

LAPORAN AKHIR PRAKTIKUM

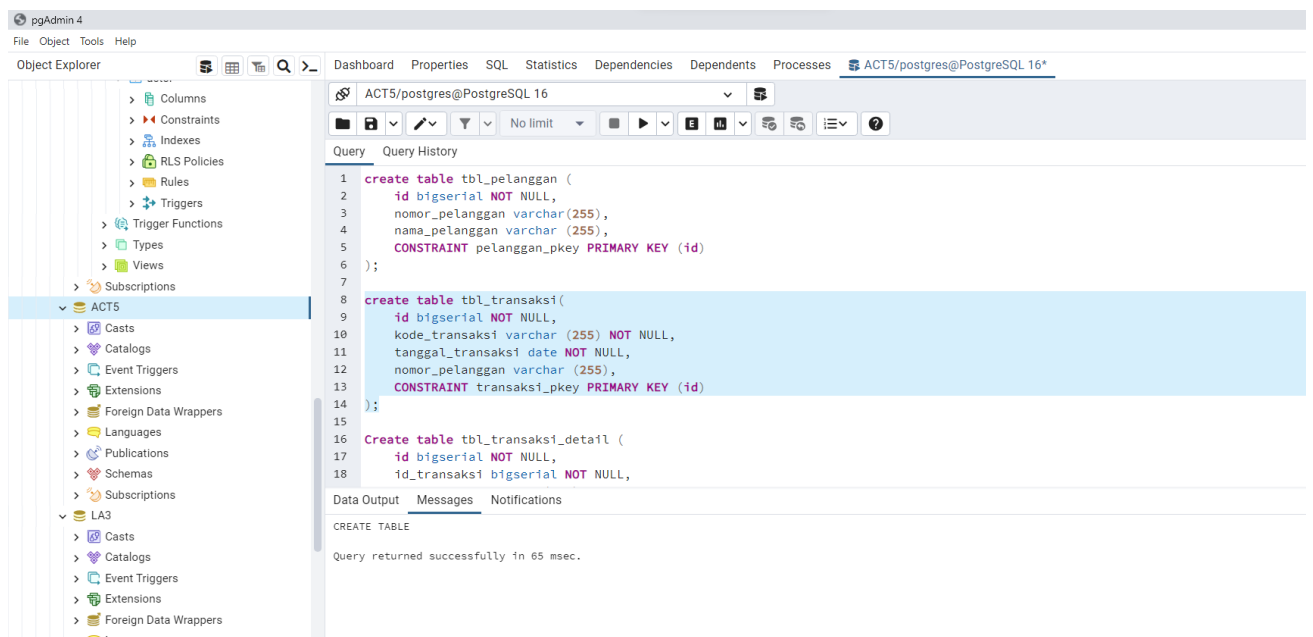
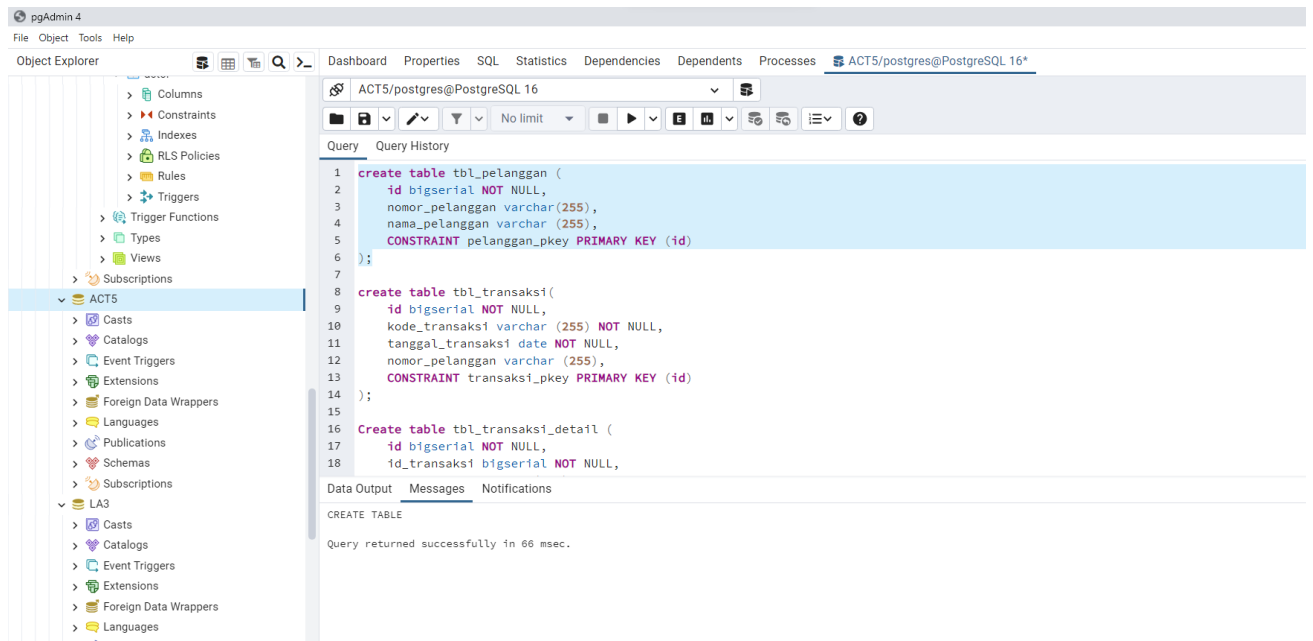
Mata Praktikum : SBD1
Kelas : 3IA19
Praktikum : 5
Tanggal : 11-13-2023
Materi : EQUI JOIN
NPM : 50421859
Nama : Muhamad Ariel D.P
Ketua Asisten :
Nama Asisten : Annisya Amanda
Paraf Asisten :
Jumlah Lembar : 9



**LABORATORIUM INFORMATIKA
UNIVERSITAS GUNADARMA
2023**

LISTING

1. Jelaskan Activity yang sudah kalian buat?



Pertama Kita buat File data base lagi dengan nama ACT 5, Selanjutnya run dua kode blok yang diatas kode diatas memiliki Kode di atas adalah perintah SQL untuk membuat dua tabel dalam database, yaitu `tbl_pelanggan` dan `tbl_transaksi`.

1. Tabel `tbl_pelanggan` memiliki empat kolom:

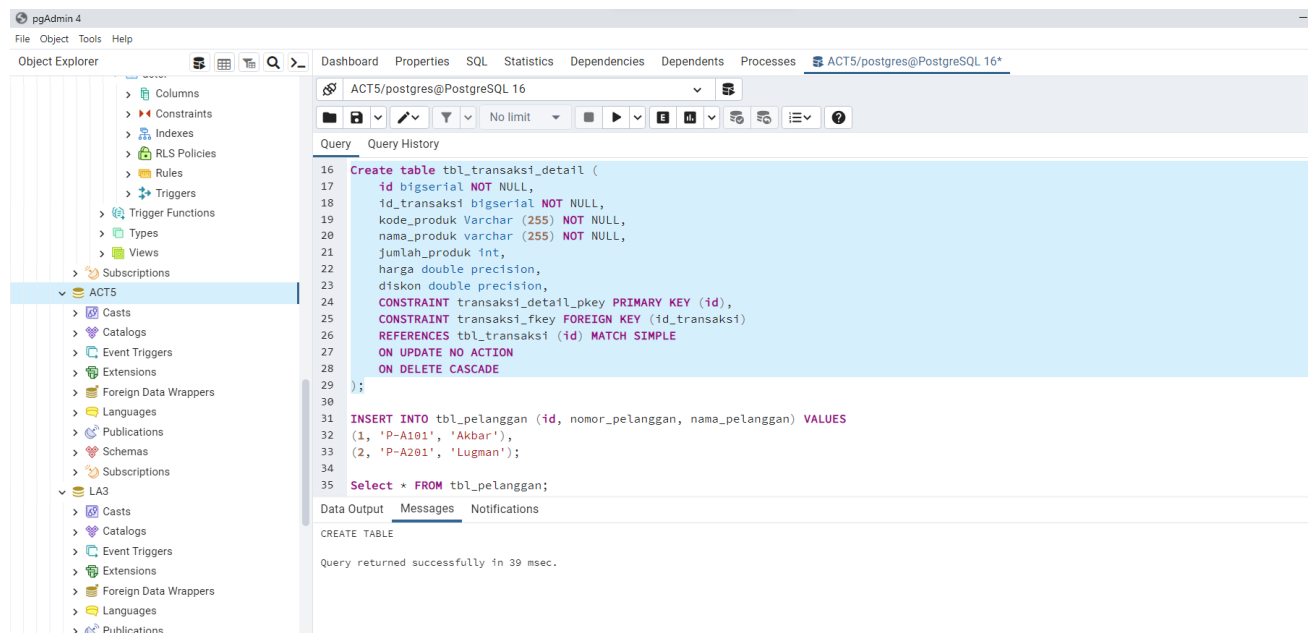
- `id`: Tipe data `bigserial` dan tidak boleh kosong (`NOT NULL`). Ini juga merupakan kunci utama (`PRIMARY KEY`) dari tabel.
- `nomor_pelanggan`: Tipe data `varchar(255)`.
- `nama_pelanggan`: Tipe data `varchar(255)`.

2. Tabel `tbl_transaksi` memiliki empat kolom:

- `id`: Tipe data `bigserial` dan tidak boleh kosong (`NOT NULL`). Ini juga merupakan kunci utama (`PRIMARY KEY`) dari tabel.
- `kode_transaksi`: Tipe data `varchar(255)` dan tidak boleh kosong (`NOT NULL`).
- `tanggal_transaksi`: Tipe data `date` dan tidak boleh kosong (`NOT NULL`).
- `nomor_pelanggan`: Tipe data `varchar(255)`.

Dalam konteks ini, `bigserial` adalah tipe data auto-increment yang biasa digunakan untuk kunci utama. `varchar(255)` adalah tipe data karakter variabel, di mana `255` adalah panjang maksimum string. `date` adalah tipe data tanggal. `NOT NULL` berarti kolom harus memiliki nilai (tidak boleh kosong), dan `PRIMARY KEY` menunjukkan bahwa kolom tersebut adalah kunci utama dari tabel. Kunci utama adalah kolom unik yang digunakan untuk mengidentifikasi setiap baris dalam tabel.

Perlu dicatat bahwa tabel `tbl_transaksi` tampaknya memiliki kolom `nomor_pelanggan` yang mungkin merujuk ke kolom `nomor_pelanggan` di tabel `tbl_pelanggan`. Ini menunjukkan adanya hubungan antara dua tabel tersebut, namun perintah SQL di atas tidak secara eksplisit mendefinisikan hubungan ini. Biasanya, hubungan semacam itu didefinisikan dengan `FOREIGN KEY`. Namun, dalam kasus ini, tampaknya pengguna database memilih untuk mengelola hubungan ini secara manual.



pgAdmin 4

File Object Tools Help

Object Explorer

- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- Trigger Functions
- Types
- Views
- Subscriptions
- ACT5
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
- LA3
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications

Dashboard Properties SQL Statistics Dependencies Dependents Processes ACT5/postgres@PostgreSQL 16*

ACT5/postgres@PostgreSQL 16

Query Query History

```
26 REFERENCES tbl_transaksi (id) MATCH SIMPLE
27 ON UPDATE NO ACTION
28 ON DELETE CASCADE
29 );
30
31 INSERT INTO tbl_pelanggan (id, nomor_pelanggan, nama_pelanggan) VALUES
32 (1, 'P-A101', 'Akbar'),
33 (2, 'P-A201', 'Lugman');
34
35 Select * FROM tbl_pelanggan;
36
37 INSERT INTO tbl_transaksi (id, kode_transaksi, tanggal_transaksi, nomor_pelanggan)
38 VALUES (1, 'TRX-101', '2022-01-29', 'P-A101');
39 INSERT INTO tbl_transaksi (id, kode_transaksi, tanggal_transaksi, nomor_pelanggan)
40 VALUES (2, 'TRX-102', '2022-02-15', 'P-A201');
41 INSERT INTO tbl_transaksi (id, kode_transaksi, tanggal_transaksi, nomor_pelanggan)
42 VALUES (3, 'TRX-103', '2022-02-23', '');
43
44 Select * from tbl_transaksi;
45
```

Data Output Messages Notifications

INSERT 0 2

Query returned successfully in 40 msec.

pgAdmin 4

File Object Tools Help

Object Explorer

- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- Trigger Functions
- Types
- Views
- Subscriptions
- ACT5
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
- LA3
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications

Dashboard Properties SQL Statistics Dependencies Dependents Processes ACT5/postgres@PostgreSQL 16*

ACT5/postgres@PostgreSQL 16

Query Query History

```
26 REFERENCES tbl_transaksi (id) MATCH SIMPLE
27 ON UPDATE NO ACTION
28 ON DELETE CASCADE
29 );
30
31 INSERT INTO tbl_pelanggan (id, nomor_pelanggan, nama_pelanggan) VALUES
32 (1, 'P-A101', 'Akbar'),
33 (2, 'P-A201', 'Lugman');
34
35 Select * FROM tbl_pelanggan;
36
37 INSERT INTO tbl_transaksi (id, kode_transaksi, tanggal_transaksi, nomor_pelanggan)
38 VALUES (1, 'TRX-101', '2022-01-29', 'P-A101');
39 INSERT INTO tbl_transaksi (id, kode_transaksi, tanggal_transaksi, nomor_pelanggan)
40 VALUES (2, 'TRX-102', '2022-02-15', 'P-A201');
41 INSERT INTO tbl_transaksi (id, kode_transaksi, tanggal_transaksi, nomor_pelanggan)
42 VALUES (3, 'TRX-103', '2022-02-23', '');
43
44 Select * from tbl_transaksi;
45
```

Data Output Messages Notifications

	id [PK] bigint	nomor_pelanggan character varying (255)	nama_pelanggan character varying (255)
1	1	P-A101	Akbar
2	2	P-A201	Lugman

pgAdmin 4

File Object Tools Help

Object Explorer

- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- Trigger Functions
- Types
- Views
- Subscriptions
- ACT5
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
- LA3
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languaoces

Dashboard Properties SQL Statistics Dependencies Dependents Processes ACT5/postgres@PostgreSQL 16*

Query Query History

```
31 INSERT INTO tbl_pelanggan (id, nomor_pelanggan, nama_pelanggan) VALUES
32 (1, 'P-A101', 'Akbar'),
33 (2, 'P-A201', 'Lugman');
34
35 Select * FROM tbl_pelanggan;
36
37 INSERT INTO tbl_transaksi (id, kode_transaksi, tanggal_transaksi, nomor_pelanggan)
38 VALUES (1, 'TRX-101', '2022-01-29', 'P-A101');
39 INSERT INTO tbl_transaksi (id, kode_transaksi, tanggal_transaksi, nomor_pelanggan)
40 VALUES (2, 'TRX-102', '2022-02-15', 'P-A201');
41 INSERT INTO tbl_transaksi (id, kode_transaksi, tanggal_transaksi, nomor_pelanggan)
42 VALUES (3, 'TRX-103', '2022-02-23', '');
43
44 Select * from tbl_transaksi;
45
46 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
47 VALUES (7, 3, 'AH-645', 'Keyboard RedDragon', 2, 350000, 16000);
48 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
49 VALUES (8, 3, 'AH-884', 'LCD Samsung 21 inch', 5, 2108800, 0);
50 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
```

Data Output Messages Notifications

INSERT 0 1

Query returned successfully in 87 msec.

pgAdmin 4

File Object Tools Help

Object Explorer

- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- Trigger Functions
- Types
- Views
- Subscriptions
- ACT5
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
- LA3
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions

Dashboard Properties SQL Statistics Dependencies Dependents Processes ACT5/postgres@PostgreSQL 16*

Query Query History

```
36
37 INSERT INTO tbl_transaksi (id, kode_transaksi, tanggal_transaksi, nomor_pelanggan)
38 VALUES (1, 'TRX-101', '2022-01-29', 'P-A101');
39 INSERT INTO tbl_transaksi (id, kode_transaksi, tanggal_transaksi, nomor_pelanggan)
40 VALUES (2, 'TRX-102', '2022-02-15', 'P-A201');
41 INSERT INTO tbl_transaksi (id, kode_transaksi, tanggal_transaksi, nomor_pelanggan)
42 VALUES (3, 'TRX-103', '2022-02-23', '');
43
44 Select * from tbl_transaksi;
45
46 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
47 VALUES (7, 3, 'AH-645', 'Keyboard RedDragon', 2, 350000, 16000);
48 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
49 VALUES (8, 3, 'AH-884', 'LCD Samsung 21 inch', 5, 2108800, 0);
50 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
51 VALUES (9, 3, 'CK-849', 'DDR4 16GB Corsair', 2, 867000, 0);
52 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
53 VALUES (4, 2, 'CK-834', 'SSD 512 ADATA', 1, 456000, 9000);
54 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
55 VALUES (5, 2, 'AH-501', 'Mouse Logitech', 2, 299000, 20000);
```

Data Output Messages Notifications

	id [PK] bigint	kode_transaksi character varying (255)	tanggal_transaksi date	nomor_pelanggan character varying (255)
1	1	TRX-101	2022-01-29	P-A101
2	2	TRX-102	2022-02-15	P-A201
3	3	TRX-103	2022-02-23	

pgAdmin 4

File Object Tools Help

Object Explorer

- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- Trigger Functions
- Types
- Views
- Subscriptions
- ACTS
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
- LA3
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications

Dashboard Properties SQL Statistics Dependencies Dependents Processes ACT5/postgres@PostgreSQL 16*

Query Query History

```
41 INSERT INTO tbl_transaksi (id, kode_transaksi, tanggal_transaksi, nomor_pelanggan)
42 VALUES (3, 'TRX-103', '2022-02-23', '');
43
44 Select * from tbl_transaksi;
45
46 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
47 VALUES (7, 3, 'AH-645', 'Keyboard RedDragon', 2, 358000, 16000);
48 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
49 VALUES (8, 3, 'AH-884', 'LCD Samsung 21 inch', 5, 2108800, 0);
50 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
51 VALUES (9, 3, 'CK-849', 'DDR4 18GB Corsair', 2, 867000, 0);
52 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
53 VALUES (4, 2, 'CK-834', 'SSD 512 ADATA', 1, 456000, 9000);
54 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
55 VALUES (5, 2, 'AH-501', 'Mouse Logitech', 2, 299000, 20000);
56 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
57 VALUES (6, 2, 'AK-882', 'Baterai ABC', 4, 20000, 0);
58
59 Select * from tbl_transaksi_detail;
60
```

Data Output Messages Notifications

INSERT 0 1

Query returned successfully in 52 msec.

pgAdmin 4

File Object Tools Help

Object Explorer

- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- Trigger Functions
- Types
- Views
- Subscriptions
- ACTS
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
- LA3
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications

Dashboard Properties SQL Statistics Dependencies Dependents Processes ACT5/postgres@PostgreSQL 16*

Query Query History

```
50 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
51 VALUES (9, 3, 'CK-849', 'DDR4 18GB Corsair', 2, 867000, 0);
52 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
53 VALUES (4, 2, 'CK-834', 'SSD 512 ADATA', 1, 456000, 9000);
54 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
55 VALUES (5, 2, 'AH-501', 'Mouse Logitech', 2, 299000, 20000);
56 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
57 VALUES (6, 2, 'AK-882', 'Baterai ABC', 4, 20000, 0);
58
59 Select * from tbl_transaksi_detail;
60
61 Select tbl_pelanggan.id, tbl_pelanggan.nomor_pelanggan, tbl_pelanggan.nama_pelanggan,
62 tbl_transaksi.id, tbl_transaksi.kode_transaksi, tbl_transaksi.tanggal_transaksi, tbl_transaksi.nomor_pelanggan
63 FROM tbl_pelanggan, tbl_transaksi
64 WHERE tbl_pelanggan.id = tbl_transaksi.id
65 AND kode_transaksi = 'TRX-102';
66
67 SELECT c.nomor_pelanggan, c.nama_pelanggan, a.kode_transaksi
68 FROM tbl_pelanggan c, tbl_transaksi a
69 WHERE c.id = a.id
```

Data Output Messages Notifications

id	id_transaksi	kode_produk	nama_produk	jumlah_produk	harga	diskon
(PK) bigint	bigint	character varying (255)	character varying (255)	integer	double precision	double precision
1	7	AH-645	Keyboard RedDragon	2	358000	16000
2	8	AH-884	LCD Samsung 21 inch	5	2108800	0
3	9	CK-849	DDR4 18GB Corsair	2	867000	0
4	4	CK-834	SSD 512 ADATA	1	456000	9000
5	5	AH-501	Mouse Logitech	2	299000	20000
6	6	AK-882	Baterai ABC	4	20000	0

pgAdmin 4

File Object Tools Help

Object Explorer

- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- Trigger Functions
- Types
- Views
- Subscriptions
- ACTS
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
- LA3
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications

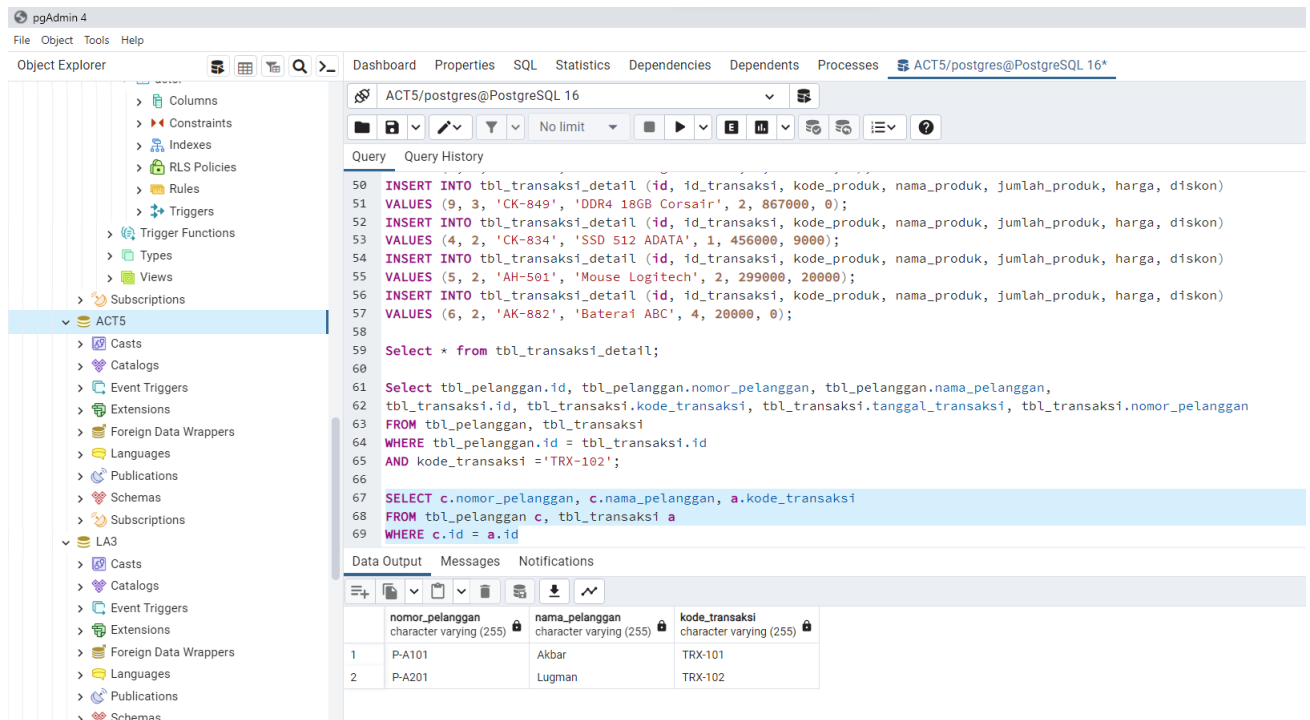
Dashboard Properties SQL Statistics Dependencies Dependents Processes ACT5/postgres@PostgreSQL 16*

Query Query History

```
50 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
51 VALUES (9, 3, 'CK-849', 'DDR4 18GB Corsair', 2, 867000, 0);
52 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
53 VALUES (4, 2, 'CK-834', 'SSD 512 ADATA', 1, 456000, 9000);
54 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
55 VALUES (5, 2, 'AH-501', 'Mouse Logitech', 2, 299000, 20000);
56 INSERT INTO tbl_transaksi_detail (id, id_transaksi, kode_produk, nama_produk, jumlah_produk, harga, diskon)
57 VALUES (6, 2, 'AK-882', 'Baterai ABC', 4, 20000, 0);
58
59 Select * from tbl_transaksi_detail;
60
61 Select tbl_pelanggan.id, tbl_pelanggan.nomor_pelanggan, tbl_pelanggan.nama_pelanggan,
62 tbl_transaksi.id, tbl_transaksi.kode_transaksi, tbl_transaksi.tanggal_transaksi, tbl_transaksi.nomor_pelanggan
63 FROM tbl_pelanggan, tbl_transaksi
64 WHERE tbl_pelanggan.id = tbl_transaksi.id
65 AND kode_transaksi = 'TRX-102';
66
67 SELECT c.nomor_pelanggan, c.nama_pelanggan, a.kode_transaksi
68 FROM tbl_pelanggan c, tbl_transaksi a
69 WHERE c.id = a.id
```

Data Output Messages Notifications

id	nomor_pelanggan	nama_pelanggan	id	kode_transaksi	tanggal_transaksi	nomor_pelanggan	
bigint	character varying (255)	character varying (255)	bigint	character varying (255)	date	character varying (255)	
1	2	P-A201	Lugman	2	TRX-102	2022-02-15	P-A201



Selanjutnya Kita Masukkan Kembali syntax seperti diatas dan kemudian run kode2 diatas memiliki arti Kode tersebut adalah serangkaian perintah SQL yang digunakan untuk membuat tabel, memasukkan data ke dalam tabel, dan mengambil data dari tabel. Berikut adalah penjelasan lengkapnya:

1. Membuat Tabel:

- `CREATE TABLE tbl_transaksi_detail`: Membuat tabel baru bernama 'tbl_transaksi_detail' dengan kolom-kolom seperti 'id', 'id_transaksi', 'kode_produk', 'nama_produk', 'jumlah_produk', 'harga', dan 'diskon'. Kolom 'id' adalah kunci utama ('PRIMARY KEY') dan 'id_transaksi' adalah kunci asing ('FOREIGN KEY') yang merujuk ke 'id' pada tabel 'tbl_transaksi'.

2. Memasukkan Data:

- `INSERT INTO tbl_pelanggan`: Memasukkan data ke dalam tabel 'tbl_pelanggan'. Misalnya, memasukkan data pelanggan dengan 'id' 1, 'nomor_pelanggan' 'P-A101', dan 'nama_pelanggan' 'Akbar'.

- `INSERT INTO tbl_transaksi`: Memasukkan data ke dalam tabel 'tbl_transaksi'. Misalnya, memasukkan data transaksi dengan 'id' 1, 'kode_transaksi' 'TRX-101', 'tanggal_transaksi' '2022-01-29', dan 'nomor_pelanggan' 'P-A101'.

- `INSERT INTO tbl_transaksi_detail`: Memasukkan data ke dalam tabel 'tbl_transaksi_detail'. Misalnya, memasukkan data detail transaksi dengan 'id' 7, 'id_transaksi' 3, 'kode_produk' 'AH-645', 'nama_produk' 'Keyboard RedDragon', 'jumlah_produk' 2, 'harga' 358000, dan 'diskon' 16000.

3. Mengambil Data:

- `SELECT * FROM tbl_pelanggan`: Mengambil semua data dari tabel `tbl_pelanggan`.
- `SELECT * FROM tbl_transaksi`: Mengambil semua data dari tabel `tbl_transaksi`.
- `SELECT * FROM tbl_transaksi_detail`: Mengambil semua data dari tabel `tbl_transaksi_detail`.
- `SELECT tbl_pelanggan.id, tbl_pelanggan.nomor_pelanggan, tbl_pelanggan.nama_pelanggan, tbl_transaksi.id, tbl_transaksi.kode_transaksi, tbl_transaksi.tanggal_transaksi, tbl_transaksi.nomor_pelanggan FROM tbl_pelanggan, tbl_transaksi WHERE tbl_pelanggan.id = tbl_transaksi.id AND kode_transaksi = 'TRX-102': Mengambil data tertentu dari tabel `tbl_pelanggan` dan `tbl_transaksi` berdasarkan kondisi tertentu.
- `SELECT c.nomor_pelanggan, c.nama_pelanggan, a.kode_transaksi FROM tbl_pelanggan c, tbl_transaksi a WHERE c.id = a.id`: Mengambil data tertentu dari tabel `tbl_pelanggan` dan `tbl_transaksi` berdasarkan kondisi tertentu.

Secara umum, kode ini digunakan untuk mengelola data pelanggan, transaksi, dan detail transaksi dalam database. Ini adalah bagian penting dari sistem manajemen database relasional.

2. Jelaskan Secara Singkat Pengertian Outer dan fungsinya

Jawab:

Outer Join adalah operasi SQL yang mencakup baris yang berada di bagian dalam tabel (bagian dalam) dengan baris yang tidak cocok diantara dua tabel². Outer Join pada dasarnya beda dengan Inner Join terkait bagaimana cara menangani kondisi tabel yang tidak cocok antara kedua tabel². Dari operasi Outer Join ini ternyata masih dibagi lagi menjadi tiga jenis yaitu:

1. **Left Outer Join**: Mengembalikan semua baris dari tabel sebelah kiri dan mencocokkan data yang sesuai diantara kedua tabel². Baris – baris yang tidak memiliki pasangan pada tabel yang ada disebelah kiri kata “JOIN” akan tetap ditampilkan¹.
2. **Right Outer Join**: Mengembalikan semua baris dari tabel sebelah kanan dengan mencocokkan data yang pas antara kedua tabel². Data pada tabel sebelah kiri hanya ditampilkan jika data terkait pada tabel tersebut muncul pada tabel sebelah kanan¹.
3. **Full Outer Join**: Menggabungkan hasil antara kedua teknik Outer Join yaitu Left Outer Join dengan Right Outer Join². Mengutip W3school, terdapat beberapa perbedaan antara Inner dan Outer Join².

Secara umum, Outer Join digunakan ketika Anda ingin mempertahankan setiap baris dari satu tabel atau lebih, tetapi juga ingin menarik data dari tabel lain yang cocok. Jika tidak ada kecocokan, hasilnya akan NULL..