Basic introduction into pgAdmin III and SQL queries

Christina Robinson

Contents

Manuals and documentation produced by Oxford Archaeology North:

Hodgkinson, Anna (2010) *Open Source Survey & GIS Manual.* Documentation. Oxford Archaeology North. (Unpublished): http://library.thehumanjourney.net/367/

Robinson, Christina and Campbell, Dana and Hodgkinson, Anna (2010) *Archaeological maps from qGIS and Inkscape: A brief guide.* Documentation. Oxford Archaeology North. (Unpublished): http://library.thehumanjourney.net/366/

Hodgkinson, Anna (2011) *Using the Helmert (two-point) transformation in Quantum GIS.* Documentation. Oxford Archaeological Unit Ltd.. (Unpublished): http://library.thehumanjourney.net/462/

Software sources and downloads:

Qqis: http://qgis.org/

pgAdmin: http://www.pgadmin.org/

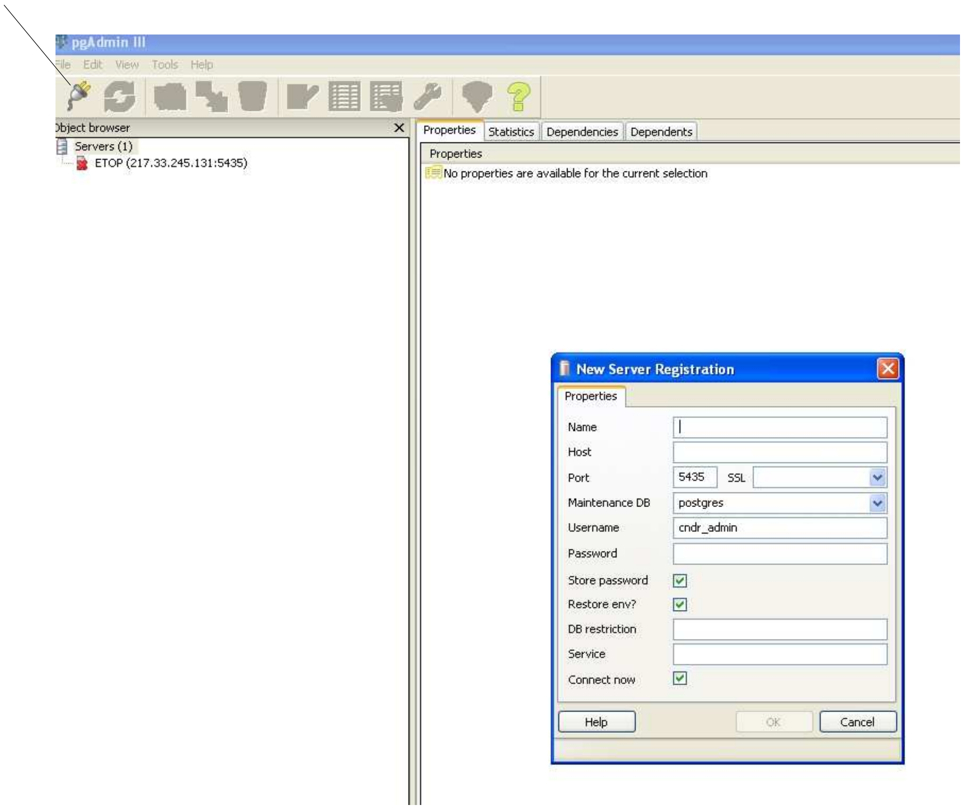
Introduction

This is a very basic introduction into pgAdmin III and how to add a server and get started with writing SQL queries. There are a number of basic queries that have been written below and some basic syntaxes for SQL queries along with some basic rules you need to follow when writing SQL queries.

Getting started with pgAdmin III

Once pgAdmin III has been installed a server will be needed to be connected, this is done through the little plug icon. The 'new server registration' box will appear and the details of the server should be entered (get the details of your server from the appropriate member of staff or see http://pgadmin.org/docs/1.12/connect.html for further details on what is required for the server details).

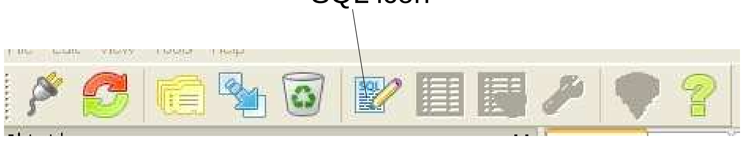
Server connection icon



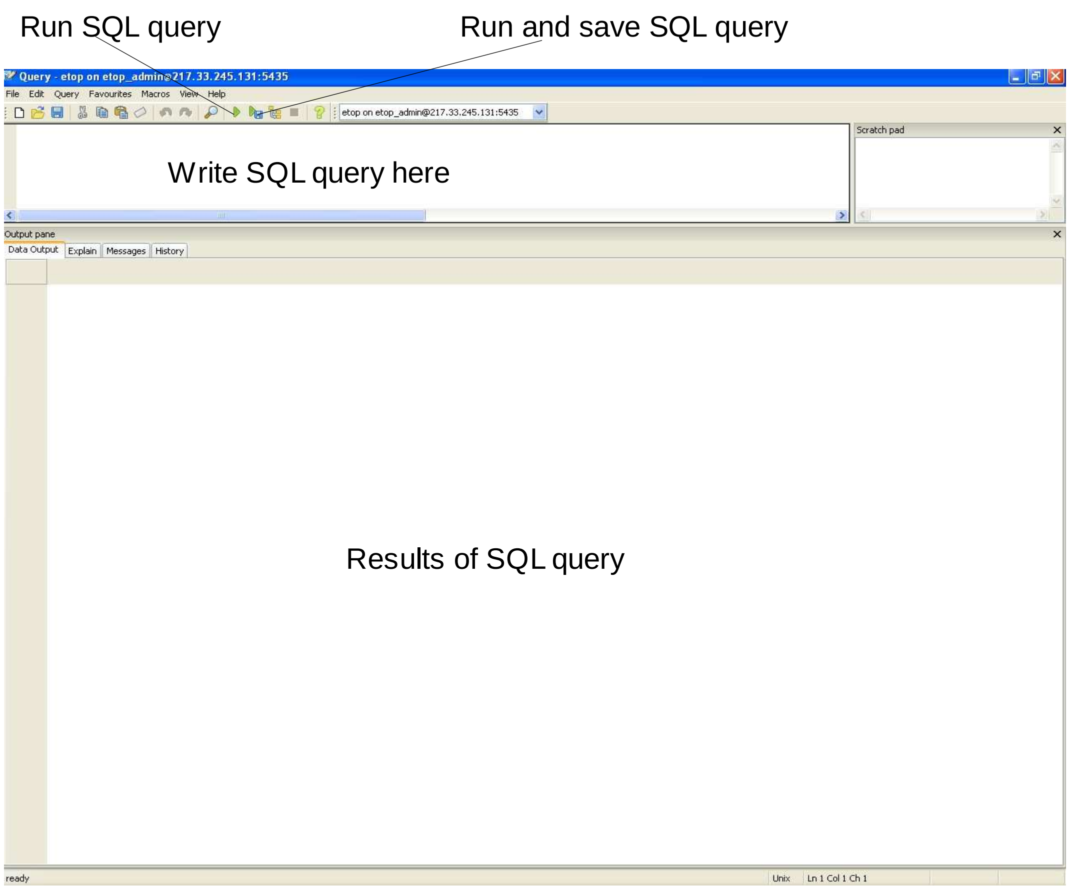
Name	
Host	
Port	5435 SSL
Maintenance DB	postgres
Username	cndr_admin
Password	
Store password	✓
Restore env?	✓
DB restriction	
Service	
Connect now	✓

The server should now be shown in the left hand box and you can navigate your way round your database tables. To query the data within these tables an SQL query is necessary, clicking the SQL icon will produce another window in which the query should be written.

SQL icon



Below is screen shot of the SQL window where the queries are written.



Below is a selection of queries that can be run through pgAdmin III.

<u>How to write SQL queries in pgAdmin III</u>

To start here are some basic rules that need to be followed to get the queries to run and also some of the basic syntaxes used in SQL language.

All table headings need to be lower case and without spaces within pgAdmin III and when using table headings, column headings or values they need to be written as they are in the database.

The <> are not need as syntax, they are there just as a reference as to what needs replacing.

Syntaxes and their meanings for pgAdmin III:

*	all from a table
“ “	needs to be added to column headings
‘ ‘	needs to be added to any values
INSERT	makes a new record
\	means escape
\\	need to appear if backslash wanted in text
::	CAST: defines the data type wanted
||’ ‘||	adding a string to join two columns together in same table
WHERE	places a condition on a query
SELECT	selects values requested
UPDATE	updates values with selected value or value from another table
CREATE VIEW	creates a view from selected items with geometries
SRID	Spatial Reference ID
centroid	middle point of a polygon
%	used to identify value that begins with or ends with a certain suffix (785%)
LIKE	used to select something that is similar to
IS NULL	has a null value
NOT NULL	not a null value
AND	lets you join tables together with matching values and also always to have multiple conditions

Basic queries for pgAdmin III

General UPDATE queries:

UPDATE <table name wanting to be updated> SET <”column which is wanting to be updated”> = <table to be updated from>.<”and column”> FROM <table to be updated from> WHERE <table to be updated>.<'column value to be matched'>=<table to be updated from>.<'column value to be matched'>

Specific update queries:

Updating geometry from x and y:

UPDATE <table name> SET “the_geom” = ST_SetSRID(ST_makepoint (“Eastings”, “Northings”), 27700)

Updating/converting characters into integers within the same table:

UPDATE spe5_cnxt SET “cnxt” = spe5_cnxt.”CONTEXT”::integer

Updating a table with specific value:

UPDATE tbl_name SET “column_name”=’value you want adding’ WHERE “column_name” = ‘value’

Creating a point geometry from the centroid of a polygon geometry:

UPDATE table name SET “Eastings” = x (centroid (geom_ctx)) FROM tbl_contexts WHERE table-name.”column heading” = table name.”column heading”

SELECT queries:

SELECT * FROM <table heading> WHERE <”column heading”> = <‘value’ >AND <”column heading”> =<‘value'>

This selects items that begin with a certain value, other way round is a value ending with that certain value:

SELECT “<column name>” FROM <table name> WHERE “<column name>” LIKE ‘value%’

This query is to compare differences of numbers between two tables/vector layers and also contains a sub query:

SELECT e.ep09.grp FROM ep09.grp e WHERE e.ep09.grp NOT IN (SELECT e.ep09.grp FROM ep09_grple, groups_for_gis g WHERE e.ep09.grp = g.context)

The first full stop in the e.ep09.grp, is just an abbreviation of the table heading.

Joining tables:

To join tables within a query that have matching values add

AND <table name>.”<column name>”=<table name>.”<column name>”

to the end of a query.

CREATING VIEWS:

These are to create views from pgAdmin that then can be loaded into QGIS.

CREATE VIEW view name AS SELECT * FROM table heading WHERE “column heading” = ‘value’

When creating a view you also need to create permissions so that the views can be updated via the GIS. These are the permissions you need:

ALTER TABLE <view name> OWNER TO <database>
GRANT ALL ON TABLE <table name> TO <database>
GRANT SELECT ON TABLE <table name> TO <backup>
GRANT SELECT, UPDATE, INSERT ON TABLE <table name> TO <database>

There are also rules which need to be added, again, so editing of the views can take place within the front end of the database and GIS (this will also edit the information in the database).

<u>Front end editing</u>
CREATE OR REPLACE RULE <rule name> AS UPDATE TO <view name> DO INSTEAD UPDATE <table name> SET “<column heading”> = <new tbl>.<”new column”>

and

<u>GIS editing</u>
CREATE OR REPLACE RULE name of rule AS ON UPDATE TO name of view DO INSTEAD UP DATE table where geom stored SET the_geom=new.the_geom WHERE table where geom stored.OID=new.OID

To upload these views that have been created into QGIS please take a look at the QGIS handy hints here: http://library.thehumanjourney.net/657/