALUES
```

```
(102, 'Ahmad', 'Semit', 'ahmadsemit@example.com', '081234567890');
```

Soal nomor 10

Temporary table seperti temp_pelanggan_baru hanya berlaku selama sesi SQL berjalan. **Setelah sesi SQL selesai, temporary table akan otomatis dihapus**, dan datanya tidak lagi tersedia.