11/20/2024

Tugas Pertemuan Ke-10

temporary table

Nama: rifandy arnas

nim/npm: 232310001

Kelas: TI – 23 – PA (Lab 2)

BASIS DATA

PERTEMUAN X

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.