

Managing PostgreSQL on Windows/Ubuntu

Outline

1. The pgAdmin
2. Practices – Create a new database
3. Access to database objects: Login/Group Roles

- localhost
- Port: 5432
- Account: postgres
- Password: admin

1. The pgAdmin (pgadmin III / pgadmin4)

The pgAdmin

- Any function you need to perform on your PostgreSQL system you can do from within the pgAdmin III/ pgAdmin IV graphical interface
- Location: `~bin\pgadmin3.exe`
- Default:
 - localhost
 - port: 5432
- Add new connect: File → Add server
- Connect server: right click → Connect

Browser window showing pgAdmin 4 interface. The address bar displays `127.0.0.1:64092/browser/`. The pgAdmin logo and title "pgAdmin Management Tools for PostgreSQL" are visible. The "Create" menu is open, showing options: "Server Group...", "Server...", "Refresh...", "Remove Server Group", and "Properties...". The "Server..." option is selected.

pgAdmin Management Tools for PostgreSQL

Feature rich | Maximises PostgreSQL | Open Source

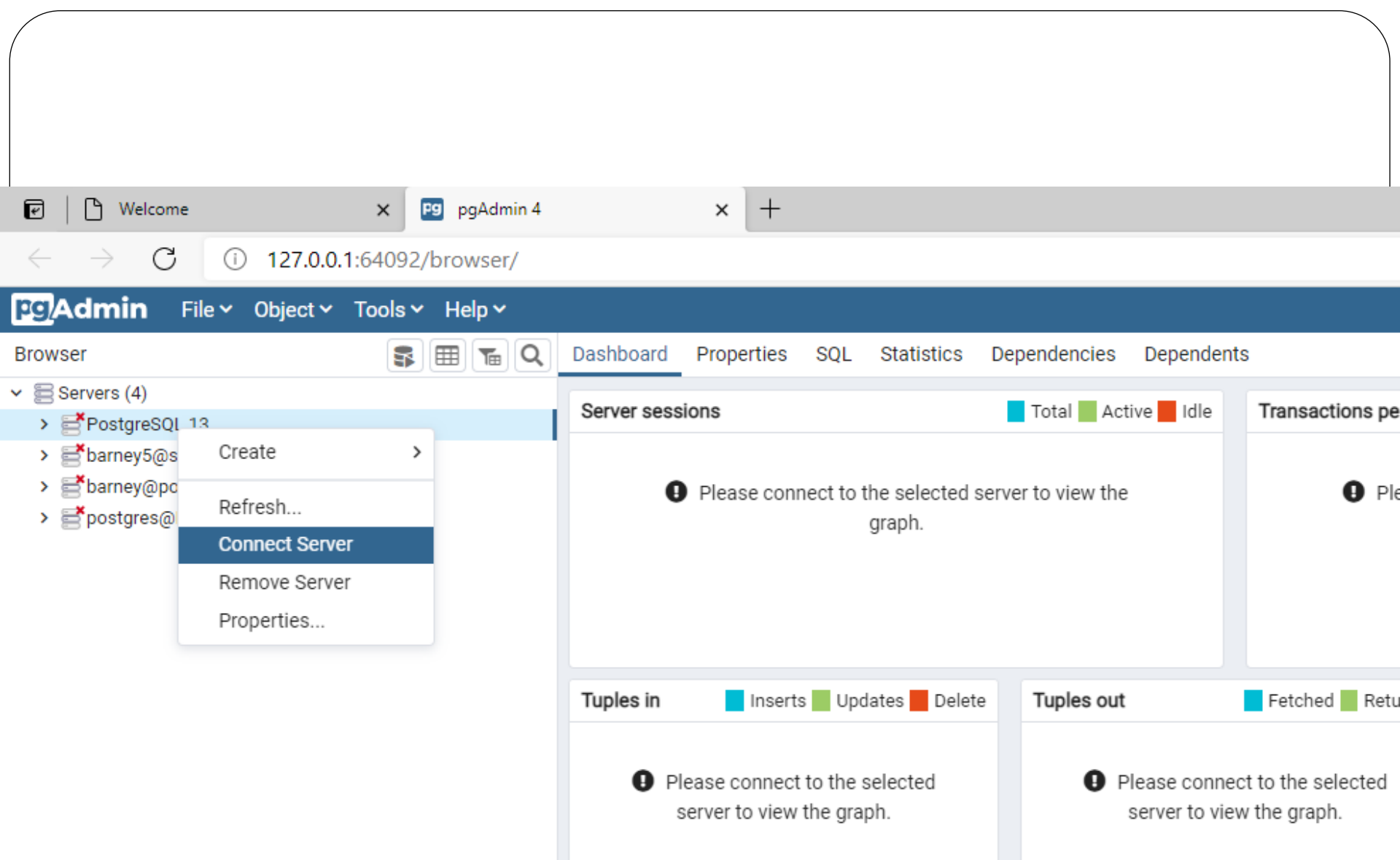
pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.

Quick Links

- Add New Server
- Configure pgAdmin

Getting Started

- PostgreSQL Documentation
- pgAdmin Website
- Planet PostgreSQL
- Community Support



Welcome pgAdmin 4

127.0.0.1:64092/browser/

pgAdmin File Object Tools Help

Browser

- Servers (4)
 - PostgreSQL 13
 - Databases (1)
 - postgres
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Schemas (1)
 - public
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Procedures
 - Sequences
 - Tables
 - Trigger Functions
 - Types
 - Views

Database sessions

1

0

Tuples in Inserts Updates

1

0

Server activity

Sessions Locks Prepared Transactions

			PID	User	Application
✖	■	▶	4496	postgres	pgAdmin

2. Practices – Create a new database

Practices – Create a new database

- Create a database ***test***
 - Customer
 - Product
 - Order
- Create two Group Roles (**later**)
 - *Salesman* Group Role: write permission on the Customer and Order, only read permission on the Product
 - *Accountant* Group Role: write permission on the Product and Order, read permission on the Customer
- Create two Login Roles (**later**)
 - salesman - Barney
 - accountant - Fred

Creating a New Database

The screenshot displays the pgAdmin 4 web interface in a browser window. The address bar shows the URL `127.0.0.1:64092/browser/`. The main interface is divided into a left sidebar, a central workspace, and a right sidebar.

Left Sidebar: Shows a tree view of the database system. Under "Servers (4)", "PostgreSQL 13" is expanded, showing "Databases (2)", "Login/Group Roles", and "Tablespaces". Below this, there are roles: "barney5@server", "barney@postgres10", and "postgres@PostgreSQL 10".

Central Workspace: A "Create - Database" dialog box is open, showing the "General" tab. The fields are as follows:

- Database:** `test`
- Owner:** `postgres` (selected from a dropdown)
- Comment:** (empty text area)

At the bottom of the dialog are buttons for "Cancel", "Reset", and "Save".

Right Sidebar: Contains monitoring dashboards. The top dashboard shows "Transactions", "Commits", and "Rollbacks" with a line graph. The bottom dashboard shows "Block I/O" with "Reads" and "Hits" and another line graph. Below the graphs is a search bar and a table with columns: "State", "Wait event", and "Blocking".

Bottom Taskbar: Shows the Windows taskbar with a search bar "Type here to search" and several application icons. The system clock in the bottom right corner shows "4:03 PM 3/10/2021".

Template

- CREATE DATABASE actually works by copying an existing database
- Default, it copies the standard system database *template1*
- There is a second standard system database named *template0*
 - the same data as the initial contents of *template1*
 - *never be changed*

Creating a New Schema

The screenshot shows the pgAdmin 4 web interface in a browser window. The address bar displays `127.0.0.1:64092/browser/`. The pgAdmin 4 logo and menu (File, Object, Tools, Help) are at the top. On the left, the 'Browser' pane shows a tree view of the database structure: Servers (4) > PostgreSQL 13 > Databases (2) > test > Schemas (1) > public. The 'Schemas (1)' folder is selected. A 'Create - Schema' dialog box is open in the foreground, with the 'General' tab active. The 'Name' field contains 'store', the 'Owner' is set to 'postgres', and the 'Comment' field is empty. At the bottom of the dialog are buttons for 'i', '?', 'Cancel', 'Reset', and 'Save'. The background interface is partially obscured by the dialog box.

Schemas (1/3)

- The **most important objects** within the database
- A database **contains one or more schemas**, which contain **database object** (table, data type, domain, function, trigger) **definitions**
- While users can only access objects **within one database at a time**, they can access **all of the schemas within that database**, *if it has permissions*
- Unlike databases, schemas are not rigidly separated

Schemas (2/3)

- Sometimes **related applications** can share the **same database**, but use **different schemas** to hold their separate data. This makes it easier for users to find tables related to the applications within the database. This is especially true if tables have the same names.
- Table names must be **unique within a schema**, but can be duplicated between schemas.
- Tables are referenced in SQL statements using the format: ***schemaName.tableName***
- Default schemas: ***public***

Schemas (3/3)

Schema Object	Description
Aggregates	Defines functions that produce results based on processing input values from multiple records in a table (such as a sum or average)
Conversions	Defines conversions between character set encodings
Domains	User-defined data types
Functions	User-defined functions
Trigger Functions	User-defined table triggers
Procedures	User-defined functions that manipulate data but do not return a value
Operators	User-defined operators used to compare data
Operator Classes	Defines how a data type can be used within an index
Sequences	Defines a sequenced number generator
Tables	User-created data repositories
Types	User-defined data types used in the database
Views	User-created queries combining data from multiple tables

Creating the Tables

The screenshot displays the pgAdmin 4 interface for creating a new table. The 'Create - Table' dialog box is open, showing the 'General' tab. The 'Name' field is set to 'customer', the 'Owner' is 'postgres', and the 'Schema' is 'store'. The 'Tablespace' is set to 'Select an item...'. The 'Partitioned table?' checkbox is set to 'No'. The 'Comment' field is empty. The 'Browser' pane on the left shows the database structure, with 'Tables' highlighted under the 'store' schema. The bottom status bar shows the connection details: 18356, postgres, pgAdmin 4 - DB:test, ::1, 2021-03-10 16:03:13 +07.

pgAdmin File ▾ Object ▾ Tools ▾ Help ▾

Browser

- > Event Triggers
- > Extensions
- > Foreign Data Wrappers
- > Languages
- ▼ Schemas (2)
 - > public
 - ▼ store
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Procedures
 - > Sequences
 - Tables**
 - > Trigger Functions
 - > Types
 - > Views
- > Login/Group Roles
- > Tablespaces
- > barney5@server

Create - Table

General Columns Advanced Constraints Partitions Parameters Security SQL

Name: customer

Owner: postgres

Schema: store

Tablespace: Select an item...

Partitioned table?: No

Comment:

Buttons: [i] [?] [Cancel] [Reset] [Save]

Status Bar: 18356 postgres pgAdmin 4 - DB:test ::1 2021-03-10 16:03:13 +07

Customer Table Columns

Column	Data Type	Description
CustomerID	char—six characters	Unique identifier for each customer
LastName	varchar	Last name of customer
FirstName	varchar	First name of customer
Address	varchar	Street address of customer
City	varchar	City of customer
State	char—two characters	State of customer
Zip	char—five characters	Postal ZIP code of customer
Phone	varchar	Phone number of customer

- > Event Triggers
- > Extensions
- > Foreign Data Wrappers
- > Languages
- ▼ Schemas (2)
 - > public
 - ▼ store
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Procedures
 - > Sequences
 - Tables
 - > Trigger Functions
 - > Types
 - > Views
- > Login/Group Roles
- > Tablespaces
- > barney5@server

Create - Table

General Columns Advanced Constraints Partitions Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
	customerid	character	6		Yes	Yes

Cancel Reset Save

pgAdmin 4

127.0.0.1:64092/browser/

pgAdmin File Object Tools Help

Browser

- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Schemas (2)
 - public
 - store
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Procedures
 - Sequences
 - Tables**
 - Trigger Functions
 - Types
 - Views
- Login/Group Roles
- Tablespaces

Create - Table

General Columns Advanced Constraints Partitions Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

	Name	Columns
<input checked="" type="checkbox"/>	pk_customer	customerid

General Definition

Columns

Include columns

Tablespace

Fill factor

Deferrable?

Deferred?

Common PostgreSQL Data Types

Name	Aliases	Description
<code>bigint</code>	<code>int8</code>	signed eight-byte integer
<code>bigserial</code>	<code>serial8</code>	autoincrementing eight-byte integer
<code>bit [(n)]</code>		fixed-length bit string
<code>bit varying [(n)]</code>	<code>varbit [(n)]</code>	variable-length bit string
<code>boolean</code>	<code>bool</code>	logical Boolean (true/false)
<code>box</code>		rectangular box on a plane
<code>bytea</code>		binary data ("byte array")
<code>character [(n)]</code>	<code>char [(n)]</code>	fixed-length character string
<code>character varying [(n)]</code>	<code>varchar [(n)]</code>	variable-length character string
<code>date</code>		calendar date (year, month, day)
<code>double precision</code>	<code>float8</code>	double precision floating-point number (8 bytes)
<code>inet</code>		IPv4 or IPv6 host address
<code>integer</code>	<code>int</code> , <code>int4</code>	signed four-byte integer
<code>money</code>		currency amount, two-decimal place floating-point number
<code>numeric [(p, s)]</code>	<code>decimal [(p, s)]</code>	exact numeric of selectable precision
<code>real</code>	<code>float4</code>	single precision floating-point number (4 bytes)
<code>smallint</code>	<code>int2</code>	signed two-byte integer
<code>smallserial</code>	<code>serial2</code>	autoincrementing two-byte integer
<code>serial</code>	<code>serial4</code>	autoincrementing four-byte integer
<code>text</code>		variable-length character string
<code>time [(p)] [without time zone]</code>		time of day (no time zone)
<code>time [(p)] with time zone</code>	<code>timetz</code>	time of day, including time zone

The Product Table Columns

Column Name	Data Type	Description
ProductID	char—six characters	Unique primary key identifier that is not NULL
ProductName	varchar	Name of the product
Model	varchar	Product model number
Manufacturer	varchar	Name of the manufacturer
UnitPrice	money	Current price of product
Inventory	int4	Number of units in inventory

The Columns for the Order Table

Column Name	Data Type	Description
OrderID	char—six characters	Unique primary key identifier that is not NULL
CustomerID	char—six characters	The CustomerID from the Customer table (not NULL)
ProductID	char—six characters	The ProductID from the Product table (not NULL)
PurchaseDate	date	Date of purchase
Quantity	int4	The number of items purchased
TotalCost	money	The total cost of the purchase

New Foreign Key window for the Order table

pgAdmin 4 browser window showing the 'Create - Table' dialog for the 'customer' table in the 'store' schema. The 'Constraints' tab is active, showing a new foreign key constraint named 'fk_order_2customer' being created, linking the 'customerid' column to the 'customerid' column of the 'store.customer' table.

pgAdmin 4 | Welcome | pgAdmin 4 | 127.0.0.1:64092/browser/

pgAdmin | File | Object | Tools | Help

Browser

- Schemas (2)
 - public
 - store
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Procedures
 - Sequences
 - Tables (1)**
 - customer**
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - Trigger Functions
 - Types

Create - Table

General | Columns | Advanced | **Constraints** | Partitions | Parameters | Security | SQL

	Name	Columns	Referenced Table
<input checked="" type="checkbox"/>	fk_order_2customer	(customerid) -> (customerid)	store.customer

General | Definition | **Columns** | Action

Columns

Local	Referenced	Referenced Table
customerid	customerid	store.customer

Buttons: i ? Cancel Reset Save

Entering and Viewing Data

The screenshot shows the pgAdmin 4 web interface in a browser window. The address bar displays `127.0.0.1:64092/browser/`. The left sidebar shows a tree view of database objects, with 'Tables (2)' expanded to show 'customer' and 'order'. The main panel displays a SQL query in the 'Query Editor' tab:

```
1 SELECT * FROM store.custom
2 ORDER BY customerid ASC
```

Below the query editor, a table of results is shown:

1	1
2	2
3	1

An error message 'ERRO' is visible in the bottom right corner.

Entering and Viewing Data

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays a tree view of the database structure, with the 'customer' table under the 'store' schema selected. The main panel is divided into two sections: the 'Query Editor' and the 'Data Output' section.

Query Editor: Contains the following SQL query:

```
1 SELECT * FROM store.customer
2 ORDER BY customerid ASC
```

Data Output: Displays the results of the query in a table format:

	customerid
1	1
2	2
3	1

Annotations on the image include:

- A red arrow pointing to the 'Save Data Changes (F6)' button in the top toolbar, with the text 'Click here to save'.
- A red arrow pointing to the 'Data Output' section, with the text 'Edit here'.

Product

	ProductID [PK] character (6)	ProductName character varying (40)	Model character varying (10)	Manufacturer character varying (40)	UnitPrice money	Inventory integer
1	LAP001	Vaio CR31Z	CR	Sony Vaio	\$1.30	5
2	LAP002	HP AZE	HP	[null]	\$1.00	18
3	LAP003	HP 34	HP	HP	\$1,000.00	200

Customer

	CustomerID [PK] character (6)	LastName character varying (40)	FirstName character varying (40)	Address character varying (100)	City character varying (40)	State character (2)	Zip character (10)	Phone character varying (20)
1	BLU001	Blum	Jessica	229 State	Whiting	IN	46300	555-0921
2	BLU003	AAAA	Katie	342 Pine	Hammond	IN	46200	555-9242
3	BLU005	Bbbbbbbb	Rich	123 Main St.	Chicago	IL	60633	555-1234
4	WIL001	Williams	Frank	456 Oak St.	Hammond	IN	46102	[null]

Data Output Explain Messages Query History

Order

	ProductID character (6)	OrderID [PK] character (6)	CustomerID character (6)	PurchaseDate date	Quantity integer	TotalCost money
1	LAP001	ODR001	BLU001	2012-08-21	1	\$1.30
2	LAP002	ODR002	BLU003	2012-02-03	2	\$2.00
3	LAP001	ORD003	WIL001	2012-06-06	1	\$1.30

Violating column constraint

pgAdmin 4

pgAdmin 4 File Object Tools Help

Browser

- store
 - Collations
 - Domains
 - FTS Configurat
 - FTS Dictionarie
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Vi
 - 1.3 Sequences
 - Tables (3)
 - Customer
 - Column
 - Constra
 - Indexes
 - Rules
 - Triggers
 - Order
 - Column
 - Constra
 - Indexe

Dashboard Properties SQL Statistics Dependencies Dependents Edit Data - Postgr...

PostgreSQL 10 - test2 - store.Customer

```
1 SELECT * FROM store."Customer"
2 ORDER BY "CustomerID" ASC
```

Data Output Explain Messages Query History

	CustomerID [PK] character (6)	LastName character varying (20)	FirstName character varying (10)	Address character varying (50)	City character varying (20)
1	BLU001	Blum	Jessica	229 State	Whiting
2	BLU003	AAAA	Katie	342 Pine	Hammond
3	BLU005	Bbbbbbbb	Rich	123 Main St.	Chicago
4	WIL001	Williams	Frank	456 Oak St.	Hammond
5	WIL001	Williams	Frank	[null]	[null]

ERROR: duplicate key value violates unique constraint "pk_Customer" DETAIL: Key ("CustomerID")=(WIL001) already exists. .

Type here to search

11:48 AM 2/26/2019

Violating column constraint

pgAdmin 4

File Object Tools Help

Browser

- store
 - Collations
 - Domains
 - FTS Configurat
 - FTS Dictionarie
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Vi
 - 1.3 Sequences
 - Tables (3)
 - Customer
 - Order
 - Column
 - Constra
 - Indexes
 - Rules
 - Triggers
 - Product
 - Column
 - Constra

Dashboard Properties SQL Statistics Dependencies Dependents Edit Data - Postgr...

PostgreSQL 10 - test2 - store.Order

```
1 SELECT * FROM store."Order"  
2 ORDER BY "OrderID" ASC
```

Data Output Explain Messages Query History

	ProductID character (6)	OrderID [PK] character (6)	CustomerID character (6)	PurchaseDate date	Quantity integer	TotalCost money
1	LAP001	ODR001	BLU001	2012-08-21	1	\$1.30
2	LAP002	ODR002	BLU003	2012-02-03	2	\$2.00
3	LAP001	ORD003	WIL001	2012-06-06	1	\$1.30
4	LAP001	ORD004	WIL002	2016-02-25	1	1.4
5	[null]	[null]	[null]	[null]	[null]	[null]

ERROR: insert or update on table "Order" violates foreign key constraint "fk_order_customer" DETAIL: Key (CustomerID)=(WIL002) is not present in table "Customer".

THE pgADMIN QUERY TOOL

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Browser' pane shows a tree view of database objects, with 'Customer' selected under 'Tables (3)'. The main pane shows a SQL query: `Select * from store."Customer";`. Below the query, the 'Data Output' tab is active, displaying a table with 4 rows and 6 columns: CustomerID, LastName, FirstName, Address, and City. The table data is as follows:

	CustomerID character (6)	LastName character varying (20)	FirstName character varying (10)	Address character varying (50)	City character varying (20)
1	BLU003	AAAA	Katie	342 Pine	Hammond
2	BLU001	Blum	Jessica	229 State	Whiting
3	BLU005	Bbbbbbbb	Rich	123 Main St.	Chicago
4	WIL001	Williams	Frank	456 Oak St.	Hammond

The bottom of the image shows a Windows taskbar with various application icons and a system clock indicating 11:50 AM on 2/26/2019.

