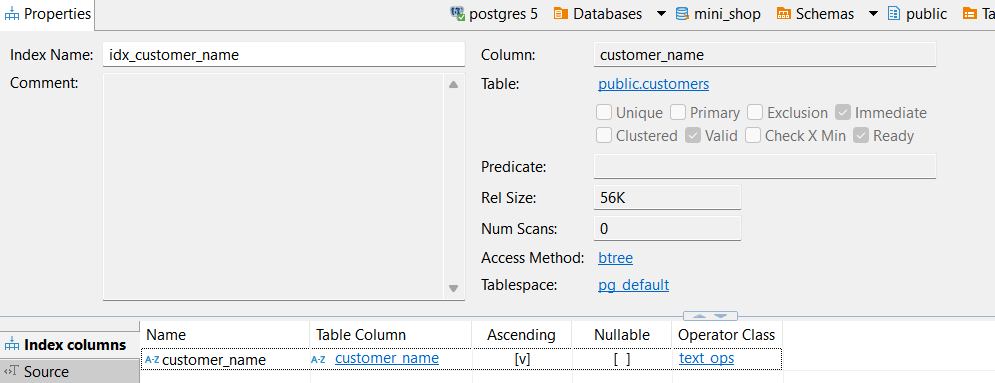
**Part 2**

**Task 3 : Create Index an index on the customer\_name column in the customers**

1. **Syntax**:

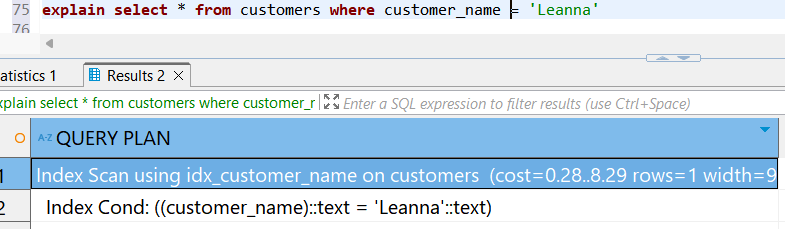
**CREATE** **INDEX** idx\_customer\_name **ON** customers(customer\_name);

1. **Description**
2. Setelah execute syntax di atas maka kita sudah berhasil membuat index customer name untuk meningkatkan kecepatan pencarian berdasarkan customer\_name



1. Gunakan query **EXPLAIN** untuk melihat plan dari pertintah query

**EXPLAIN** **SELECT** \* **FROM** customers **WHERE** customer\_name = 'Leanna'



Index Scan menandakan pencarian sudah berdasarkan idx\_customer\_name

**Task 4: Partition the orders table based on the order\_date column to enhance query performance involving date ranges**

Untuk mmeudahkan kompare data setelahnya, disini kita bisa crate new table untuk orders table sehingga nanti kita masih punya data backup untuk table order. Kemudian disini data yang dipakai untuk partition adalah data tahun 2021, karena data 2023 tidak lengkap dan juga supaya tidak terlalu banyak partisi.

1. **Syntax: Create Partition**

Membuat **table baru** yakni **orders\_new** dengan partisi order\_date. Untuk order\_date di buat dengan data type DATE supaya bisa di partisi, dan juga di jadikan PK:

-- Create new table for orders table

CREATE TABLE public.orders\_new (

    order\_id INT NOT NULL,

    customer\_id INT,

    payment INT,

    order\_date DATE,

    delivery\_date DATE,

    payment\_status VARCHAR(20) DEFAULT 'Pending',

    CONSTRAINT pk\_order\_new PRIMARY KEY (order\_id, order\_date), -- Menambahkan order\_date ke dalam primary key

    CONSTRAINT fk\_customer\_new FOREIGN KEY (customer\_id) REFERENCES public.customers(customer\_id)

) PARTITION BY RANGE (order\_date);

Setelah itu, membuat **partisi per kuartal** dengan rincian sebagai berikut:

--Create table partition

-- Partisi untuk Kuartal 1 (1 Jan - 31 Mar 2021)

CREATE TABLE orders\_new\_2021\_q1 PARTITION OF orders\_new

    FOR VALUES FROM ('2021-01-01') TO ('2021-04-01');

-- Partisi untuk Kuartal 2 (1 Apr - 30 Jun 2021)

CREATE TABLE orders\_new\_2021\_q2 PARTITION OF orders\_new

    FOR VALUES FROM ('2021-04-01') TO ('2021-07-01');

-- Partisi untuk Kuartal 3 (1 Jul - 30 Sep 2021)

CREATE TABLE orders\_new\_2021\_q3 PARTITION OF orders\_new

    FOR VALUES FROM ('2021-07-01') TO ('2021-10-01');

-- Partisi untuk Kuartal 4 (1 Okt - 31 Des 2021)

CREATE TABLE orders\_new\_2021\_q4 PARTITION OF orders\_new

    FOR VALUES FROM ('2021-10-01') TO ('2022-01-01');

1. **Syntax: Insert Data**

Setelah partition table sudah tersedia, kita bisa eksekusi query insert data dari table lama ke table baru.

-- Insert data to new table

INSERT INTO orders\_new (order\_id, customer\_id, payment, order\_date, delivery\_date, payment\_status)

SELECT order\_id, customer\_id, payment, order\_date::DATE, delivery\_date::DATE, payment\_status

FROM orders

WHERE order\_date >= '2021-01-01' AND order\_date < '2022-01-01';

1. **Syntax: Select from partition table**

Setelah data sudah dimasukkan ke table partition, kita bisa select per kuartal.

Misalkan kita mau select data orders dari order\_date 2021-01-01 sampai dnegan order data tanggal 2021-03-31 maka kita bisa select data dari table orders\_new\_2021\_q1.

**SELECT** \* **FROM** public.orders\_new\_2021\_q1;

**SELECT** \* **FROM** public.orders\_new\_2021\_q2;

**SELECT** \* **FROM** public.orders\_new\_2021\_q3;

**SELECT** \* **FROM** public.orders\_new\_2021\_q4;

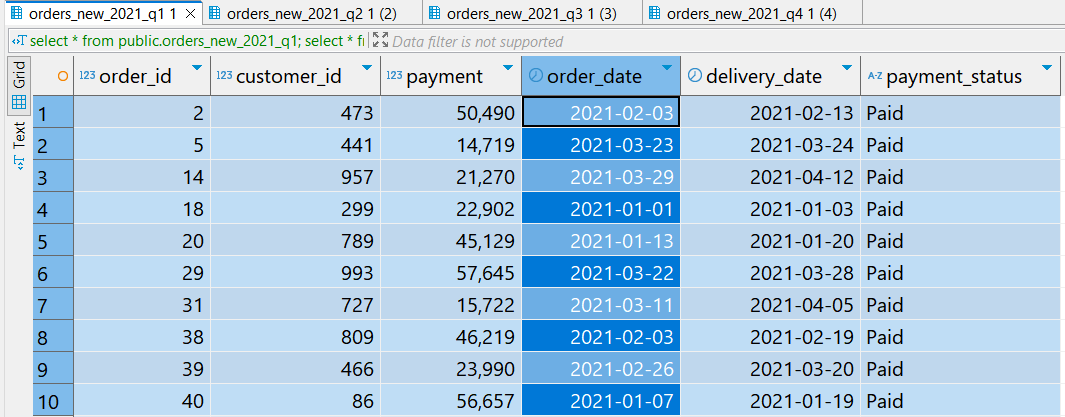


Image 1 Kuartal 1

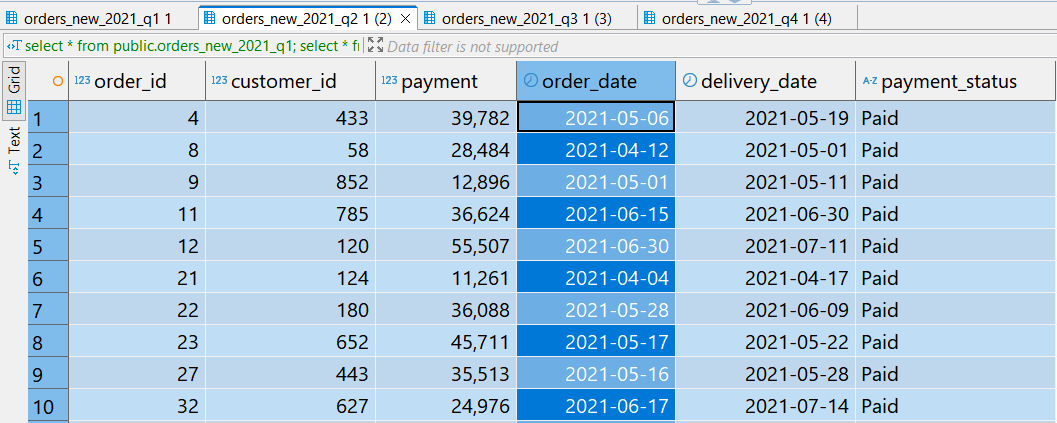


Image 2 Kuartal 2

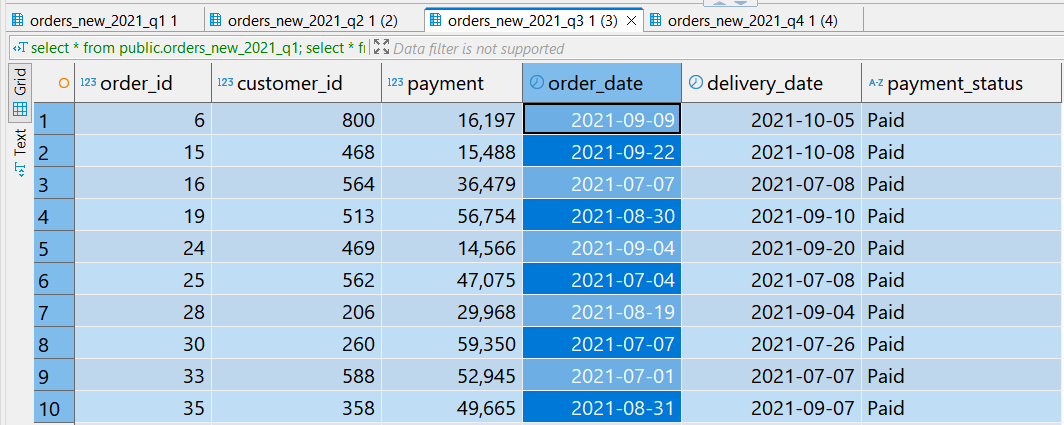


Image 3 Kuartal 3

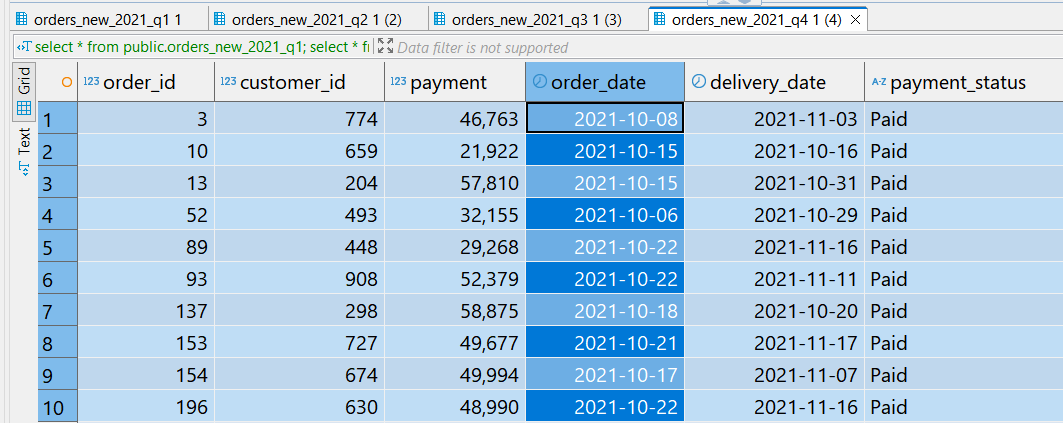


Image 4 Kuartal 4

**Task 5: Searches often use filters based on product type.Create a partition to enhanced the search process. Create a partition table and insert data from the products table.**

1. **Syntax: Create Partition**

Membuat **table baru** yakni **products\_new** dengan partisi **product\_type**:

**CREATE** **TABLE** products\_new (

product\_id **INT** **NOT** **NULL**,

product\_type **VARCHAR**(100) **NULL**,

product\_name **VARCHAR**(100) **NULL**,

"size" **VARCHAR**(100) **NULL**,

colour **VARCHAR**(100) **NULL**,

price **INT** **NULL**,

quantity **INT** **NULL**,

description **VARCHAR**(100) **NULL**,

store\_id **INT** **NULL**,

**CONSTRAINT** pk\_product\_new **PRIMARY** **KEY** (product\_id, product\_type),

**CONSTRAINT** fk\_store\_new **FOREIGN** **KEY** (store\_id) **REFERENCES** public.stores(store\_id)

) **PARTITION** **BY** LIST (product\_type);

Setelah itu membuat table partisi per product\_type:

-- Partisi untuk jenis produk Trousers

**CREATE** **TABLE** products\_new\_trousers **PARTITION** **OF** products\_new

**FOR** **VALUES** **IN** ('Trousers');

-- Partisi untuk jenis produk Shirt

**CREATE** **TABLE** products\_new\_shirt **PARTITION** **OF** products\_new

**FOR** **VALUES** **IN** ('Shirt');

-- Partisi untuk jenis produk Jacket

**CREATE** **TABLE** products\_new\_jacket **PARTITION** **OF** products\_new

**FOR** **VALUES** **IN** ('Jacket');

1. **Syntax: Insert Data**

Masukkan data lama ke table baru:

**INSERT** **INTO** products\_new (product\_id, product\_type, product\_name, "size", colour, price, quantity, description, store\_id)

**SELECT** product\_id, product\_type, product\_name, "size", colour, price, quantity, description, store\_id

**FROM** products;

1. **Syntax: Select from partition table**

Setelah table partisi selesai di buat. Table ini bisa diguankan ketika kita ingin select data product dengan type yang kita inginkan.

Misalkan jika ingin filter data product jacket. Maka kita bisa select dari table products\_new\_jacket. Data ini bisa bikombinasi dnegan filter lainnya, seperti price dll.

**SELECT** \* **FROM** public.products\_new\_jacket;

**SELECT** \* **FROM** public.products\_new\_shirt;

**SELECT** \* **FROM** public.products\_new\_trousers;

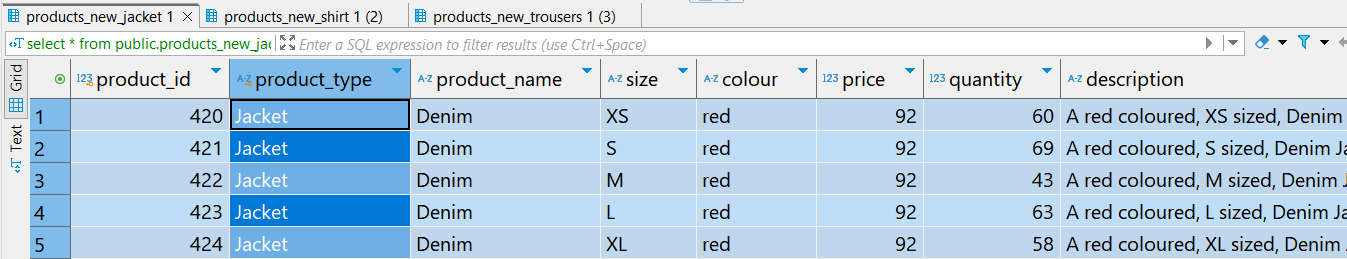


Image 5 Product type Jacket

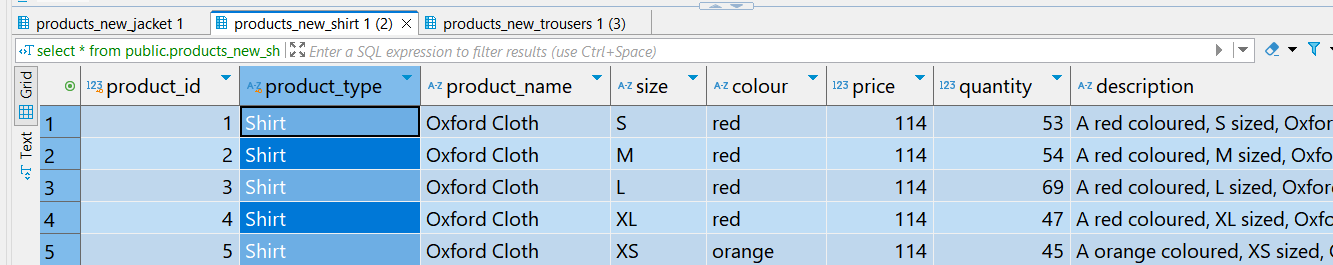


Image 6 Product type Shirt

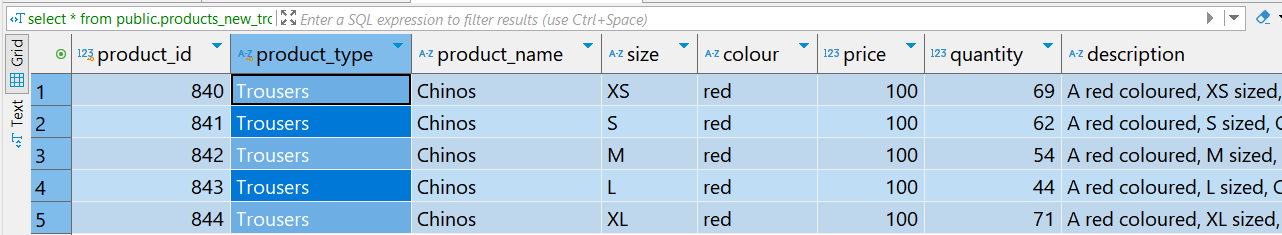


Image 7 Product type Trousers