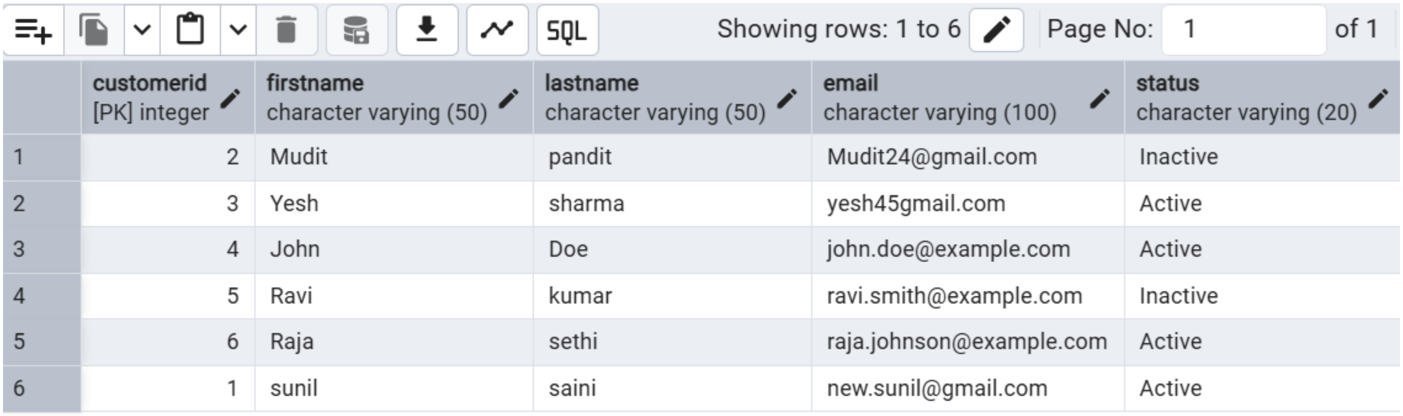
```sql
CREATE TABLE Customers (
    CustomerId INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100),
    Status VARCHAR(20)
);

INSERT INTO Customers (CustomerId, FirstName, LastName, Email, Status) VALUES
(1, 'sunil ', 'saini', 'sunil123@gmail.com', 'Active'),
(2, 'Mudit', 'pandit', 'Mudit24@gmail.com', 'Inactive'),
(3, 'Yesh', 'sharma', 'yesh45gmail.com', 'Active'),
(4, 'John', 'Doe', 'john.doe@example.com', 'Active'),
(5, 'Ravi', 'kumar', 'ravi.smith@example.com', 'Inactive'),
(6, 'Raja', 'sethi', 'raja.johnson@example.com', 'Active');
SELECT GetCustomerFullName(1);



-- 2. Create Products table
CREATE TABLE Products (
    ProductId INT PRIMARY KEY,
    ProductName VARCHAR(100),
    Price NUMERIC(10, 2)
);

-- Sample Inserts (optional)
INSERT INTO Products VALUES
(101, 'Laptop', 50000.00),
(102, 'Mouse', 500.00),
(103, 'Keyboard', 1000.00),
(104, 'Monitor', 7000.00),
(105, 'Printer', 8500.00),
(106, 'USB Cable', 250.00);
```

```
-- 3. Create Orders table
CREATE TABLE Orders (
    OrderId INT PRIMARY KEY,
    CustomerId INT REFERENCES Customers(CustomerId),
    OrderDate DATE
);

INSERT INTO Orders VALUES
(201, 1, '2024-12-01'),
(202, 2, '2024-12-05'),
(203, 1, '2025-01-10'),
(204, 3, '2025-02-20'),
(205, 5, '2025-03-15'),
(206, 2, '2025-04-01');

-- 4. Create OrderDetails table
CREATE TABLE OrderDetails (
    OrderDetailId INT PRIMARY KEY,
    OrderId INT REFERENCES Orders(OrderId),
    ProductId INT REFERENCES Products(ProductId),
    Quantity INT
);

INSERT INTO OrderDetails VALUES
(301, 201, 101, 1),
(302, 201, 102, 2),
(303, 202, 104, 1),
(304, 203, 103, 1),
(305, 204, 105, 1),
(306, 205, 106, 3);
```

-- 1. Create an index on Customers.Email

CREATE INDEX idx_customers_email ON Customers(Email);

SELECT * FROM Customers;

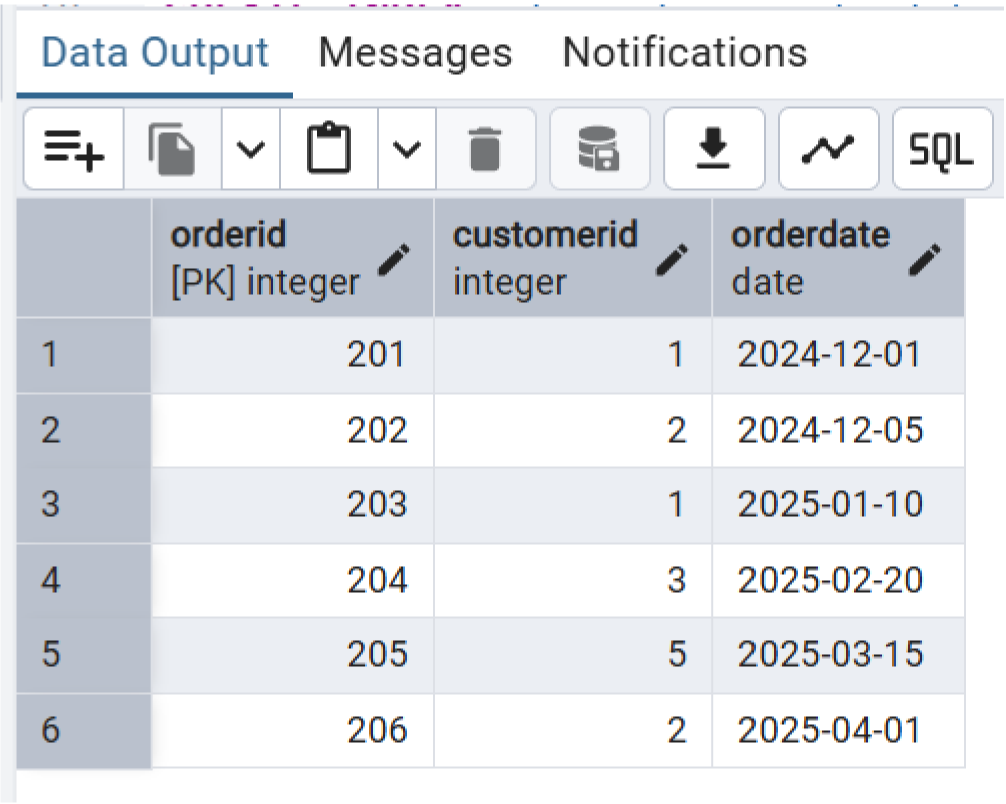


	customerid [PK] integer	firstname character varying (50)	lastname character varying (50)	email character varying (100)	status character varying (20)
1	2	Mudit	pandit	Mudit24@gmail.com	Inactive
2	3	Yesh	sharma	yesh45gmail.com	Active
3	4	John	Doe	john.doe@example.com	Active
4	5	Ravi	kumar	ravi.smith@example.com	Inactive
5	6	Raja	sethi	raja.johnson@example.com	Active
6	1	sunil	saini	new.sunil@gmail.com	Active

-- 2. Create an index on Orders.OrderDate to speed up date-based queries

CREATE INDEX idx_orders_orderdate ON Orders(OrderDate);

SELECT * FROM Orders;

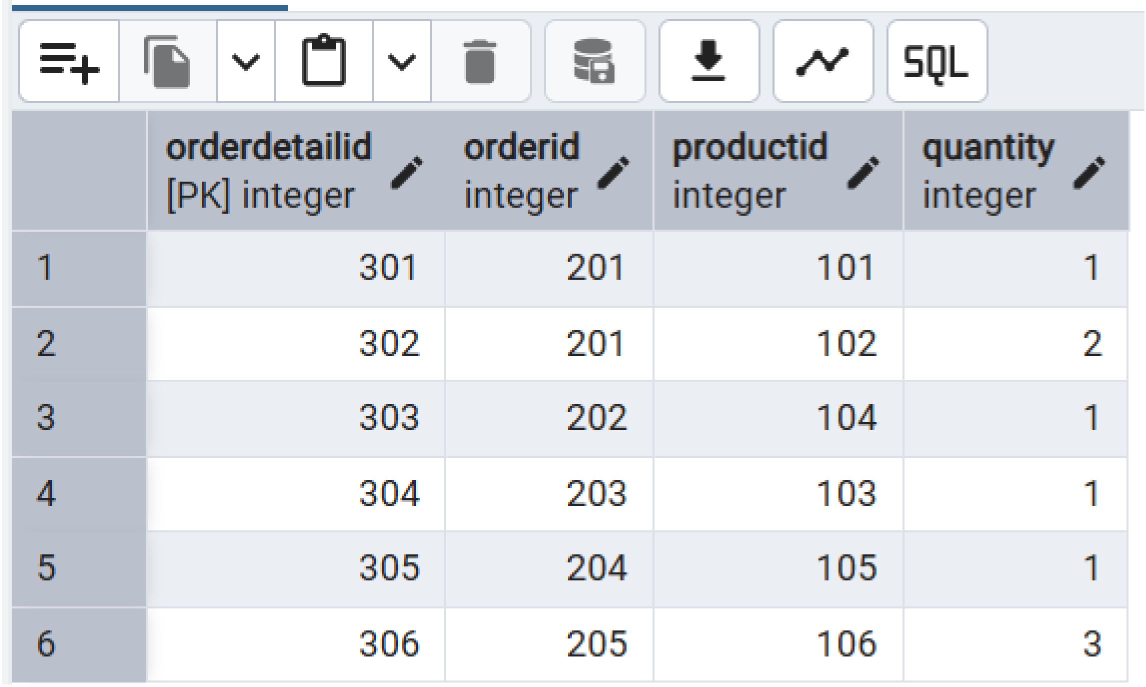


	orderid [PK] integer	customerid integer	orderdate date
1	201	1	2024-12-01
2	202	2	2024-12-05
3	203	1	2025-01-10
4	204	3	2025-02-20
5	205	5	2025-03-15
6	206	2	2025-04-01

-- 3. Create a composite index on OrderDetails(ProductId, Quantity)

CREATE INDEX idx_orderdetails_product_quantity ON OrderDetails(ProductId, Quantity);

SELECT * FROM OrderDetails;

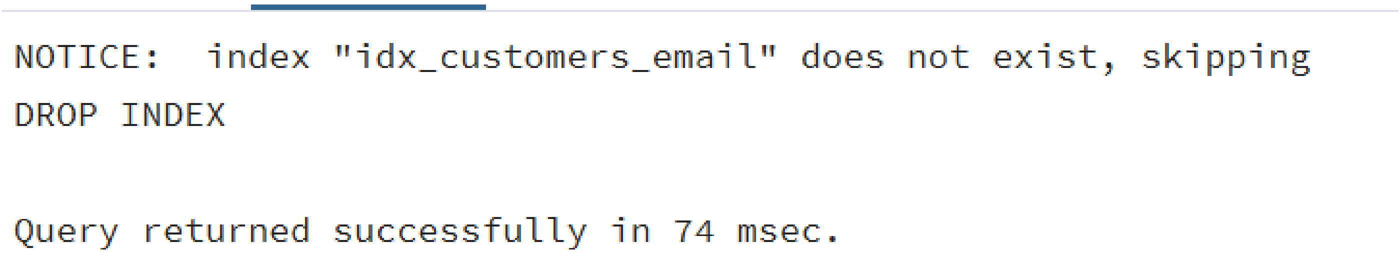


	orderdetailid [PK] integer	orderid integer	productid integer	quantity integer
1	301	201	101	1
2	302	201	102	2
3	303	202	104	1
4	304	203	103	1
5	305	204	105	1
6	306	205	106	3

-- 4. Drop an index if it is no longer needed Dropping index on Email

CREATE INDEX idx_customers_email ON Customers(Email);

DROP INDEX IF EXISTS idx_customers_email;

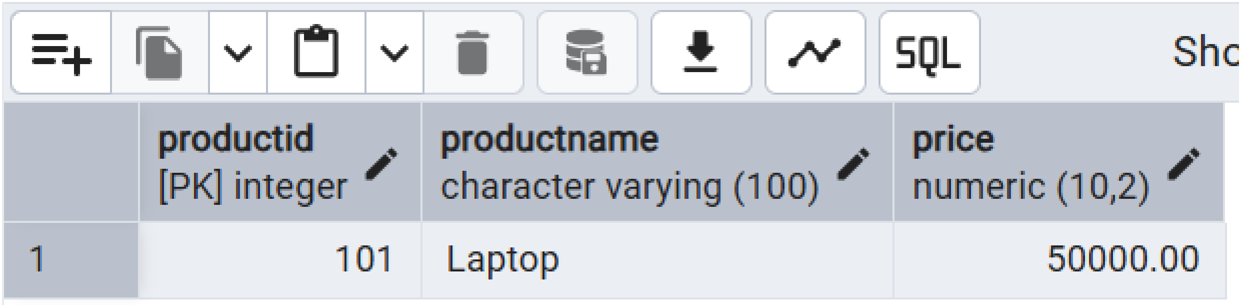


```
NOTICE:  index "idx_customers_email" does not exist, skipping
DROP INDEX

Query returned successfully in 74 msec.
```

-- 5. Check performance difference by running SELECT with and without index on ProductName

-- Before Index

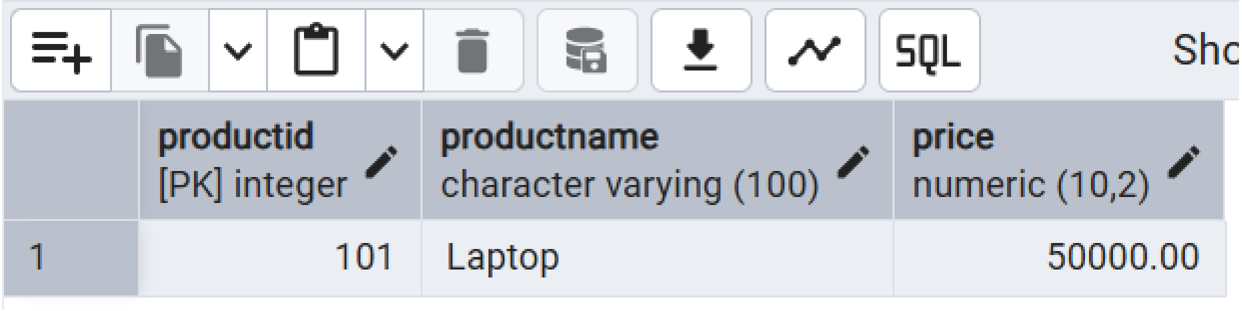
SELECT * FROM Products WHERE ProductName = 'Laptop';



	productid [PK] integer	productname character varying (100)	price numeric (10,2)
1	101	Laptop	50000.00

```
-- Create index on ProductName

CREATE INDEX idx_products_productname ON Products(ProductName);
```

```
-- After Index

SELECT * FROM Products WHERE ProductName = 'Laptop';
```



	productid [PK] integer	productname character varying (100)	price numeric (10,2)
1	101	Laptop	50000.00

6. Create a unique index on Email column to prevent duplicates

```
-- Note: Email is already NOT UNIQUE, so check for duplicates before applying

CREATE UNIQUE INDEX idx_customers_email_unique ON Customers(Email);
```

```
-- 7. Create a filtered index on Customers where Status = 'Active'

-- PostgreSQL uses partial indexes for filtering

CREATE INDEX idx_customers_active ON Customers(Email)

WHERE Status = 'Active';
```