Nama : Rafael Nicholas Tanaja

NIM : 2540118656

**Esai**

1. GROUP BY merupakan fungsi aggregate yang digunakan untuk menggabungkan secara spesifik baris attribute yang akan ditampilkan sedangkan DISTINCT digunakan untuk menghapus baris yang duplikat. Persamaan antara kedua query ini adalah dapat digunakan untuk membatasi jumlah baris yang ditampilkan.

**Data Awal**



**DISTINCT**

A picture containing text

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

**GROUP BY**

Text

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

**Referensi:**

* Ramez Elmasri , Shamkant Navathe (2015). Fundamentals of Database Systems 7th Edition. New York: Pearson. Halaman 218-219 dan Halaman 194

2. Di dalam locking protocol terdapat 2 fase yaitu:

- Growing Phase (1st Phase)

Pada fase ini, locks dapat didapatkan tetapi tidak ada yang dapat dilepaskan.

- Shrinking Phase (2nd Phase)

Pada fase ini, locks dapat dilepaskan tetapi tidak ada yang dapat didapatkan

**Referensi :**

* Ramez Elmasri , Shamkant Navathe (2015). Fundamentals of Database Systems 7th Edition. New York: Pearson. Halaman 786-787.

3. Embedded SQL pada query SQL merupakan query database yang di embedded ke dalam host bahasa programming yang menggunakan special prefix. Kekurangan dari embedded sql ini adalah terdapat impedence mismatch yaitu masalah uang muncul karena perbedaan database model dan perbedaan model bahasa pemrograman. Contoh dari impedence mismatch adalah perbedaan data types dari bahasa pemrograman dengan attribute data type yang ada di data model. Kelebihan dari embedded sql ini adalah memudahkan developer untuk menakses database melalui berbagai macam bahasa pemrograman lainnya.

**Referensi :**

* Ramez Elmasri , Shamkant Navathe (2015). Fundamentals of Database Systems 7th Edition. New York: Pearson. Halaman 309, 311-312.

4. Konsep transaksi ACID pada kasus ini yaitu:

Atomicity: Transaksi merupakan unit atomic yang harus dilakukan sepenuhnya atau tidak sama sekali. Sehingga pada kasus ini, transaksi 50$ harus terjadi secara sepenuhnya dari read(A) hingga write(B) dan jika terdapat error maka transaksi harus di rollback.

Consistency: Transaksi yang dilakukan harus dieksekusi dari awal hingga akhir secara konsisten tanpa interferensi. Maka pada kasus ini, dengan read(A) untuk mengecek saldo rekening agar transfer 50$ kepada rekening B hingga write(B) dapat terjadi sepenuhnya. Jika saldo rekening A tidak cukup maka transaksi akan di rollback.

Isolation: Transaksi yang dilakukan harus tereksekusi secara terisolasi dari transaksi lainnya. Pada kasus ini maka transaksi harus berjalan dari step 1 – 6 secara terisolasi tanpa gangguan dari transaksi lainnya.

Durability: Perubahan database saat transaksi di commit harus tetap tersimpan di database. Hasil dari transaksi ini yaitu A <- A – 50 dan B <- B + 50 harus terjadi secara permanen dan tersimpan didalam database.

**Referensi :**

* Ramez Elmasri , Shamkant Navathe (2015). Fundamentals of Database Systems 7th Edition. New York: Pearson. Halaman 757-758.

**Kasus**

Nomor 1

SELECT nama\_pembeli, jumlah\_pembelian

FROM pembelian

WHERE jumlah\_pembelian > 5

Nomor 2

SELECT id\_barang, nama\_barang, stok\_barang

FROM barang

ORDER BY stok\_barang ASC

LIMIT 1

Nomor 3

SELECT a.id\_barang, a.nama\_barang, b.jumlah\_pembelian

FROM barang a JOIN pembelian b ON a.id\_barang = b.id\_barang

ORDER BY b.tanggal\_beli ASC

Nomor 4

SELECT a.nama\_barang, a.tanggal\_terima, a.stok\_barang

FROM barang a JOIN pembelian b ON a.id\_barang = b.id\_barang

WHERE b.jumlah\_pembelian > 3 AND b.tanggal\_beli BETWEEN '01-06-2022' AND '30-12-2022'

Nomor 5

SELECT tanggal\_beli, nama\_pembeli, jumlah\_pembelian

FROM pembelian

WHERE jumlah\_pembelian = 2

UNION

SELECT tanggal\_beli, nama\_pembeli, jumlah\_pembelian

FROM pembelian

WHERE nama\_pembeli LIKE '%al%'

Nomor 6

SELECT a.id\_barang, a.nama\_barang, b.tanggal\_beli, b.nama\_pembeli, b.jumlah\_pembelian

FROM barang a JOIN pembelian b ON a.id\_barang = b.id\_barang

ORDER BY b.jumlah\_pembelian DESC

LIMIT 1