



Introduction to PostGIS

Tutorial ID: IGET_WEBGIS_002



This tutorial has been developed by BVIEER as part of the IGET web portal intended to provide easy access to geospatial education. This tutorial is released under the Creative Commons license. Your support will help our team to improve the content and to continue to offer high quality geospatial educational resources. For suggestions and feedback please visit www.iget.in.

Introduction to PostGIS

Objective

In this tutorial we will learn, how to register a new server, creating a spatial database in PostGIS, importing data to the spatial database, spatial indexing and some important PostGIS special functions.

Software: OpenGeo Suite 3.0

Level: Beginner

Time required: 3 Hour

Software: OpenGeo Suite 3.0

Level: Beginner

Prerequisites and Geospatial Skills

- Basic computer skills
- IGET_WEBGIS_001 should be completed
- Basic knowledge of <u>SQL</u> is excepted

Readings

1. Introduction to the OpenGeo Suite, Chapter 2: PostGIS, pp. 21 – 38.

http://presentations.opengeo.org/2012 FOSSGIS/suiteintro.pdf

2. Chapter 13. PostGIS Special Functions Index, http://suite.opengeo.org/docs/postgis/postgis/html/PostGIS Special Functions Index.html

Tutorial Data: The tutorial data of this exercise may be downloaded from this link:



Introduction

PostGIS is an open source spatial database extension that turns PostgreSQL database system into a spatial database. It adds support for geographic objects allowing location queries, analytical functions for raster and vector data, raster map algebra, spatial re-projection, network topology, Geodetic measurements and much more.

In this tutorial we will learn how to register a server and creation of a spatial database, and importing the shapefiles into it, for this purpose we are using the shapefiles digitized during the tutorial *IGET_GIS_005*: *Digitization of Toposheet using Quantum GIS*. We will also learn about the Spatial Indexing and PostGIS Special functions in this tutorial.

Registering new server

- 1. Start OpenGeo Suite Dashboard via, Start \rightarrow All Programs \rightarrow OpenGeo Suite 3.0 \rightarrow OpenGeo Suite Dashboard.
- 2. In OpenGeo Suite Dashboard click on Green 'Start' button to start the server.
- 3. Start the 'pgAdmin III' interface by Clicking on 'Manage' link in 'PostGIS' section under 'Components'.

OpenGeo Suite Dashbo	ard	
	OpenGeo Suite 30	Shutdown
Dashboard Components	At a Glance	
Logs	Datastores	Import data to PostGIS or GeoServer Manage databases
Preferences	Layers	Style layers Edit layers Cache layers
Documentation	Maps	Compose maps Recipe Book
Getting Started FAQ About	Components	Manage Import shapefiles About Documentation
	GeoServer	Configure Import data About Documentation
	GeoWebCache	Configure About Documentation
	SeoExplorer	Launch About Documentation
	Client SDK	About Tutorial API Documentation
🛞 OpenGeo		

4. Now you will present with '*pgAdmin III*' popup window as shown below. To add new server, click on '*File* \rightarrow Add Server' or click on **'**Add a connection to a server' located on the top left corner.

	🌳 pgAdmin III	_
	File Edit Plugins View Tools Help	
4	<u> </u>	
	Objection of the asserted as a server.	×
	Server Groups	

5. 'New Server Registration' window will popup, fill the information as shown in the below figure to register 'BVU' as new server, click on 'OK' to finish.

New Serv	er Reg	gistration 🛛 🔍
Properties	SSL	Advanced
Name		BVU
Host		localhost
Port		54321
Service		
Maintenance	e DB	postgres -
Username	_	postgres
Password		•••••• postgres
Store passw	ord	
Colour		
Group		Servers 🗸
Help		OK Cancel

6. Once you clicked on 'OK', you might be presented with a 'Guru Hint - Saving passwords' window, click on 'OK' in it. After few seconds, you will notice 'BVU' server added to the 'Server Groups' tree located under 'Object browser window'.

6	👎 pgAdmin III
	File Edit Plugins View Tools Help
	🎽 🛃 💼 🍡 🐼 🗍
	Object browser X
	Server Groups Servers (2) BVU (localhost:54321) PostGIS (localhost:54321)

To Create Spatial Database in PostGIS

Spatial database system is a special kind of database system that offers spatial data types in its data model, query language and supports spatial data types in its implementation, providing at least spatial indexing and spatial join methods. Spatial database systems offer the underlying database technology for geographic information systems and other applications (<u>Ralf Hartmut Güting, 1994</u>).

7. Select the **BVU** server \rightarrow Click on the '+' symbol left side to the **BVU** server \rightarrow right click on the 'Databases' \rightarrow Click on 'New Database' to create a new database.



8. Now you will presented with 'New Database' popup window, Fill the following information under 'Properties' tab as shown in the below figure to create a database named 'BVU_IEER_DB'.

Properties	Definition Variables Privileges SQL
Name	BVU_IEER_DB
OID	
Owner	postgres
Comment	This database contains the shapefiles digitised from south Pune Toposheet of BVU, IEER, Pune
Help	OK Cancel

9. Now click on '*Definition*' tab and select the following options as shown in below figure from the down drop list.

9 New Database	X
Properties Definiti	on Variables Privileges SQL
Encoding	UTF8
Template	template_postgis
Tablespace	<default tablespace=""></default>
Collation	▼
Character type	▼
Connection Limit	-1
Schema restriction	
Help	OK Cancel

10. Click on the '*Privilege*' tab \rightarrow 'Check' the Check box of '*ALL*' \rightarrow Now Click On '*OK*' in *New Database* Window

Add/Change Remove Privileges Role public Image: ALL WITH GRANT OPTION Image: CREATE WITH GRANT OPTION Image: Temp WITH GRANT OPTION Image: CONNECT WITH GRANT OPTION	User/Group	Privileges	
Add/Change Remove Privileges Role Role public Image: ALL WITH GRANT OPTION Image: CREATE WITH GRANT OPTION Image: Temp WITH GRANT OPTION Image: CONNECT WITH GRANT OPTION			
Add/Change Remove Privileges Role Role public Image: ALL WITH GRANT OPTION Image: CREATE WITH GRANT OPTION Image: Temp WITH GRANT OPTION Image: CONNECT WITH GRANT OPTION			
Add/Change Remove Privileges Role public Image: ALL WITH GRANT OPTION Image: CREATE WITH GRANT OPTION Image: Temp WITH GRANT OPTION Image: CONNECT WITH GRANT OPTION			
Add/Change Remove Privileges Role Role public Image: Constant of the second			
Add/Change Remove Privileges Role Role public Image: CREATE WITH GRANT OPTION Image: CONNECT WITH GRANT OPTION			
Privileges Role public Image: ALL WITH GRANT OPTION Image: CREATE WITH GRANT OPTION Image: TEMP WITH GRANT OPTION Image: CONNECT WITH GRANT OPTION	Add/Change	Remove	
Role public Image: Constraint of the second seco	Privileges		
Image: CREATE WITH GRANT OPTION Image: CREATE WITH GRANT OPTION Image: CONNECT WITH GRANT OPTION	Role public		
CREATE WITH GRANT OPTION TEMP CONNECT WITH GRANT OPTION	ALL	WITH GRANT OPTION	
TEMP WITH GRANT OPTION CONNECT WITH GRANT OPTION		WITH GRANT OPTION	
CONNECT WITH GRANT OPTION	CREATE		
	CREATE	WITH GRANT OPTION	

11. After few seconds you will notice 'BVU_IEER_DB' database has been added to the 'Databases' under



'BVU' server.



Importing data to PostGIS

Now we are ready to import the data files into our newly created 'BVU_IEER_DB' database.

12. Now go to the 'OpenGeo Suite Dashboard' \rightarrow Click on 'Import Shapefiles' under PostGIS.

OpenGeo Suite Dashbo	pard	
12	OpenGeo Suite 30	Shutdown
Dashboard Components	At a Glance	
Logs	Datastores	Import data to PostGIS or GeoServer Manage databases
Preferences	Layers	Style layers Edit layers Cache layers
Documentation	Maps	Compose maps Recipe Book
Getting Started FAQ	Components	
About	PostGIS	Manage Import shapefiles About Documentation
	GeoServer	Configure Import data About Documentation
	GeoWebCache	Configure About Documentation
	SeoExplorer	Launch About Documentation
	Client SDK	About Tutorial API Documentation
🐼 OpenGeo		

13. Now you will notice 'PostGIS Shapefile Import/Export Manager' popup window on your screen → Click on 'View connection details...' under 'PostGIS connection' section.

13

Jacoba Com	lection					
	١	/iew co	nnection deta	ils		
	u+					
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,					
	Echomo	Table	Coo Column		Mada	Dm
Snapeme	Schema	Table	Geo Column	SRID	mode	RIII
			Add File			
			Add File			
Options	•	Import	Add File	out		Cance
Options		Import	Add File	out		Cance
Options	·	Import	Add File	out		Cance
Options og Window	·	Import	Add File	put		Cance
Options og Window	•	Import	Add File	put		Cance
Options og Window	•	Import	Add File	Dut		Cance
Options og Window	•	Import	Add File	but		Cance
Options og Window	•	Import	Add File	put		Cance
Options og Window	•	Import	Add File	put		Cance
Options og Window	•	Import	Add File	put		Cance

14. Now 'PostGIS Connection' window will popup, enter *username* and *password* as **postgres**, *Server Host* as 'localhost' & '54321', Database as 'BVU_IEER_DB' as show in below figure and click on 'OK'.

14	PostGIS connection		
	PostGIS Connection	on	
	Username:	postgres	
	Password:	••••••	postgres
	Server Host:	localhost	54321
	Database:	BVU_IEER_DB	
- 1			
		OK	

15. If you entered are correct credentials, you will see 'connection succeeded' message in the 'Log Window' as shown below. If not, you have to ensure the credentials entered are same as in Step 5: 'Properties of New server registration'.

15	Log Window Connecting: host=localhost port=54321 user=postgres password='*******' dbname=BVU_IEER_DB Connection succeeded.
1	

16. Now our database is ready to import the shapefiles, to do this click on 'Add File' button under 'Import' tab in 'PostGIS Shapefile Import/Export Manager' window.

Shape file	Schema	Table	Geo Column	SRID	Mode	Rn

17. Browse the location where you have saved the tutorial data i.e., 'IGET_WEBGIS_002_Data' folder in browser window and select all the shape files in the folder and click 'Open'.

Select a Shape File	Select a Shape File							
📝 👌 Bharti Work 🛛 C	penGeo Suite IGET_WEBGIS_002_Data							
<u>P</u> laces	Name	▲ Modified ▲						
Recently Used	🖊 Bank_BVU.shp	4/15/2013						
	🞽 Barren_BVU.shp	4/15/2013						
Desktop	🞽 Busstops_BVU.shp	12/1/2012						
I ocal Disk (C:)	🞽 Colony_BVU.shp	12/1/2012						
Softwares (D:)	🞽 Contour_BVU.shp	4/15/2013						
leisure (E:)	🞽 EducationalInst_BVU.shp	12/1/2012						
Data, Tutorials, Vide	🞽 highway_BVUshp	11/27/2012						
DVD RW Drive (G:)	🔀 HolyPlaces_BVU.shp	12/1/2012						
	🗡 Hospitals_BVU.shp	12/1/2012						
	🞽 Masjid_BVU.shp	12/1/2012						
	💋 OpenSpace_BVU.shp	4/15/2013						
	🞽 Quary_BVU.shp	11/27/2012						
	🕺 ReservedForest_BVU.shp	11/27/2012						
	🞽 Scrub_BVU.shp	11/27/2012						
	🕺 StreetRoads_BVU.shp	11/27/2012						
	🞽 Urban_BVU.shp	4/15/2013						
	🞽 UrbanGreen_BVU.shp	11/27/2012						
	🞽 Water_BVU.shp	11/27/2012 🚽						
a∰a <u>A</u> dd ann <u>R</u> emove		Shape Files (*.shp) 🔻						
		💥 <u>C</u> ancel 📄 <u>O</u> pen						

18. Now you can see the list of selected shapefiles under *Import list*, click on '*Import*' button to start the

import process.

Shapefile		Schema	Table	Geo Column	SRID	Mode	Rm	-		
F:\Bharti Work\OpenGeo Suite\IG	ET_WEBGIS_002_Data\Agi	riculture_public	agriculture_bvu	geom	0	Create				
F:\Bharti Work\OpenGeo Suite\IG	ET_WEBGIS_002_Data\ATI	M_BVU public	atm_bvu	geom	0	Create				
F:\Bharti Work\OpenGeo Suite\IG	ET_WEBGIS_002_Data\Bai	nk_BVU public	bank_bvu	geom	0	Create				
F:\Bharti Work\OpenGeo Suite\IG	ET_WEBGIS_002_Data\Bai	rren_BVI public	barren_bvu	geom	0	Create				
F:\Bharti Work\OpenGeo Suite\IG	ET_WEBGIS_002_Data\Bus	F:\Bharti Work\OpenGeo Suite\IGET_WEBGIS_002_Data\Busstops_E public busstops_bvu geom 0 Create 🗌 🔽								

19. A 'working status' window flashes on your screen and after some time you can see 'Shapefile import completed' message in the 'Log window'. Close this window after completion of import of all shape files.

Working	X
Creating spatial index	
	Cancel

20. Now, go back to the 'pgAdmin III' Window → Expand the 'Databases' under 'BVU' server in 'Object browser' → Expand the 'BVU_IEER_DB' → expand 'Schemas' → then expand 'Public' → expand 'Tables' to see the list of imported files.

<u>File Edit Plugins View T</u> ools <u>H</u> elp		
🔌 🥵 💼 💁 🗑 🙉 🗐		
bject browser	Properties Statistics Dependencies Dependents	.
BVU (localhost:54321)		
E Databases (5)	Table Owner Comment	
B-U BVU_IEER_DB	agriculture_bvu postgres	
H Catalogs (2)	atm_bvu postgres	
Extensions (1)	🔄 bank_bvu postgres	
E- Schemas (1)	🔝 barren_bvu postgres	
	🔝 busstops_bvu postgres	
Collations (0)	🔝 colony_bvu postgres	
Domains (0)	contour byu postgres	
FTS Configurations (0)	educationalinst byu postgres	
FTS Dictorialies (0)	in highway byu postores	
ETS Tomplates (0)	I holyplaces by postgres	
Functions (525)	i hospitals byu postres	
Sequences (20)	macid but postares	
Tables (21)		
i agriculture byu	i openapacijeva postaras	
at hu	in gauge und for the posterior	
Bank by		
Here barren byu	scrub_over postgres	
Busstops byu	spatial retrys postgres	
😥 🖬 colony byu	streetroads_ovu postgres	
😥 🔄 contour byu	i urban_bvu postgres	
🕀 📅 educationalinst bvu	urbangreen_ovu postgres	
🕀 📅 highway byu	a water_bvu postgres	
⊞ 💀 hospitals_bvu		
⊞ 🔝 masjid_bvu	· · · · · · · · · · · · · · · · · · ·	4
⊕- 📰 openspace_bvu		
⊕- 🔢 quary_bvu	SQL pane	×
Image: Book of the second s		
te-100 scrub_bvu		
i∰- 🔝 spatial_ref_sys		
😟 📅 streetroads_bvu		
😟 📅 urban_bvu		
🗟 🔤 urbangreen_bvu		
😟 📅 water_bvu		
Trigger Functions (2)		
Retrieving details on tables Done		0.00 secs

Spatial indexing

The main purpose of spatial indexing is to support spatial selection, that is, to retrieve from a large set of spatial objects (<u>Ralf Hartmut Güting, 1994</u>) in less time more efficiently.

By default '*Shapefile importer*' tool creates the spatial index after importing the data. See the following figure, however, now we will learn manually how to drop and create spatial indexing using SQL command.



Introduction to PostGIS



Dropping and Creating spatial index of a table

- 21. Spatial indexing speeds up the query by applying R-tree algorithm to the tables for more information see http://revenant.ca/www/postgis/workshop/indexing.html
- 22. Click on the SQL queries' tool to open 'Query' Window.
 23. Now we will drop the spatial indexing of 'agriculture_bvu' table by using <u>DROP INDEX</u> statement, type 'DROP INDEX agriculture_bvu_geom_gist;' in the text box of SQL Editor.
- 24. Click on voice the command (to run/ fire the query).
- 25. Select the table 'agriculture_bvu' under Tables in 'Object browser', click on 🗭 'Refresh Selected Object' button from Toolbar in 'pgAdmin III'. Now you can notice the index of 'agriculture_bvu' table has been removed.

🦞 pgAdmin 🎞	and setting in hand, as he through the	
<u> Eile Edit Plugins View T</u> ools <u>H</u> elp		
🎽 🛃 😕 🗑 😰 🖉 🎽	• • ?	
Dbject browser X	Properties Statistics Dependencies Dependents	
Query - BVU_IEER_DB on postgres@localhost:54321 *		
File Edit Query Favourites Macros View Help		
Ĩ : <mark> </mark>	양 ፤ □ BVU_IEER_DB on postgres@localhost:54321	
SQL Editor Graphical Query Builder (24)	₹	ratch pad X
Previous queries	Delete All	^
DROP INDEX agriculture bvu geom gist;		
(23)		
		-
	۱ (۱ ۱	Þ
Output pane		×
Data Output Explain Messages History		=
Ouerv returned successfully with no result in 33 ms.		
		-
<		Þ
ок.	Unix Ln 1, Col 38, Ch 38	33 ms



11



26. Again to create the spatial indexing of 'agriculture_bvu' table, use '<u>CREATE INDEX</u>' statement, type 'CREATE INDEX agriculture_bvu_geom_gist ON agriculture_bvu USING GIST (geom);' in SQL Editor →

Click on 💌 to execute the command.

BVU_IEER_DB on postgres@localhost:54321 *	
<u>File Edit Query Favourites Macros View H</u> elp	
👔 🗅 📂 🔚 🐰 🛅 🛱 🗢 🦘 🦚 🔎 🕨 🎎 🐜 🍓 = 💡 👔 🗆 BVU_IEER_DB on postgres@localhost:54321	
SQL Editor Graphical Query Builder	atch pad X
Previous queries Delete All	
CREATE INDEX agriculture_bvu_geom_gist ON agriculture_bvu USING GIST (geom);	
Image: Contract of the second seco	
Output pane	×
Data Output Explain Messages History	=
Query returned successfully with no result in 431 ms.	~
4	
OK. Unix Ln 1, Col 77, Ch 77	431 ms

Vacuum and Analyze

Vacuuming the database is a necessary action to perform after creation of Indexes, updates, inserts and deletes. It allows *PostgreSQL* to recover the unused space in the database to work more efficiently. PostgreSQL offers an 'autovacuum' option, it automatically takes care of recovering the space and update statistics on a sensible time intervals. To recover spaces we have to use 'VACUUM' command, and we have to use 'ANALYZE' command to update the statistics of a table after a large number of updates and deletions. It is highly recommended to run 'VACUUM' and 'ANALYZE' command manually after creating indexes and data uploads to table/ entire database.

27. Now run 'VACUUM agriculture_bvu;' command in *SQL Editor* to perform vacuuming on 'agriculture_bvu' table after creation of index.

Query - BVU_IEER_DB on postgres@localhost:54321
File Edit Query Favourites Macros View Help
🔢 🗅 📂 🔜 🐰 🗈 🖷 🧼 🦘 🏊 🔎 🕨 🔩 🖿
SQL Editor Graphical Query Builder
Previous queries
VACUUM agriculture_bvu; 27
Output pane
Data Output Explain Messages History
Query returned successfully with no result in 33 ms.

28. Similarly, to update the statistics of 'agriculture_bvu' table, run 'ANALYZE agriculture_bvu;' command in *SQL Editor* Window.

Query - BVU_IEER_DB on postgres@localhost:54321 *
File Edit Query Favourites Macros View Help
] 🗅 🚰 🔜 🌡 🖿 📽 🖉 🔿 🔿 🎾 🕨 🐜 🐜 🖷
SQL Editor Graphical Query Builder
Previous queries
ANALYZE agriculture_bvu; (28)
Output pane
Data Output Explain Messages History
Query returned successfully with no result in 72 ms.

29. We can also perform both Vacuum and Analyze commands at a time on table/database by running 'VACUUM ANALYZE agriculture_bvu;' command.

Query - BVU_IEER_DB on postgres@localhost:54321 *
File Edit Query Favourites Macros View Help
i 🖻 🚰 🔚 🌡 🗈 🖷 🧼 🦘 🏊 🔎 🕨 陆 🖷 1
SQL Editor Graphical Query Builder
Previous queries
VACUUM ANALYZE agriculture_bvu; 29
<
Output pane
Data Output Explain Messages History
Query returned successfully with no result in 42 ms.

Note: you can also run multiple commands on Query tool.



Clustering data

Clustering feature in *PostgreSQL* allows us to reorder records in the table on disk from unordered state to the order state. This ensures that records with similar attributes have a high likelihood of being found in the same page, reducing the number of pages that must be read into memory for some types of queries.

30. To cluster the data in 'agriculture_bvu' table, run 'CLUSTER agriculture_bvu USING agriculture_bvu_geom_gist;' command in SQL Editor.

Query - BVU_IEER_DB on postgres@localhost:54321 *	
File Edit Query Favourites Macros View Help	
🗄 🗅 🚰 🔚 🌡 🗈 🖷 🧼 🧒 🎮 🔎 🕨 🎼 🐂 🏣 💡 📗 D BVU	ĮEE
SQL Editor Graphical Query Builder	
Previous queries Delete	
CLUSTER agriculture_bvu USING agriculture_bvu_geom_gist;	
<	
Output pane	
Data Output Explain Messages History	
Query returned successfully with no result in 195 ms.	

PostGIS Special Functions

PostGIS has a 525 special functions, therefore we will discuss few important functions here, which we use frequently. Please refer '<u>Chapter 13: PostGIS Special Functions Index</u>', for full list of functions available in PostGIS.

We will cover following special function in this exercise to get an idea of how we can use special functions.

- ST_IsValid
- ST_IsValidReason
- ST_Centroid
- ST_contains
- ST_Length

- ST_Buffer
- ST_Intersects
- ST_Distance
- ST_AsText
- ST_Intersection

Validation

Before performing any operations on database, it is very important to validate the data in the database to avoid future conflicts. This validation can be by visual inspection or using special function. No one method alone ensures the data is free from errors, this means some errors still exist after visual inspection and vice versa. It is always recommend practicing the both methods will yield good results. In this tutorial we will use the special function called '<u>ST IsValid</u>' to validate the geometry of data present in the 'urban_bvu' table.

31. The following query shows the validity of polygon having 'gid 39' in 'urban_bvu' table. runthe following command in 'SQL Editor'

SELECT ST_IsValid ((select geom from urban_bvu where gid=39));

32. The result will be displayed in 'Output pane' showing 't', it stands for Boolean true, this means the polygon with 'gid 39' has correct geometry.



	₿ Q	uery	- BVU_	IEER	_DB or	n post	tgres@	local	host:5	4321 *	•	-					-
	File	Edit	Que	ery	Favou	rites	Mac	ros	View	Help							
	: 1	B		X	d	\bigcirc	5	2	$\left \right\rangle$		₽ GS D			8		0 E	BVU_IE
	SQ	L Edi	tor	Grapł	hical Qu	ery Bu	uilder									Ŧ	Scrat
	Previ	ous qu	ueries (C	elete		Delet	e All		
31		ST	IsVal	lid	((sel	ect	geom	fro	n urb	an_bv	u w	here	gid	=39))	;		
Ŭ																	
	•																•
	Outpu	ut pan	e														
22	Da	ta Oi	utput	Ex	plain	Mes	sages	Hi	story								
32			st_isv boolea	alid an													
	1	. 1	t														
					1												

33. It is really a tough job to check the validity for each and every feature individually, if we are dealing with huge data. In order to find all *invalid* polygons in the 'urban_bvu' table, along with the gid of the corresponding polygon run the following query. You can see we used 'NOT' keyword before 'ST_IsValid' function to find invalid geometries.

SELECT gid FROM urban_bvu WHERE NOT ST_IsValid(geom);

	SQL Ed	itor G	ra	phical Qu	ery Builder					
33	Previous o	queries								
	SELECT gid FROM urban_bvu WHERE NOT ST_IsValid(geom);									
	Output pa	ne								
	Data 0	utput	E	xplain	Messages	Hi				
		gid integer								
	1	45								
	2	46								
	3	47								
	4	49								

34. We don't know the reason for invalidity of above polygons, in order to find the reason for invalidity use '<u>ST IsValidReason'</u> function. Run the following query in SQL Editor.

SELECT gid, ST_IsValidReason(geom)



FROM urban_bvu

WHERE NOT ST_IsValid(geom);

35. The result is shown below; it says self intersection is the reason for invalid of the polygons.

	SQ	L Editor G	raphical Que	ery Builder			-	; S
	Previo	ous queries				- Dele	te Delete All	
34		SELECT g FROM urb WHERE NO	id, ST_I an_bvu T ST_IsV	sValidReas alid(geom)	on (geom) ;			
	•						4	
	Outpu	it pane						
35	Da	ta Output	Explain	Messages	History			
Ŭ		gid integer	st_isvali text	dreason				
	1	45	Self-in	tersection	[379816.	933398036	2039923.785439	72]
	2	46	Self-in	tersection	[379135.	108040535	2039501.278930	43]
	3	47	Self-in	tersection	[376789.]	28353432	2040214.9590423	1]
	4	49	Self_in	terrection	1270096	026497166	2041212 271744	161

Rectifying the invalid geometry using ST_Buffer

Buffering is very common proximity analysis tool in Geographic Information Science. You can find a detailed explanation to buffer analysis in the article <u>Buffer Analysis in GIS</u>, written by Nagapramod. PostGIS allows users to perform buffer analysis through '<u>ST Buffer</u>', using this function you can buffer a geometry in 2 ways

- Positive buffer: A buffer extending outwards/ away from geometry
- Negative buffer: A buffer that extends inside the geometry (this type of buffer don't work on point and line and in case where the negative buffer is greater than the size of polygon)
- 36. The following query will create a new table named '**BVUHospital_Buffer**', it contains a circular polygon area falling around 750 metres of *Bharati Vidyapeet Hospital* in 'Hospital_BVU' table. This buffer zone contains approximately 1.5 km long Pune- Satara highway, unfortunately if any accident occurs on this road, the victim may be reach this hospital in 2-3 mintues.

CREATE TABLE bvuhospital_buffer AS SELECT ST_Buffer(geom,750) AS geom FROM hospitals_bvu Where gid = 21;

Note: To see the 'bvuhospital_buffer' table under the list of tables, select Tables in the object browser and click on \Im 'Refresh Selected Object' button from Toolbar in 'pgAdmin III'.

	SQL Editor Graphical Query Builder =	So
	Previous queries CREATE TABLE bvuhospital_buffer AS : Delete Delete All	
36	CREATE TABLE bvuhospital_buffer AS SELECT ST_Buffer(geom, 750) AS geom FROM hospitals bvu Where gid = 21;	
	Output pane	
	Data Output Explain Messages History	
	Query returned successfully: 1 row affected, 135 ms execution time	me .

37. The result of above buffer zone can be visualized in QGIS, via Layer \rightarrow Add PostGIS Layers \rightarrow establish the connection as per the credentials in *Step 5 & 8* in '*Add PostGis Layer*' window \rightarrow click on '*Connect*' \rightarrow select '*bvuhospital_buffer*' table \rightarrow click on '*Add*' \rightarrow select 'WGS84 / UTM zone 43N' as CRS \rightarrow click '*OK*'



38. We can also use *ST_Buffer* function keeping the buffer distance as zero meters to rectify the self intersection error in the geometry. Run the following command to create a valid urban table, i.e., urban_valid_bvu from 'urban_bvu' table.

CREATE TABLE urban_valid_bvu AS

SELECT ST_Buffer(geom,0) AS geom ,gid

FROM urban_bvu;

	SQL Editor Graphical Query Builder
38 urban byu	Previous queries Delete Delete
	CREATE TABLE urban_valid_bvu AS SELECT ST_Buffer(geom,0) AS geom ,gid FROM urban_bvu;
	Output pane
iVectorData	Data Output Explain Messages History
'ver rd	Query returned successfully: 49 rows affected, 493 ms execution time.

39. By running the following query in the SQL Editor, we can see the manipulation of urban polygons with zero buffer distance worked or not?

SELECT gid,

ST_IsValidReason(geom)

FROM urban_valid_bvu

WHERE NOT ST_IsValid(geom);

	SQL Editor Graphical Query Builder	
30	Previous queries	
	SELECT gid, ST_IsValidReason(geom) FROM urban_valid_bvu WHERE NOT ST_IsValid(geom);	
		_
	Data Output Explain Messages History	
40	gid st_isvalidreason integer text	

40. Yes! From the blank results we can conclude that is there is no invalid geometries exist in the 'urban_valid_bvu' table.

ST_ Centroid: This function will return a point geometry that is placed at the centre of a given geometry.

For example, Pune Municipal Corporation (PMC), want to erect a 'Chatrapati Shivaji Monument and a fountain' in the center of 'Upper Katraj lake' to attract people to the park, caould you help them by supply the coordinates of center of lake?

41. Yes, you can use '<u>ST Centroid</u>' function to compute the center of *Upper Katraj Lake*. Run the following query in *SQL Editor*.

select ST_AsText(ST_Centroid

((select geom from water_bvu where gid=2)));

42. In above query we are requesting the center of lake as text by using '<u>ST_AsText</u>' function. This will yield result with Easting and Northing values of center of lake in *Data Output* pane.

	SQL Ed	itor Graphical Query Builder
(41)	Previous	gueries SELECT "Name" as BusStop_Name FRO 🔻 Delete Delete
	□ se L ((<pre>lect ST_AsText(ST_Centroid select geom from water_bvu where gid=2))); </pre>
	Output pa	ne
	Data C	utput Explain Messages History
(42)		st_astext text
$\overline{}$	1	POINT (379786.277236707 2040786.31334766)



ST_Contains : It checks for and returns the geometries that are completely inside of another geometry. For example, PMC wants to modernize the bus stops in ward ID number 75. For this purpose they need the name of bus stops exist in ward 75, how you can help PMC?

43. You can '<u>ST_Contains</u>' function to answer above query. Run the following query to get the results.

SELECT bs_name as BusStop_Name

FROM wards_bvu, busstops_bvu

WHERE ST_Contains (wards_bvu.geom,busstops_bvu.geom)

AND wards_bvu.id =75;

44. The above query will result a table of 25 rows in output pane showing the bus stop falling under ward ID:75



ST_Intersects: This function yields a Boolean output, it return true, when the geometries overlap, touches or intersects each other and false in case of disjoint.

For example: Roads are often fragments the reserved forest areas, this is a barrier for wild life passages. In India road kill of wild life is very common, in order to prevent this situation, we need to take useful measures along the roads passing through the forests. We can use '<u>ST Intersects</u>' function to identify the roads passing through the forests.

45. The following code shows, highway road segment, which are passing through the reserved forest area of south Pune toposheet.

SELECT name as HighwayName

FROM highway_bvu as hw, reservedforest_bvu as rf



			Scrate
	SQL E	ditor Graphical Query Builder 🗢	
	Previous	queries SELECT "Name" as BusStop_Name FRO ▼ Delete Delete All	
45	SI FI W	ELECT name as HighwayName ROM highway_bvu as hw, reservedforest_bvu as rf HERE ST_Intersects (hw.geom,rf.geom);	
			•
	Output pa	ane	
	Data	Output Explain Messages History	
		highwayname character varying(80)	
	1	NH4	

WHERE ST_Intersects (hw.geom,rf.geom);

- 46. The result of query in *step 45*, showed that '*NH4*' highway is passing through the reserved forest area, now we are interested to know the length of the segment of NH4, which is passing through the reserved forest, it helps us to estimate the cost of a project, which is aimed to create a fence along the road and optimal underground wildlife passages. For this purpose we will use '<u>ST Length</u>' function along with '<u>ST Intersection</u>'.
- 47. Run the following query in *SQL Editor* to get the length of NH4 segment passing through the reserved forest area.

	SELECT SUM (ST_Length (ST_Intersection (hw.geom, rf.geom))) as total_length_of_highway_in_RF FROM highway_bvu as hw ,reservedforest_bvu as rf;							
	SQL Editor	Graphical Query Builder			₹			
47	Previous queries	SELECT "Name" as BusStop	Name FRO 🔻	Delete	Delete All			
Ŭ	SELECT SUM (ST_Length (ST_Intersection (hw.geom, rf.geom as total_length_of_highway_in_RF FROM highway_bvu as hw ,reservedforest_bvu as rf;							
l	<				Þ			
	Output pane							
	Data Output	t Explain Messages	History					
	tota dout	_length_of_highway_in_ le precision	ŗf					

48. Measuring area is a very common task in GIS. '<u>ST Area</u>' function is used to measure area in PostGIS. Following query will yield the areas of five biggest wards of PMC in descending order.

SELECT gid, ST_Area (geom)/10000 as "Area in Hectares" FROM wards_bvu ORDER BY "Area in Hectares" DESC

392.8769676836



1

LIMIt 5;



49. Often it is very useful to measure the distance between starting and destination stations. For such tasks we will use '<u>ST_Distance</u>'. This spatial function will give cartiesian minimum distance between two geometries. Use the following command to find distance between bus stops having gid 1& 15 in project units, in our case meters.

SELECT ST_Distance ((SELECT geom from busstops_bvu WHERE gid=1), (SELECT geom from busstops_bvu WHERE gid=15));

49
\sim

JULI		arapıncai Qu	ier y builder				
Previous	queries				•	Delete	Delete All
	SELECT S ((SELECT) (SELECT)	ST_Distan I geom fr geom fro	nce rom busstop om busstops	s_bvu WH	IERE RE	gid=1), gid=15));	
Output p	III Dane						
Data	Output	Explain	Messages	History			
-		-					
	st_dist double	tance precision	1				