

How Oracle DBAs can get started with Postgres

By Timothy Steward
April-13-2021

Thinking about becoming a PostgreSQL dba and puzzled on where to start. Maybe recent articles published by InfoWorld ranking PostgreSQL in fourth place amongst the world of database giants has sparked your interest. Leading you to search for the right book or spending countless hours searching google for the magical website to give you the answers you need. When the magic to learning postgres sits in the mind of an Oracle dba.

Years, ago my first introduction to becoming an Oracle dba was from a mentor who taught me the most valuable lesson. Which was in order to be an Oracle dba one had to be able to pay attention to detail, this lesson started my dba career.

With that in mind as a detailed oriented Oracle dba, one should break the task into 4 key areas.

- Installation
- Architecture
- Security
- Administration

Focusing on the key areas keep in mind that postgres, postgres sql, PostgreSQL that database with many pronunciations is a true RDBMS with many advanced features and less not forget it is ACID compliant to provide transactional integrity along with having MVCC multiversion currency control.

Installation of your PostgreSQL database as your first challenge will put the controls in the hands of the dba to start. You need to make sure you have access to run an rpm install; knowing that Linux is the more popular platform uses all flavors from RHEL to Debian to Ubuntu and definitely Centos are options. Even windows for those that may be intrigued by a more GUI touch and feel environment. Navigation around a yum repository may be the only assistance you need from your systems administrator if you're not familiar. Verification of the packages needed will be important, it may help to make sure you are obtaining a certified combination. For your operating system. Postgresql.org or Enterprisedb.com are a couple of valid locations for download of the software and instructions.

Architecture concepts within PostgreSQL won't confuse you unless you try to grasp the entire database in one lesson. Remember it has been around for many years as has evolved into a mature relational database. The fourth on the list to be exact, and moving up fast. Focusing on the key internal elements that will be critical to understand and will help ease your way into PostgreSQL. First realize its' a database based on processes, meaning you will have several running from the moment the database is started. In Oracle the listener is a process that accepts the incoming connections. In PostgreSQL a similar function is provided by the



postmaster, which will spawn a separate process for each connection. When the postmaster is up and running you will notice it creates the postmaster.pid file containing the main process id. One of the key internal elements would be the wal log, Think redo the home of your transaction logs. This is probably one of the most notable features of PostgreSQL, since it's directly related to the streaming replication process which can be synchronous or asynchronous. The next element would be the memory, that's tunable just like the SGA. Equipped with a local memory to control query specific tuning for order by clauses, re-indexing, vacuuming, and the shared memory controlling server specific tuning to help with the prevention of data loss.

Local Memory Area	Shared Memory Area
Work_mem	Shared Buffer Pool
Maintenance_work_mem	WAL Buffer
Temp_buffers	Commit Log

Securing any database is vital, but understanding the difference between how you define your Oracle security and the PostgreSQL security limitations is crucial. You don't have the advanced security option, but understanding the basic concept can help. In Oracle we have users, roles and groups to help with segregation of duties. We can still accomplish this but you have to understand that you still have a role concept. A user and a group are actually a type of role, so the concept is much different from Oracle where you use the role as a grouping of permissions that you assign. In PostgreSQL the role can actually login to the database. The simplest way to grasp the concept is to visualize two entities.

Entity	Description
User Login	Actual database user
Group/Role	Assign to users as a set of privileges

Once created the only difference would be the user has permissions to login to the database upon creation, the Group/Role has to be granted permission as a secondary step, Similar to Oracle you have the capability to grant or revoke all permissions assigned.

With users' roles and groups, let's talk about schemas. Do you ever remember having to create a schema in Oracle? The create schema command is there, but does it really work would be another question? It definitely works but not how you may think. In Oracle the user is all that you need, if you issue create schema it will create a user for you. This user will be the owner of your future objects. In PostgreSQL a schema is an actual namespace, which sounds confusing.

In Oracle you create a user and have that user own objects to organize your data. The PostgreSQL schema is created for that same reason. Your schema will be based on your organized tables or views that support a particular application. PostgreSQL attempts to make security easy so using your Oracle concepts will still work.

Command	Description
---------	-------------



Create schema payroll	Creating a new schema
Alter table salary set schema payroll	Create a table in the new schema
Drop schema payroll cascade	Drop schema including contents

Administering of your new PostgreSQL environment may be easier than you think. Similar to making an Oracle connection where you need to use the listener port, SID and the username. Your PostgreSQL connections will start with the same information. The common 1521 listener port will be similar to the PostgreSQL 5432 defaulted database port. Using the SID or database name and host you can establish a command line connection through PSQL. Yes, psql will present some navigation differences to get familiar with, but it contains its' own power.

Psql -h hostname – U username databasename

In the PostgreSQL world you will find that the commands tend to get right to the point and will be simple to use taking the form of a single letter opposed to words. For example, to simply describe a table the command is: \d tablename

How many times have you had to select all the tables from the dictionary, in that case just remove the tablename from the command and it works. It offers simple commands for listing privileges as well.

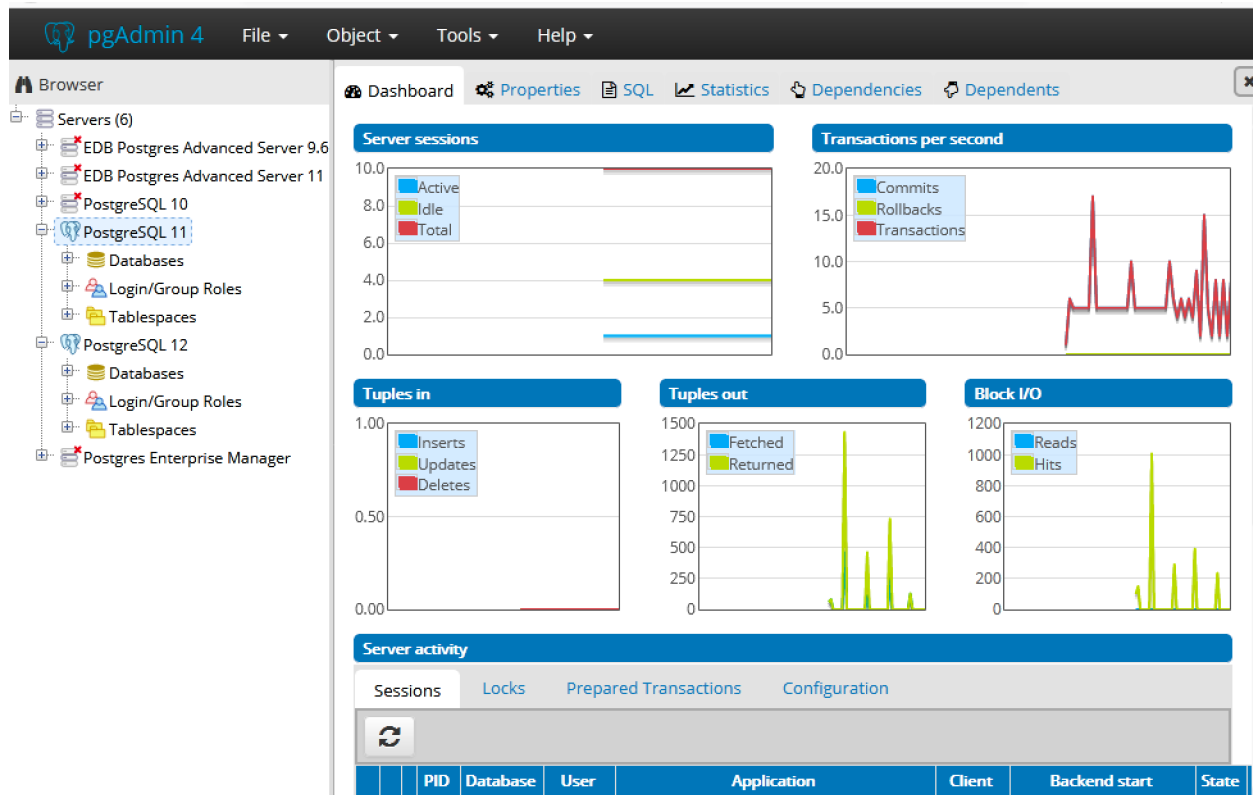
\ Command	Describe
\h	Sql command help
\du	List Users/Roles
\d tablename	Describe a table
\dT	List data types
\dn	List schemas

No worries you can still write your power sql scripts against the catalog/dictionary tables. Such as the sql scripts a dba uses to monitor the V\$ views and dictionary for locks, sql activity, and object permissions can be created. Here is my top 10 list of administration tables.

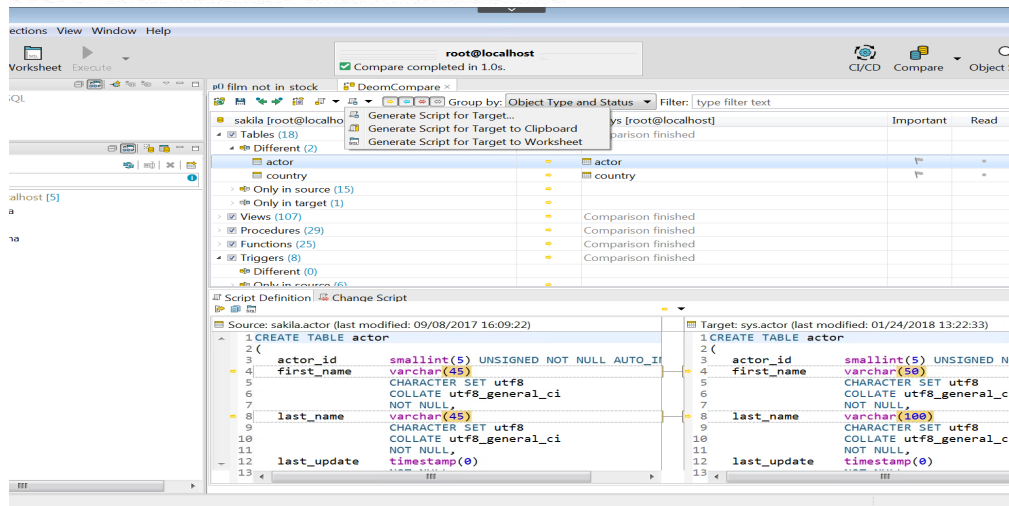
Table	Description
PG_DATABASE	Database details including max connections
PG_CONSTRAINT	View primary, unique and foreign key and constraint details
PG_EXTENSION	Details on any add ons/extentions you added such as postgis
PG_INIT_PRIVS	Privileges granted on objects
PG_LOCKS	Locks held on objects, use with PG_DEPENDS and PG_DESCRIPTION for more details
PG_VIEWS	Info of any views
PG_TABLES	Info on any tables
PG_INDEXES	Info on any indexes
PG_USER	Info on all users

PG_SETTINGS	Server parameters at startup
PG_STAT_ACTIVITY	List current transaction activity

By default, you will also get a free GUI tool for basic administration called pgadmin4. This will allow you to easily connect with the same user and port details to view and maintain your databases.



The flexibility of using other familiar tools is also there, with the creation of toad edge you can still gain access and navigate around your database with a similar interface as the toad for Oracle presents.



Postgresql will offer a lot of flexibility much similar to and Oracle database, by flexibility I mean the utilities that you use to maintain the environment. Things like stopping and starting with scripts, or importing and exporting from the database. Thinking about your Oracle_home or Oracle base level directory, your installation will load things into a similar structure so locating the bin directory will be the source of your utilities. My top five are listed.

Command	Description
PG_DUMP	Dump a single database
PG_DUMPALL	Dump everything
PG_RESTORE	Restore
PG_CTL	Start or Stop
PG_CONFIG	View your configuration settings

Tuning your database will definitely be important as you gain experience. You will find configuration files much like the init files within Oracle that house your main tuning parameters. With a similar function some changes will only take affect after a restart.

PG_HBA.CONF	Host base authentication
POSTGRESQL.CONF	Control database performance

Tuning a database requires using a proper methodology, which we won't get into but from an overall maintenance perspective if I share the most important thing you should learn with your first experience with PostgreSQL it will be the Vacuum. Think about the time when you had to defrag in Oracle or tried to reduce chaining of rows. The most important fact is to protect against data lost. Gaining a full understanding of vacuum and using the supplied utility pg_autovacuum will help you fee up space used by deleted or updated rows and prevent a mysterious occurrence of transaction wraparound which can cause data loss.



As an Oracle dba getting started with postgres will not be as challenging as you may think, just remember the concepts you learned administering Oracle can still be applied to PostgreSQL. Always leveraging the ability to pay attention to details as you progress to the more advanced stages, but remember you are not alone. There are tons of blogs, websites and resources out there to guide you along the way. Enterprisedb has a new tutorial website which allows you search for information around the key areas such as installation, and within that area you can narrow down the search for information based on your skill level, be it beginner or advanced. Remember there is only one obstacle keeping the Oracle dba in you from becoming a PostgreSQL dba, it's not a magical book or website it's truly just one download.