Managing PostgreSQL on Windows/Ubuntu

Outline

1. The pgAdmin
2. Parts of the PostgreSQL system
3. Practices – Create a new application

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

1. The pgAdmin (pgadmin III / pgadmin4)

The pgAmin

- 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

pgAdmin III

File Edit Tools Display Help

Servers (1)

PostgreSQL Database Server 8.1 (localhost:5432)

Properties | Statistics | Depends on | Referenced by

Property	Value
Description	PostgreSQL Database Server 8.1
Hostname	localhost
Port	5432
Service	pgsql-8.1
Maintenance database	postgres
Username	postgres
Store password?	No
Connected?	No
Running?	Yes

Retrieving server properties... Done.

0.00 secs

pgAdmin III

File Edit Tools Display Help

- Servers (1)
  - PostgreSQL Database Server 8.1 (localhost:5432)
    - Databases (1)
      - postgres
        - Casts (0)
        - Languages (0)
        - Schemas (1)
          - public
            - Aggregates (0)
            - Conversions (0)
            - Domains (0)
            - Functions (0)
            - Trigger Functions (0)
            - Procedures (0)
            - Operators (0)
            - Operator Classes (0)
            - Sequences (0)
            - Tables (0)
            - Types (0)
            - Views (0)
        - Replication (0)
    - Tablespaces (2)
    - Group Roles (0)
    - Login Roles (1)

Properties | Statistics | Depends on | Referenced by

Property	Value
Name	public
OID	2200
Owner	postgres
ACL	{postgres=UC/postgres,=UC/postgres}
System schema?	No
Comment	Standard public schema

```
-- Schema: "public"

-- DROP SCHEMA public;

CREATE SCHEMA public
  AUTHORIZATION postgres;
GRANT ALL ON SCHEMA public TO postgres;
GRANT ALL ON SCHEMA public TO public;
COMMENT ON SCHEMA public IS 'Standard public schema';
```

Restoring previous environment... Done. 0.35 secs

2. Practices – Create a new application

Practices – Create a new application

- 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

pgAdmin III

File Edit Tools Display Help

Servers (1)
- PostgreSQL Database Server 8.1
  - Databases (1)
    - postgres
      - Casts (0)
      - Languages (0)
      - Schemas (1)
        - public
          - Aggregates (
          - Conversions (
          - Domains (0)
          - Functions (0)
          - Trigger Functi
          - Procedures (0
          - Operators (0)
          - Operator Clas
          - Sequences (0
          - Tables (0)
          - Types (0)
          - Views (0)
      - Replication (0)
  - Tablespaces (2)
  - Group Roles (0)
  - Login Roles (1)

New Database

Properties | Variables | Privileges | SQL

Name	test
OID	
Owner	postgres
Encoding	UTF8
Template	template1
Tablespace	pg_default
Comment	

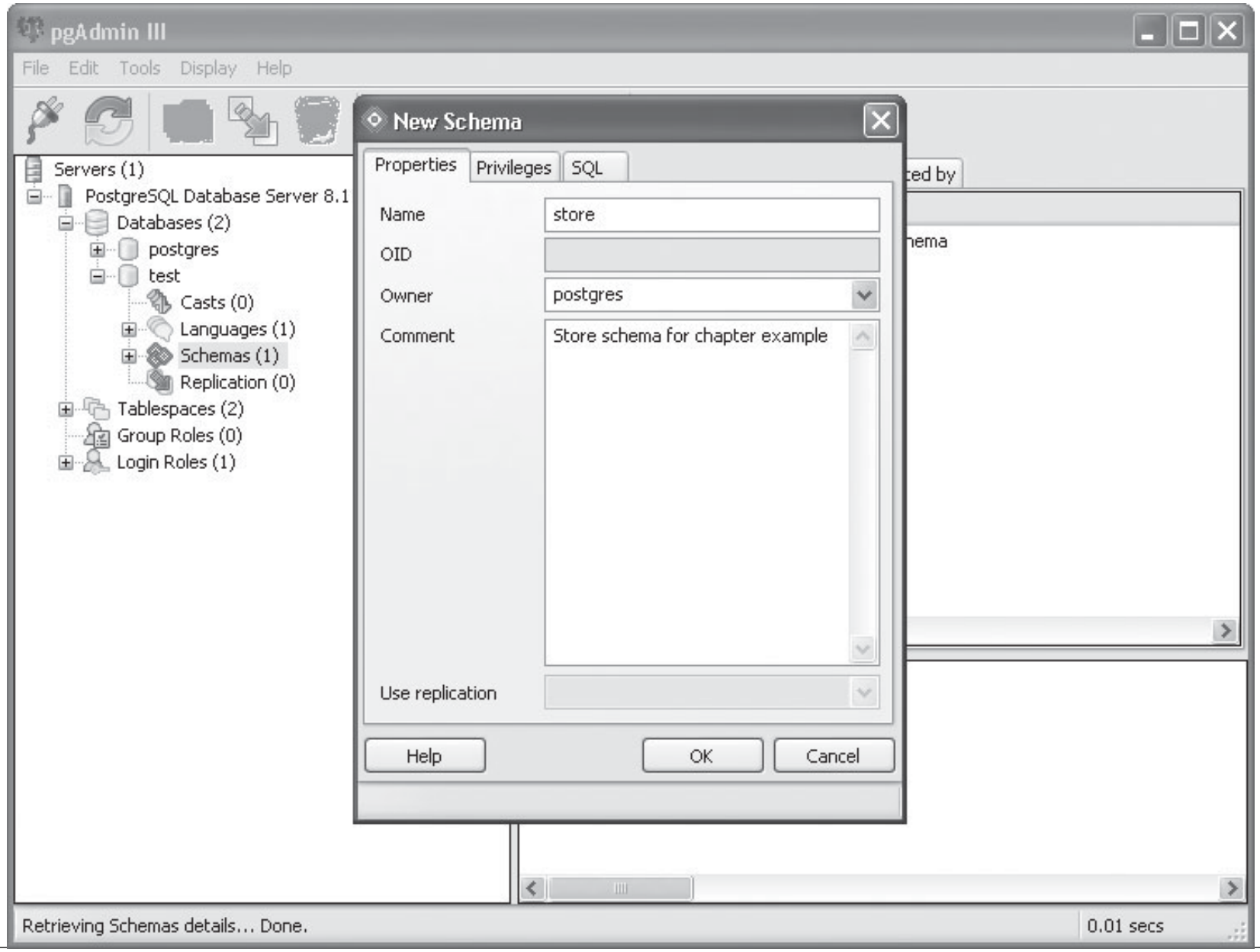
Help | OK | Cancel

Retrieving Databases details... Done. 0.00 secs

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



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

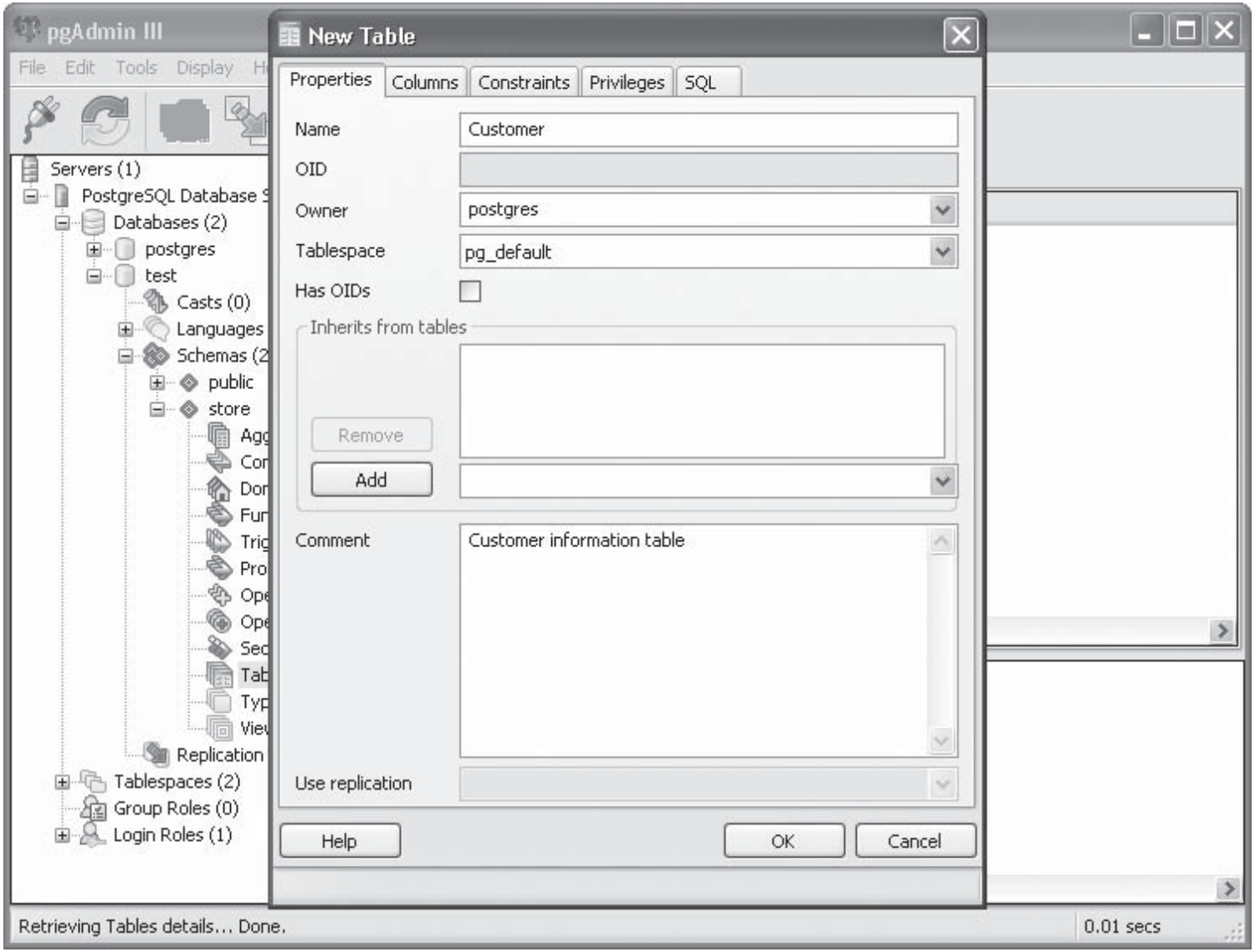
Schemas (2/3)

- 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*

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



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

pgAdmin III

File Edit Tools Display

Servers (1)
PostgreSQL Database S
Databases (2)
postgres
test
Casts (0)
Languages
Schemas (2
public
store
Agg
Con
Dom
Fun
Trig
Pro
Ope
Ope
Sec
Tab
Typ
Vie
Replication
Tablespaces (2)
Group Roles (0)
Login Roles (1)

Table store.Customer

Properties | Columns | Constraints | Vacuum settings | Privileges | SQL

Colu
Custo
LastN
FirstN
Addre
City
State
Zip
Phone

New Column

Properties

Name: CustomerID

Data type: char

Length: 6 Precision:

Default value:

Sequence: <new sequence>

Not NULL: ☑

Statistics:

Comment: Unique customer identifier

Use replication:

Help | OK | Cancel

Help | OK | Cancel

Retrieving Table details... Done.

0.00 secs

Common PostgreSQL Data Types

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

https://www.postgresql.org/docs/11/datatype.html

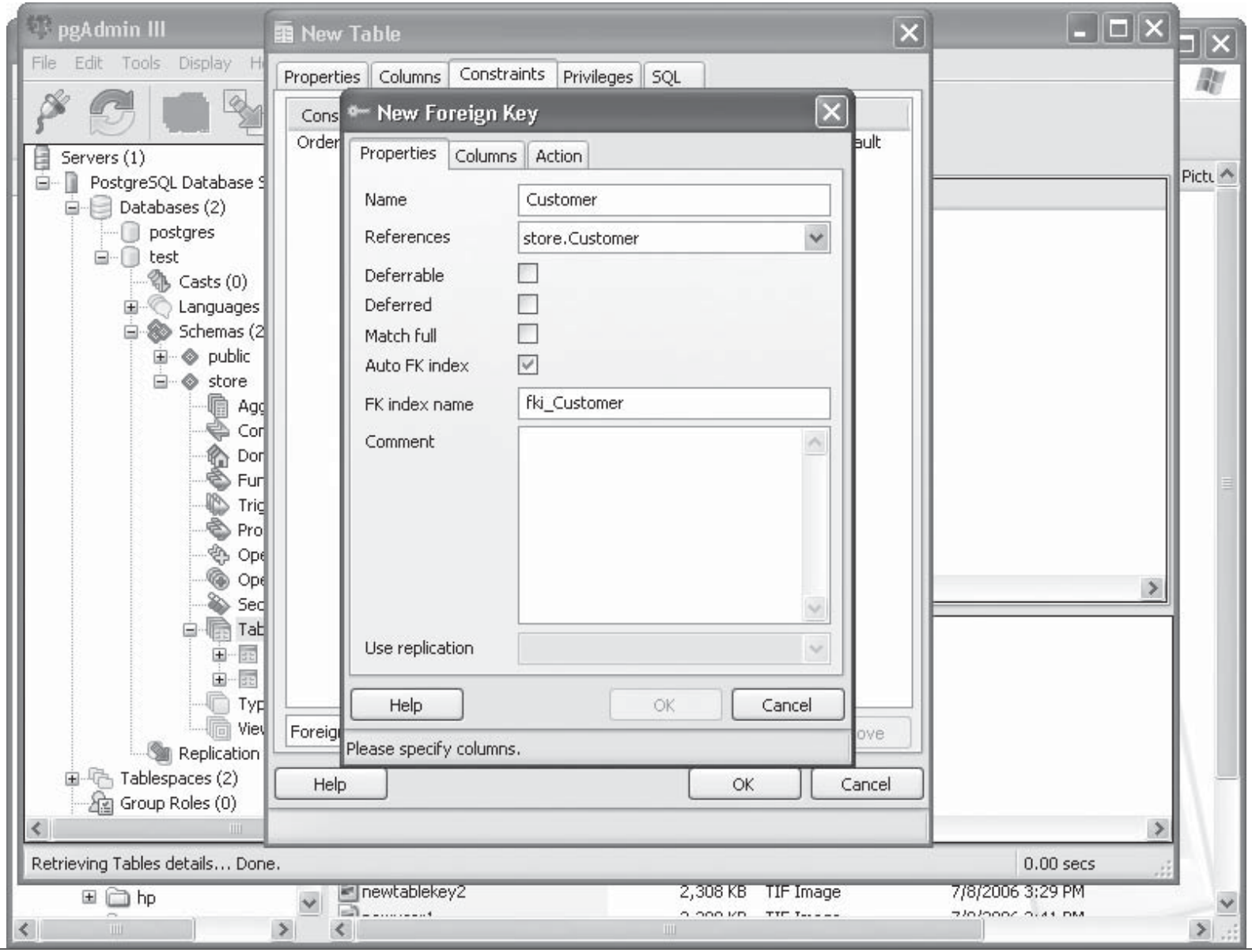
The Product Table Columns

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

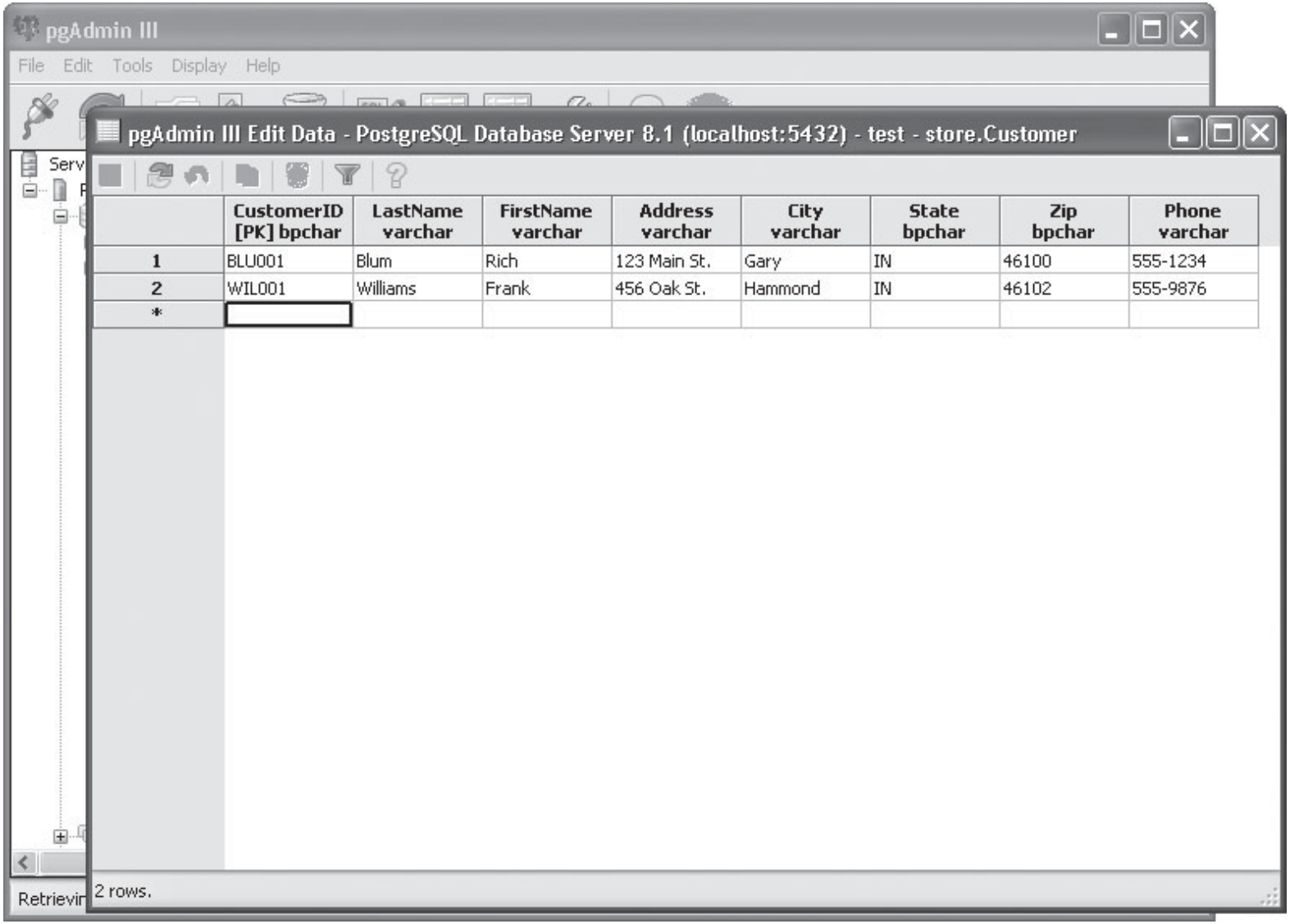
The Columns for the Order Table

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



Entering and Viewing Data



	CustomerID [PK] bpchar	LastName varchar	FirstName varchar	Address varchar	City varchar	State bpchar	Zip bpchar	Phone varchar
1	BLU001	Blum	Rich	123 Main St.	Gary	IN	46100	555-1234
2	WIL001	Williams	Frank	456 Oak St.	Hammond	IN	46102	555-9876
*								

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] characte	LastName character varyir	FirstName character varying	Address character varying	City character varyi	State character	Zip characte	Phone character var
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]

Order

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

Violating column constraint

pgAdmin 4

File Object Tools Help

Browser

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

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

No limit

PostgreSQL 10 - test2 - store.Customer

```
SELECT * FROM store."Customer"
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

ENG 11:48 AM 2/26/2019

Violating column constraint

pgAdmin 4

File Object Tools Help

Browser

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

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

No limit

PostgreSQL 10 - test2 - store.Order

```
SELECT * FROM store."Order"
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". .

Type here to search ENG 11:46 AM 2/26/2019

THE pgADMIN QUERY TOOL

pgAdmin 4

pgAdmin 4 File Object Tools Help

Browser

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

Dashboard Properties SQL Statistics Dependencies Dependents Query - test2 on pos

No limit

test2 on postgres@PostgreSQL 10

```
1  Select * from store."Customer";
```

Data Output Explain Messages Query History

	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

Type here to search ENG 11:50 AM 2/26/2019

3. Access to database objects: Login/Group Roles

Roles

- Actually, we use postgres account to connect to PostgreSQL system: super role having all privileges.
- PostgreSQL uses the **roles concept** to manage database access permissions. A role can be
  - **a user**: a role that has login right is called user or login role
  - or a group: a role may be a member of other roles, which are known as groups
- **Each database user** should have **an individual account** for logging into the PostgreSQL system
- **pgAdmin** allows you to create Roles and to grant Roles access to database objects

Group Roles

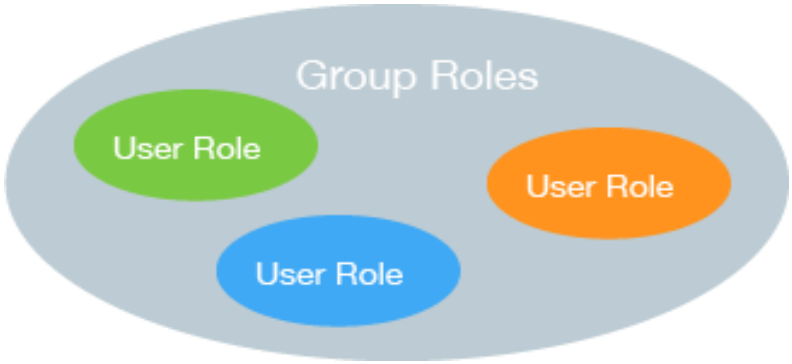
- Create access **permissions for groups of users**
- While you can grant an individual user account access directly to a database object, the preferred method is to use Group Roles
- allow you to **easily change access for database objects** without having to touch hundreds (or even thousands) of individual user.

- Default, **public** group role:
  - applies to all users on the PostgreSQL system
  - NOT able to remove any user account from the public Group Role
  - does not appear in the pgAdmin Group Roles listing

Login Roles (or user accounts)

- Are roles that are allowed to log into the PostgreSQL server

- That account is then **assigned as a member** of the appropriate **Group Roles that grant privileges** to the database objects required



Practices – (continue…)

- Create a database ***test***
  - Customer
  - Product
  - Order
- Create two Group Roles
  - *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
  - salesman - Barney
  - accountant - Fred

WORKING WITH USER ACCOUNTS

pgAdmin III

File Edit Tools Display Help

Servers (1)
PostgreSQL Database Server 8.1
Databases (2)
postgres
test
Casts (0)
Languages (1)
Schemas (2)
public
store
Aggregates (
Conversions (
Domains (0)
Functions (0)
Trigger Functi
Procedures (
Operators (0)
Operator Clas
Sequences (0
Tables (3)
Types (0)
Views (0)
Replication (0)
Tablespaces (2)
Group Roles (2)
Login Roles (3)

New Group Role

Properties | Role membership | Variables | SQL

Role name: Salesman
OID
Can login
Password
Password (again)
Account expires

Role Privileges
- Inherits rights from parent roles
- Superuser
- Can create database objects
- Can create roles
- Can modify catalog directly

Use replication

Help | OK | Cancel

Retrieving Group Roles details... Done. 0.00 secs

pgAdmin III

File Edit Tools Display

Languages

Schemas (2

public

store

Table store.Customer

Properties | Columns | Constraints | Vacuum settings | Privileges | SQL

User/Group	Privileges
postgres	arwdRxt
Salesman	arwd
Accountant	r

Add/Change | Remove

Privileges

Role: public

- ALL — WITH GRANT OPTION
- INSERT — WITH GRANT OPTION
- SELECT — WITH GRANT OPTION
- UPDATE — WITH GRANT OPTION
- DELETE — WITH GRANT OPTION
- RULE — WITH GRANT OPTION
- REFERENCES — WITH GRANT OPTION
- TRIGGER — WITH GRANT OPTION

Help | OK | Cancel

s,Salesman=arwd/postgres,Acc

ue customer identifi

stomer

Retrieving Table details... Done.

0.00 secs

pgAdmin Object Privilege Codes

Code	Privilege
a	INSERT (append)
r	SELECT (read)
w	UPDATE (write)
d	DELETE
R	RULE
x	REFERENCES
t	TRIGGER
X	EXECUTE
U	USAGE
C	CREATE
T	TEMPORARY

Creating Login Roles

pgAdmin III

File Edit Tools Display Help

Servers (1)
- PostgreSQL Database Server 8.1
  - Databases (2)
    - postgres
    - test
      - Casts (0)
      - Languages (1)
      - Schemas (2)
        - public
        - store
          - Aggregates (
          - Conversions (
          - Domains (0)
          - Functions (0)
          - Trigger Functi
          - Procedures (
          - Operators (0)
          - Operator Clas
          - Sequences (0
          - Tables (3)
          - Types (0)
          - Views (0)
      - Replication (0)
  - Tablespaces (2)
  - Group Roles (2)
  - Login Roles (3)

New Login Role

Properties | Role membership | Variables | SQL

Role name: fred

OID:

Can login: ☑

Password: ••••••

Password (again): ••••••

Account expires:

Role Privileges
- ☑ Inherits rights from parent roles
- ☐ Superuser
- ☐ Can create database objects
- ☐ Can create roles
- ☐ Can modify catalog directly

Use replication:

Help | OK | Cancel

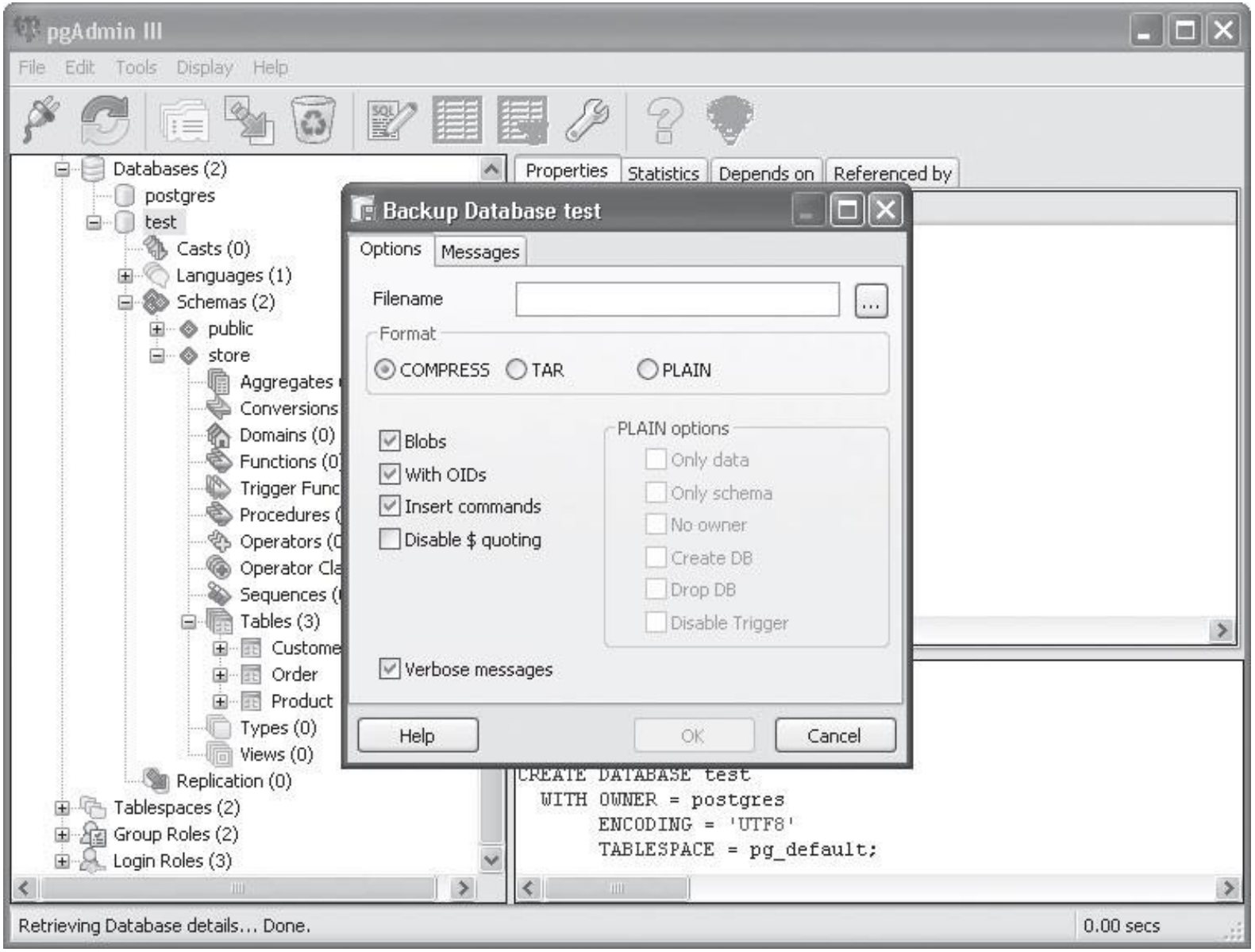
Retrieving Login Roles details... Done. 0.00 secs

Testing

- Login to the test database and the fred Login Role with pgAdmin 4
- Open Query Tool, run following commands and see what will be happen

  - SELECT * from store."Product";
  - INSERT into store."Product" VALUES ('LAP001', 'Laptop', 'TakeAlong', 'Acme', '500.00', 100);
  - INSERT into store. "Customer"("CustomerID", "LastName", "FirstName") VALUES ('Cus001', 'Thi Oanh', 'Nguyen');

BACKUPS AND RESTORES



pgAdmin III

File Edit Tools Display Help

Databases (2)
postgres
test
Casts (0)
Languages (1)
Schemas (2)
public
store
Aggregates
Conversions
Domains (0)
Functions (0
Trigger Func
Procedures
Operators (
Operator Cl
Sequences (
Tables (3)
Custome
Order
Product
Types (0)
Views (0)
Replication (0)
Tablespaces (2)
Group Roles (2)
Login Roles (3)

Properties | Statistics | Depends on | Referenced by

Property	Value

Restore Database test

Options | Contents | Messages

Filename [ ] ...

- [ ] Only data
- [ ] Only schema
- [ ] Single object
- [ ] No owner
- [ ] Disable Trigger
- [x] Verbose messages

Help | View | OK | Cancel

```
CREATE DATABASE test
  WITH OWNER = postgres
       ENCODING = 'UTF8'
       TABLESPACE = pg_default;
```

Retrieving Database details... Done. 0.00 secs