

PostgreSQL

A **QSS** Webinar

October 17, 2012

10:00 AM – 12:00 PM

Audience

- System administrators, operators, technical support staff
- Already familiar with system administration and operations of QSS/OASIS Version H using TurboIMAGE databases
- Presentation is very technical.
- Previous exposure to Linux is helpful.

PostgreSQL

- Version
 - QSS Requires a minimum version of 8.2
 - Examples will be shown using 8.3.x
 - Available versions 8.2, 8.3, 8.4, 9.0, 9.1, 9.2
- Installation
 - From Linux distribution if 8.2 or better
 - From source if not
- Website
 - www.postgresql.org

End of Life Dates

source: <http://www.postgresql.org/support/versioning/>

Version	Current minor	Supported	First release date	EOL date
9.2	9.2.1	Yes	Sep-12	Sep-17
9.1	9.1.6	Yes	Sep-11	Sep-16
9	9.0.10	Yes	Sep-10	Sep-15
8.4	8.4.14	Yes	Jul-09	Jul-14
8.3	8.3.21	Yes	Feb-08	Feb-13
8.2	8.2.23	No	Dec-06	Dec-11

QSS Databases

- Version 'L'

- finance

- hrspay

- pque

- strsmf

- sysctl

- Version 'H'

- GLDSYS, ACTRCV, BDEVOL, CONVRT, ECOMRC, FASSET, RECON, STRSYS

- PERPAY, BENMGT, PPHIST, PYHIST, RETSYS, STTSYS

- PQUE

- STRSMF

- SYSCTL, MENU, MENCTL, QDACTL, QSSPRE

PostgreSQL Database Admins

- Are valid Linux users
- postgres (usually has no password set)
 - PostgreSQL's admin user, not used by QSS
 - Created during PostgreSQL install
- qssdba - QSS's database administrator
 - create/alter/drop databases
 - create/alter/drop tables in databases
 - read/write access to all data in all databases
 - has superuser role

Additional User/Roles

- Are database users only, not real Linux users.
- qssdbr
 - read only access to all data in all databases
- qssdbu
 - read/write access to all data in all databases
- SQL Scripts for creating qssdbr / qssdbu available on version 'L' Linux server
 - PostgreSQL: /opt/qss/install/sql/qssdbx-pg.sql
 - Needs to be re-run after any database changes which create tables

psql - PostgreSQL interactive terminal

- Command line, text-based interface
- Connects to PostgreSQL database
- View structure of database and/or tables
- Maintain structure of database and/or tables using SQL data definition language (ddl) commands
- View or Maintain data using SQL data manipulation language (dml) commands

List databases

```
qssdba@linux-demo:~> psql -l
```

```
      List of databases
```

Name	Owner	Encoding
finance	qssdba	SQL_ASCII
hrspay	qssdba	SQL_ASCII
hrsweb	qssdba	SQL_ASCII
postgres	postgres	UTF8
pque	qssdba	SQL_ASCII
strsmf	qssdba	SQL_ASCII
sysctl	qssdba	SQL_ASCII
template0	postgres	UTF8
template1	postgres	UTF8

(9 rows)

To connect to a database

- `psql <database>`

```
qssdba@linux-demo:~> psql finance
```

```
Welcome to psql 8.3.5, the PostgreSQL interactive terminal.
```

```
Type:  \copyright for distribution terms  
       \h for help with SQL commands  
       \? for help with psql commands  
       \g or terminate with semicolon to execute query  
       \q to quit
```

```
finance=#
```

Commands inside psql

- Any valid SQL statement
- \q to exit
- \h for list of sql commands
- \h <sql command> for help on the command
- \? for help on PostgreSQL commands
- psql output is piped to less so you can move forwards, backwards, search, etc. Use 'q' to exit viewing output.
- If PostgreSQL built correctly psql has same redo, auto-completion features as bash

Show list of tables

```
finance=# \d
```

```
      List of relations
```

Schema	Name	Type	Owner
public	ac_bdg_detail	table	qssdba
public	account	table	qssdba
public	account_sum	table	qssdba
public	acct_sum_ctl	table	qssdba
public	acct_sum_rule	table	qssdba
public	acq_reason	table	qssdba
public	ap_detail	table	qssdba
public	ap_header	table	qssdba
public	apc_odometer	table	qssdba
public	apy_accounts	table	qssdba
public	apy_x_ref	table	qssdba
public	ar_cust_type	table	qssdba
public	ar_customer	table	qssdba

```
lines 1-16
```

Show a table layout

```
finance=# \d account
```

```
Table "public.account"
```

Column	Type	Modifiers
di_no	numeric(3,0)	not null
yr_no	numeric(4,0)	not null
acctclass	character(50)	not null
pseudo	character(24)	
status	character(1)	
roll_flag	character(1)	
date_closed	date	
app_bdg	numeric(15,2)	
rev_bdg	numeric(15,2)	
work_bdg	numeric(15,2)	
expense	numeric(15,2)	
pend	numeric(15,2)	
enc	numeric(15,2)	

```
lines 1-16
```

Show data in a table

```
finance=# select * from account where di_no=40 and yr_no=2009;
```

di_no	yr_no	acctclass	status	roll_flag	date_closed	app_bdg	rev_bdg	work_bdg	expense	pend	enc	pre_enc	descr	state_code	date_used	time_used	last_process_name	opr_used	date_changed	opr_changed	time_changed	
40	2009	0100000000000008011890551000	0			120379781.00	107694655.0	107694655.00	-119336394.00	0.00	0.00	0.00			2009-08-24	08:59:08.78	AT010					
0																						
0																						
40	2009	0100000000000008019890551000																				

```
lines 1-4
```

Expanded view of data in a table

```
finance=# \x
Expanded display is on.
finance=# select * from account where di_no=40 and yr_no=2009;
-[ RECORD 1 ]-----+-----
di_no          | 40
yr_no          | 2009
acctclass      | 01000000000000008011890551000
pseudo         |
status         | 0
roll_flag      |
date_closed    |
app_bdg        | 120379781.00
rev_bdg        | 107694655.00
work_bdg       | 107694655.00
expense        | -119336394.00
pend           | 0.00
enc            | 0.00
pre_enc        | 0.00
descr          |
lines 1-16
```

Show number of records in a table

```
finance=# select count(*) from account;
```

```
count
```

```
-----  
173035  
(1 row)
```

```
finance=# select di_no, yr_no, count(di_no) from account group by  
di_no, yr_no;
```

```
di_no | yr_no | count
```

```
-----+-----+-----  
0 | 0 | 1  
40 | 2009 | 34382  
40 | 2011 | 5  
40 | 2008 | 28372  
40 | 2010 | 56717
```

```
(5 rows)
```


Execute a SQL script

```
qssdba@linux-demo:~> psql hrspay
hrspay=# \i sqlppu3.sql
SELECT
DROP TABLE
psql:sqlppu3.sql:277: NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index
"bfte_ctl_month_di_yr_bfte_m_pk" for table "bfte_ctl_month"
CREATE TABLE
psql:sqlppu3.sql:312: ERROR: column "bfte_pos_rel_01" of relation
"bfte_ctl_month_save" already exists
INSERT 0 0
psql:sqlppu3.sql:333: NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index
"emp_odometer_di_ssn_yr_type_pk" for table "emp_odometer"
CREATE TABLE
psql:sqlppu3.sql:353: NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index
"paf_distribution_di_list_pk" for table "paf_distribution"
CREATE TABLE
psql:sqlppu3.sql:366: NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index
"distribution_method_method_pk" for table "distribution_method"
CREATE TABLE
psql:sqlppu3.sql:401: NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index
"bfte_ctl_sched_di_yr_bfte_s_pk" for table "bfte_ctl_sched"
CREATE TABLE
ALTER TABLE
ALTER TABLE
```

Sizes of PostgreSQL databases and tables

- To see the size of a database

```
qssdba@linux-demo:~> psql finance
finance=# SELECT pg_size_pretty(pg_database_size('finance')) As dbsize;
dbsize
-----
2207 MB
```

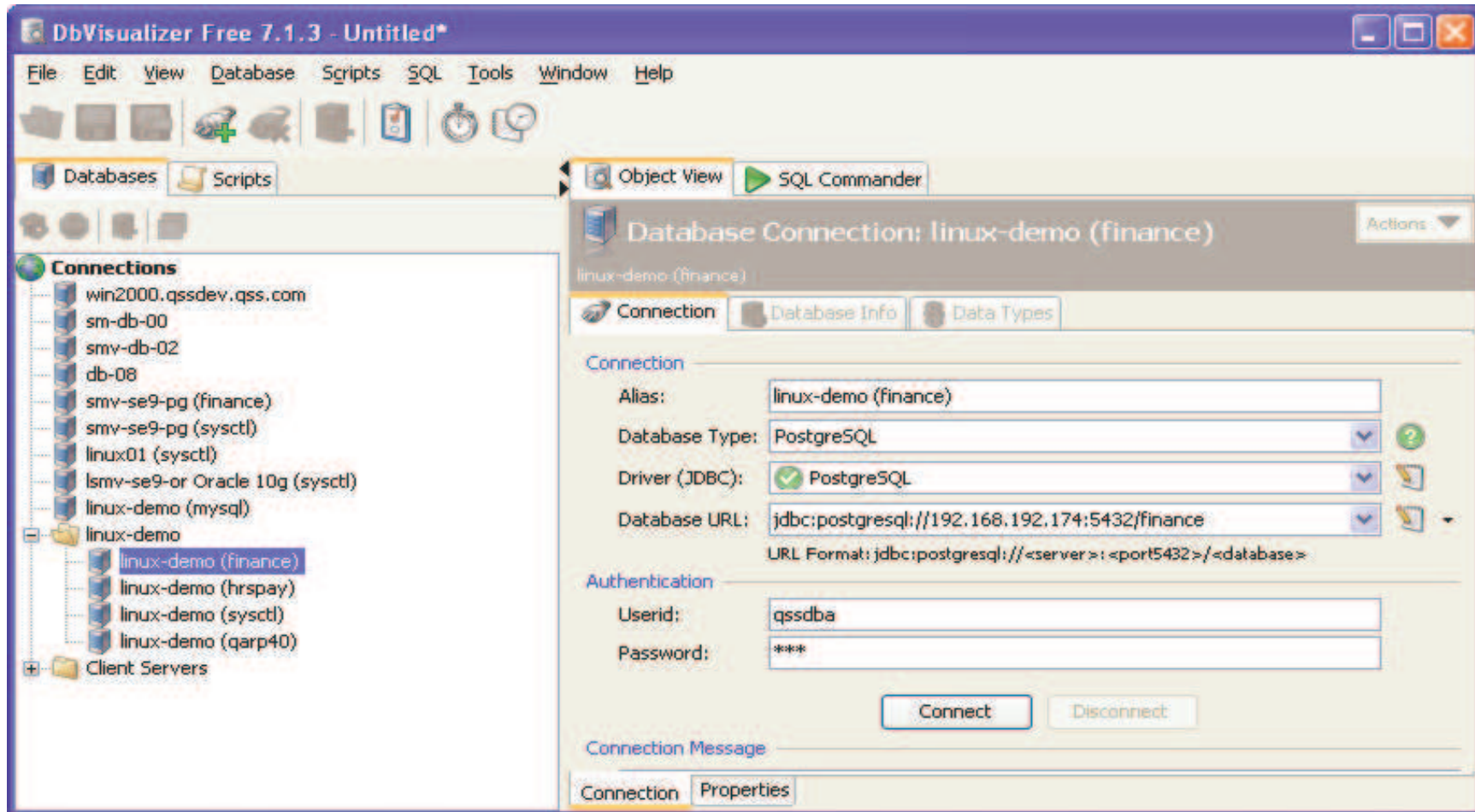
- To see the size of a table's data, or table's data plus indexes

```
finance=# pg_size_pretty(pg_relation_size('public.account')) AS tablesize,
finance=# pg_size_pretty(pg_total_relation_size('public.account')) As totalsize;
tablesize | totalsize
-----+-----
48 MB    | 76 MB
```

DbVisualizer

- Database tool that can connect to most SQL platforms.
- Graphical interface on your desktop that remotely connects to SQL database server.
 - Lets you see list of tables, table structure, table data. Can filter data. Has SQL command interface.
- Free version has fewer features than licensed version but is still very useful.
- Different SQL platforms have differences
 - Slightly different views in object tree (free vs licensed versions have differences too)
 - May be able to connect to all databases or only one

Sample DbVisualizer connection for PostgreSQL



Sample DbVisualizer object tree for PostgreSQL

The screenshot displays the DbVisualizer Free 7.1.3 interface. The left pane shows the object tree for a PostgreSQL database named 'linux-demo'. The tree is expanded to show the 'finance' schema, which contains several objects including 'information_schema', 'pg_catalog', 'pg_toast_temp_1', and 'public'. The 'public' schema is further expanded to show various system objects and tables, with the 'account' table selected.

The right pane shows the metadata for the selected 'account' table. The metadata is displayed in a table format with the following columns: Name and Value.

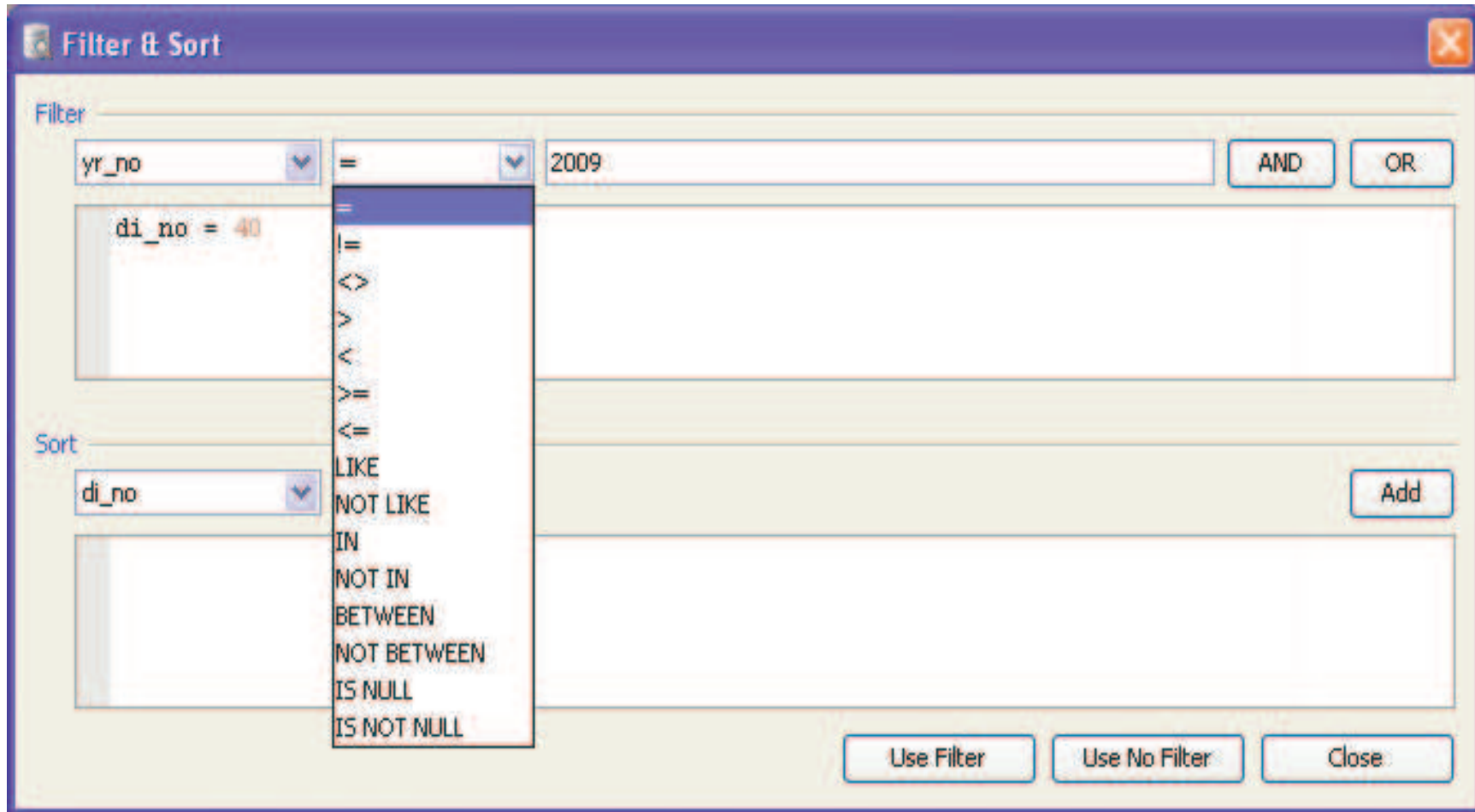
Name	Value
TABLE_CAT	(null)
TABLE_SCHEM	public
TABLE_NAME	account
TABLE_TYPE	TABLE
REMARKS	(null)

Sample DbVisualizer 'Columns' Object View

The screenshot displays the DbVisualizer Free 7.1.3 interface. The left pane shows a tree view of the 'linux-demo' database, with the 'public' schema expanded to show the 'account' table. The right pane shows the 'Columns' object view for the 'account' table, displaying a list of columns with their data types and lengths.

TABLE_CAT	TABLE_SCHEM	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_ID
(null)	public	account	di_no	2	numeri
(null)	public	account	yr_no	2	numeri
(null)	public	account	acctclass	1	bpchar
(null)	public	account	pseudo	1	bpchar
(null)	public	account	status	1	bpchar
(null)	public	account	roll_flag	1	bpchar
(null)	public	account	date_closed	91	date
(null)	public	account	app_bdg	2	numeri
(null)	public	account	rev_bdg	2	numeri
(null)	public	account	work_bdg	2	numeri
(null)	public	account	expense	2	numeri

Sample DbVisualizer 'Configure Filter'

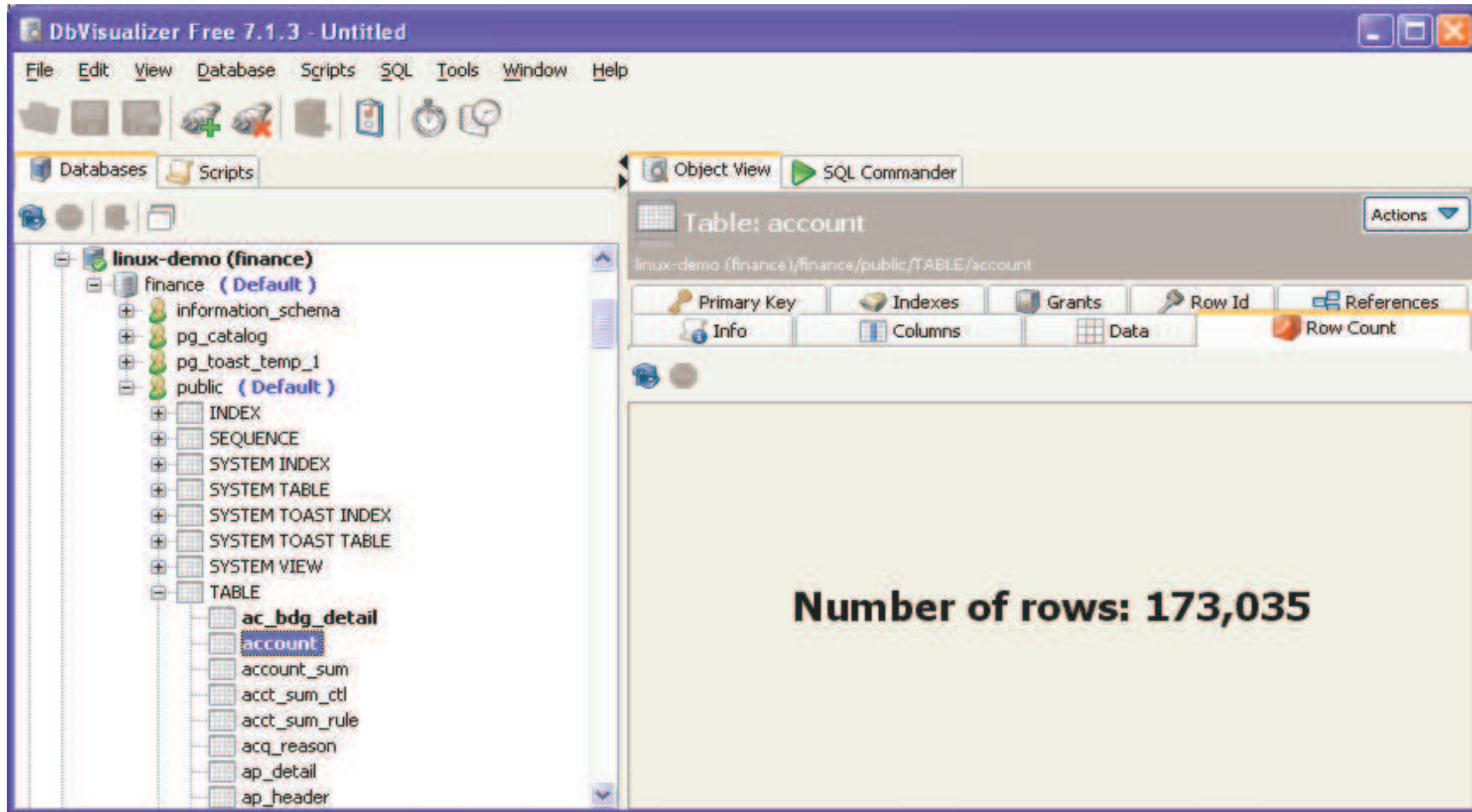


Sample DbVisualizer 'Data' Object View

The screenshot displays the DbVisualizer Free 7.1.3 interface. The left pane shows the database structure for 'linux-demo (finance)', with the 'account' table selected under the 'public (Default)' schema. The right pane shows the 'Data' view of the 'account' table, displaying 11 rows of data. The columns are 'di_no', 'yr_no', 'acctclass', and 'pse'. The status bar at the bottom indicates 'Max Rows: 10000', 'Max Chars: 0', and a query execution time of '19.002/0.844 sec'.

	di_no	yr_no	acctclass	pse
1	40	2008	0100970111010011100360110060	
2	40	2008	0100970111010011100380110060	
3	40	2008	0100970111010011100385110060	
4	40	2008	0100970111010011120300110061	
5	40	2008	0100970111010011150020110060	
6	40	2008	0100970111010011150020110061	
7	40	2008	0100970111010011150040110060	
8	40	2008	0100970111010011150040110061	
9	40	2008	0100970111010011150048110060	
10	40	2008	0100970111010011150048110061	
11	40	2008	0100970111010011150050110060	

Sample DbVisualizer 'Row Count' Object View



Sample DbVisualizer SQL Commander

The screenshot displays the DbVisualizer Free 7.1.3 interface. The left pane shows a tree view of the 'linux-demo (finance)' database, with the 'account' table selected under the 'public (Default)' schema. The central SQL Commander pane contains the following query:

```
1 select * from account
2 where di_no=40 and
3     yr_no=2009
```

The results pane shows the following data:

	di_no	yr_no	acctclass	pseudo
1	40	2009	010000000000000008011890551000	
2	40	2009	010000000000000008019890551000	
3	40	2009	010000000000000008021	
4	40	2009	010000000000000008041	

A warning message is displayed: "Number of rows limited by Max Rows". The status bar at the bottom indicates the execution time as 2.265/0.047 sec and the number of rows as 1000/23.

pgAdmin

- Graphical Interface to PostgreSQL database only
- Installed on remote PC desktop
- Can also do security changes
- Open Source (free)
- Website: <http://www.pgadmin.org/>

pgAdmin - Object tree

The screenshot displays the pgAdmin III interface. On the left, the Object browser shows a tree structure for a server named 'linux-demo (linux-demo:5432)'. The tree includes Databases (13), Schemas (1), and Tables (184). The 'public' schema is expanded, showing various objects like Domains, FTS Configurations, and Tables. The 'ac_bdg_detail' table is selected in the Tables list.

The Properties pane on the right shows the following details for the 'ac_bdg_detail' table:

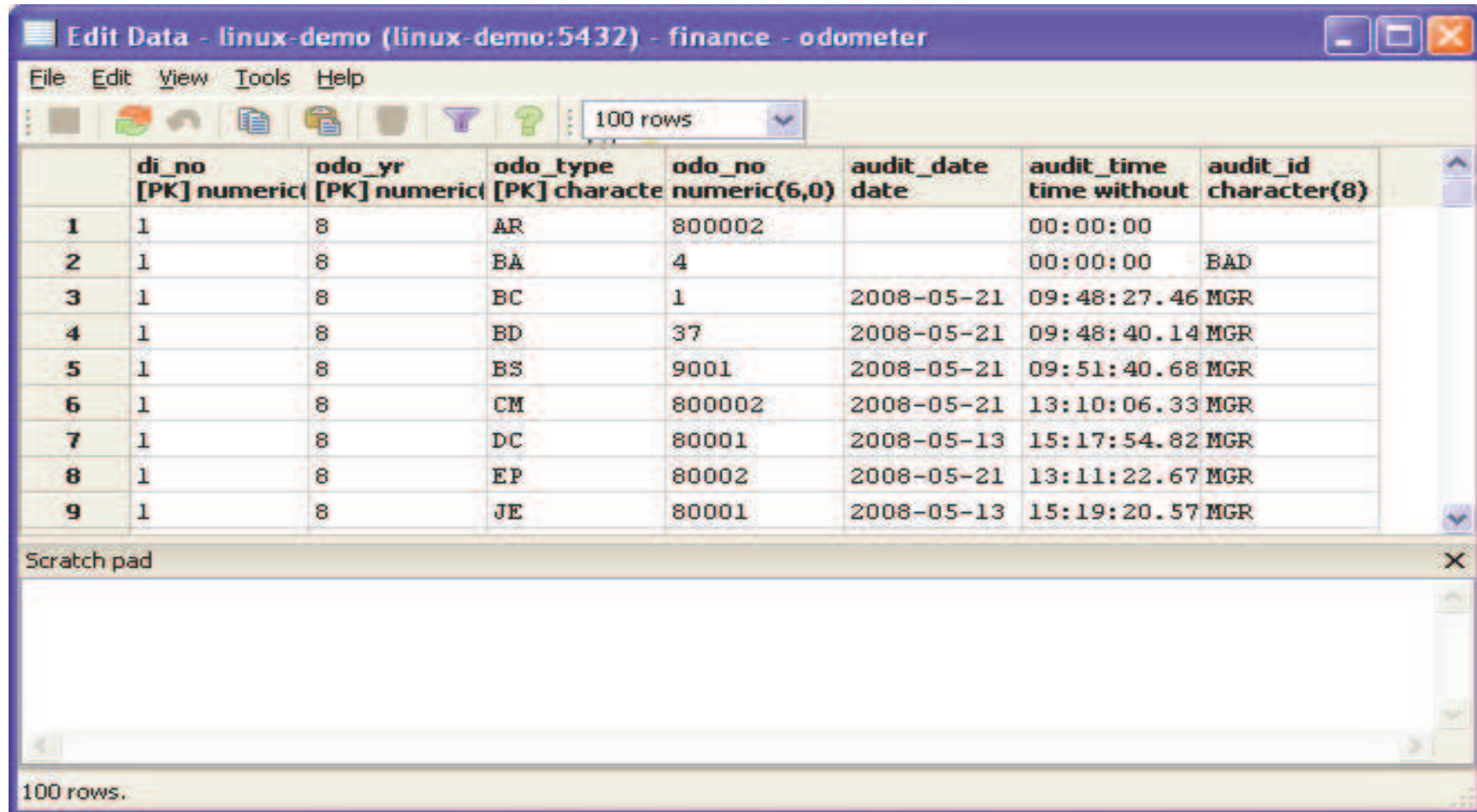
Property	Value
Name	ac_bdg_detail
OID	19505
Owner	qssdba
Tablespace	pg_default
ACL	{qssdba=arwdxt/qssdba,qssdba=r/qssdba,qssdba}
Primary key	<no primary key>
Rows (estimated)	0
Fill Factor	

The SQL pane below the Properties pane shows the following SQL code:

```
-- Table: ac_bdg_detail
-- DROP TABLE ac_bdg_detail;
CREATE TABLE ac_bdg_detail
(
  di_no numeric(3,0),
  yr_no numeric(4,0),
  acctclass character(50),
  descr character(30),
  amount1 numeric(15,2),
  amount2 numeric(15,2),
  amount3 numeric(15,2)
)
```

At the bottom of the window, a status bar indicates 'Retrieving Table details... Done.' and a timer shows '1.13 secs'.

pgAdmin - View Data



The screenshot shows the 'Edit Data' window in pgAdmin for a table named 'odometer'. The window title is 'Edit Data - linux-demo (linux-demo:5432) - finance - odometer'. The table has 8 columns: 'di_no', 'odo_yr', 'odo_type', 'odo_no', 'audit_date', 'audit_time', and 'audit_id'. The data is displayed in a grid with 9 rows. Below the table is a 'Scratch pad' area and a status bar at the bottom indicating '100 rows'.

	di_no [PK] numeric	odo_yr [PK] numeric	odo_type [PK] character	odo_no numeric(6,0)	audit_date date	audit_time time without	audit_id character(8)
1	1	8	AR	800002		00:00:00	
2	1	8	BA	4		00:00:00	BAD
3	1	8	BC	1	2008-05-21	09:48:27.46	MGR
4	1	8	BD	37	2008-05-21	09:48:40.14	MGR
5	1	8	BS	9001	2008-05-21	09:51:40.68	MGR
6	1	8	CM	800002	2008-05-21	13:10:06.33	MGR
7	1	8	DC	80001	2008-05-13	15:17:54.82	MGR
8	1	8	EP	80002	2008-05-21	13:11:22.67	MGR
9	1	8	JE	80001	2008-05-13	15:19:20.57	MGR

pgAdmin - Query

The screenshot shows the pgAdmin Query tool interface. The title bar reads "Query - finance on qssdba@linux-demo:5432". The menu bar includes "File", "Edit", "Query", "Favourites", "Macros", "View", and "Help". The toolbar contains various icons for file operations and query execution. The "SQL Editor" tab is active, displaying the query: `select * from odometer;`. The "Output pane" is visible at the bottom, showing the results of the query in a table format. The table has 7 rows and 7 columns: `di_no` (numeric(3,0)), `odo_yr` (numeric(2,0)), `odo_type` (character(2)), `odo_no` (numeric(6,0)), `audit_date` (date), and `audit_time` (time without time zone). The status bar at the bottom indicates "OK", "Unix", "Ln 1 Col 24 Ch 24", "102 rows.", and "250 ms".

	<code>di_no</code> numeric(3,0)	<code>odo_yr</code> numeric(2,0)	<code>odo_type</code> character(2)	<code>odo_no</code> numeric(6,0)	<code>audit_date</code> date	<code>audit_time</code> time without time zone
1	1	8	AR	800002		00:00:00
2	1	8	BA	4		00:00:00
3	1	8	BC	1	2008-05-2	09:48:27.46
4	1	8	BD	37	2008-05-2	09:48:40.14
5	1	8	BS	9001	2008-05-2	09:51:40.68
6	1	8	CM	800002	2008-05-2	13:10:06.33
7	1	8	DC	80001	2008-05-1	15:17:54.82

Data Compaction - vacuum

- Compacts database files by removing no longer used pages
- May take a while depending on size of database
 - Manually done using PostgreSQL command `vacuum`
 - PostgreSQL command `vacuumdb` is a wrapper for above
 - Autovacuum may be configured in `postgresql.conf`

PostgreSQL vacuumdb Help

```
qssdba@linux-demo:~> man vacuumdb
VACUUMDB(1) PostgreSQL Client Applications VACUUMDB(1)
NAME
    vacuumdb - garbage-collect and analyze a PostgreSQL database

SYNOPSIS
    vacuumdb [ connection-option... ] [ [ --full ] [ -f ] ] [ [ --verbose ] [ -v ] ]
    [ [ --analyze ] [ -z ] ] [ --table | -t table ]
    [ ( column [,...] ) ]
    [ dbname ]

    vacuumdb [ connection-options... ] [ [ --all ] [ -a ] ] [ [ --full ] [ -f ] ]
    [ [ --verbose ] [ -v ] ] [ [ --analyze ] [ -z ] ]

DESCRIPTION
    vacuumdb is a utility for cleaning a PostgreSQL database. vacuumdb will also generate
    internal statistics used by the PostgreSQL query optimizer.

    vacuumdb is a wrapper around the SQL command VACUUM [vacuum(7)]. There is no effective
    difference between vacuuming databases via this utility and via other methods for
    accessing the server.

OPTIONS
    vacuumdb accepts the following command-line arguments:

    -a
    --all Vacuum all databases.

    [-d] dbname
Manual page vacuumdb(1) line 1
```


PostgreSQL Backup

- pg_dump, pg_dumpall
- QSS Automated Task module
 - ‘pgbackup’
- Examples
 - backup finance database example

```
pg_dump -Fc finance > finance.2011-02-18
```

 - -Fc option is the most flexible format and is compressed by default

PostgreSQL Restore

- `pg_restore`
- Examples (assumes previous `pg_dump -Fc`)
 - Restore entire database replacing existing database

```
dropdb finance
createdb -E SQL_ASCII finance
pg_restore -d finance finance.2011-02-18
```
 - Restore entire database to a new database name

```
createdb -E SQL_ASCII finance.2011-02-18
pg_restore -d finance.2011-02-18 finance.2011-02-18
```

PostgreSQL Restore - examples

- Examples (continued)

- Restore one table replacing existing table

- Option 1. drops the table and restores data and table definition but not indexes

```
pg_restore -d finance -t bde_budget -c -v finance.2011-02-18
```

- Option 2. restores data only leaving existing structure in place

```
psql finance -c "DELETE from bde_budget;"
```

```
pg_restore -d finance -t bde_budget -a finance.2011-02-18
```

- Option 3. restore specific entries from dump list file

```
pg_restore -l finance.2011-02-18 | grep bde_budget > finance.restore.list
```

```
vi finance.restore.list #remove lines not associated with desired table
```

```
pg_restore -d finance -L finance.restore.list -c -v finance.2011-02-18
```

PostgreSQL Restore - examples

- Examples (continued)

- Restore one table as table_yyyy_mm_dd

```
# create a temporary database to restore into
createdb -E SQL_ASCII finance.2011-10-26
# restore only what you want
pg_restore -d finance.2011-10-26 -t po_req_header finance.2011-10-26
# rename the table to include the date of the backup
psql finance.2011-10-26 -c "alter table po_req_header rename to po_req_header_2011_10_26;"
# backup the temporary database
pg_dump -Fc finance.2011-10-26 > finance.po_req_header_2011_10_26
# restore the table named with the date for the temporary database backup.
pg_restore -d finance -t po_req_header_2011_10_26 finance.po_req_header_2011_10_26
#cleanup, temp database, temp backup file, etc.
dropdb finance.2011-10-26
rm finance.po_req_header_2011_10_26
```

QSS Automated Task Module - pgbackup

- Automated method to backup PostgreSQL databases
- Configuration file
 - /etc/opt/qss/qat.d/pgbackup
- Has naming conventions for 3 types of backups
 - Daily, weekly, monthly
 - Definition for end of week, end of month is configurable
- Based on naming conventions files can be saved until manually deleted or overwritten

PostgreSQL Automatic Startup

- Automatic start is setup via Linux
 - Use distribution specific GUI tool to manage servers run levels or `chkconfig`

Starting / Stopping PostgreSQL

- Using init script

- Done from 'root' user

```
linux-demo:~ # /etc/init.d/postgresql
Usage: /etc/init.d/postgresql {start|stop|status|try-restart|restart|force-reload|reload|probe}
linux-demo:~ # /etc/init.d/postgresql status
Checking for PostgreSQL:                                running
```

- Using PostgreSQL command

- Done from 'postgres' user

```
linux-demo:~ # su - postgres
postgres@linux-demo:~> pg_ctl --help
pg_ctl is a utility to start, stop, restart, reload configuration files,
report the status of a PostgreSQL server, or signal a PostgreSQL process.
```

Usage:

```
pg_ctl start    [-w] [-t SECS] [-D DATADIR] [-s] [-l FILENAME] [-o "OPTIONS"]
pg_ctl stop     [-W] [-t SECS] [-D DATADIR] [-s] [-m SHUTDOWN-MODE]
pg_ctl restart  [-w] [-t SECS] [-D DATADIR] [-s] [-m SHUTDOWN-MODE]
                [-o "OPTIONS"]
pg_ctl reload   [-D DATADIR] [-s]
pg_ctl status   [-D DATADIR]
```

PostgreSQL config files

- Found in PostgreSQL's data directory
 - For example: /var/lib/pgsql/data
- PostgreSQL Client Authentication Configuration File
 - pg_hba.conf
- PostgreSQL configuration file
 - postgresql.conf

pg_hba file

- **Methods**

- “trust” connections are trusted without any authentication and should be use only on local connections such as defaults setup for ‘localhost’
- "md5" is preferred since it sends encrypted passwords.

```
# TYPE  DATABASE  USER          CIDR-ADDRESS  METHOD
# "local" is for Unix domain socket connections only
local  all       all           trust
# IPv4 local connections:
host   all       all           127.0.0.1/32  trust
# IPv6 local connections:
host   all       all           ::1/128       trust

# qss
host   all       all           173.164.188.61/32  md5
```

postgresql.conf file

- More info available at <http://www.postgresql.org/docs/8.4/static/runtime-config.html>

```
listen_addresses = '*'          # default is 'localhost'
port = 5432                    # default is 5432
max_connections = 5000         # default is 100
shared_buffers = 256MB        # default is 32MB
max_prepared_transactions = 5000 # >= max_connections
max_locks_per_transaction = 64 # min 10, default is 64
standard_conforming_strings = on
```

Shared Memory Usage

- Increasing max_connections may require changes to other PostgreSQL settings as well as kernel settings.
- See the following for more details.
<http://www.postgresql.org/docs/8.4/static/kernel-resources.html>