

TSI® FMS 5 SOFTWARE

HOW TO CONFIGURE SEVERAL MONITORS ON ONE COMPUTER WRITING TO DIFFERENT DATABASES

TECHNICAL BULLETIN TCC-107 (US)

Contents

Prerequisites	2
Assumptions	2
PostgreSQL 9.3.5 Configuration Instructions	2
1. Configuring pga_conf	2
2. Create Database “fms02”	4
3. Set Privileges on “fms02” Database.....	7
4. Create Schema “fms02”	9
5. Create Tables in Schema “fms02”	11
6. FMS Configuration Instructions	19



Prerequisites

To run this procedure FMS 5.0.7, including all latest Software Updates, must be installed along with 32bit PostgreSQL Version 9.3.5 or above.

Assumptions

This procedure explains how to set more than one Node on the same Monitor.

We will assume running two Nodes called:

- FMS_DEMO
- FMS_DEMO_1

Both Nodes will then store data in different PostgreSQL Databases that we will call:

- "fms01" for NODE_DEMO
- "fms02" for NODE_DEMO_1

We assume that PostgreSQL has been installed to the default installation folder which is:

C:\PROGRAM FILES (X86)\POSTGRESQL\9.3\DATA

If PostgreSQL has been installed in a different folder please replace, in this procedure, the path shown above with the one you have chosen during PostgreSQL installation.

PostgreSQL 9.3.5 Configuration Instructions

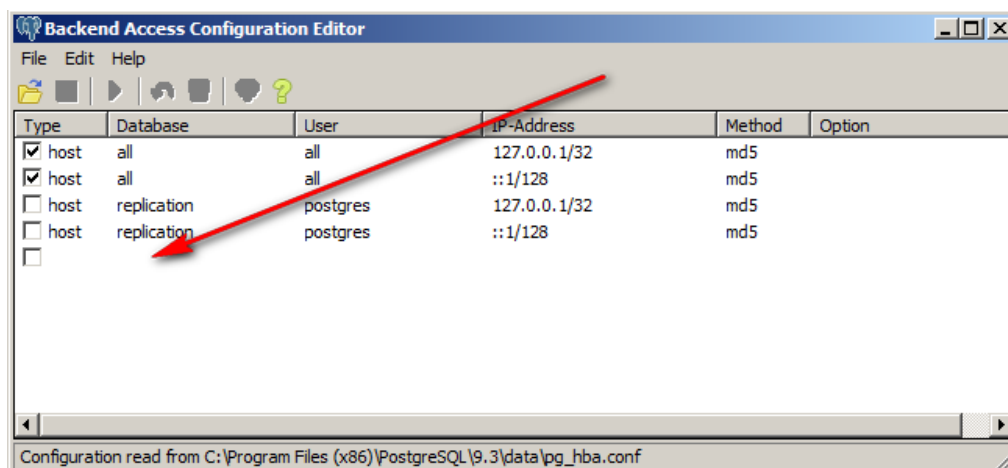
1. Configuring pga_conf

1.1 Start pgAdmin III.

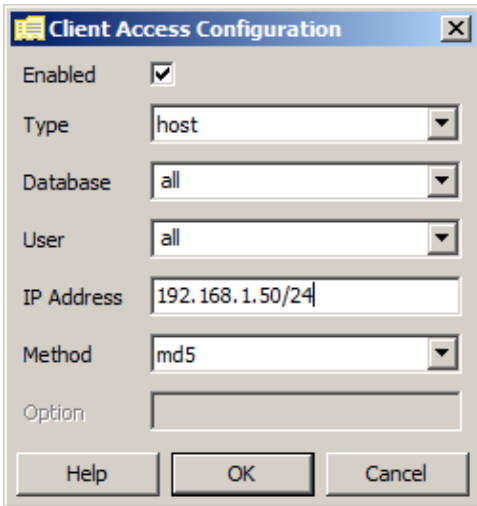
1.2 Add the local IP address to the file pga_hba.conf by starting:

- Select in Menu File – Open pga_hba.conf
- Browse for this file within the Postgres 9.3 installation folder (C:\PROGRAM FILES (X86)\POSTGRESQL\9.3\DATA).

1.3 On the following screen double click on the last line as shown below.

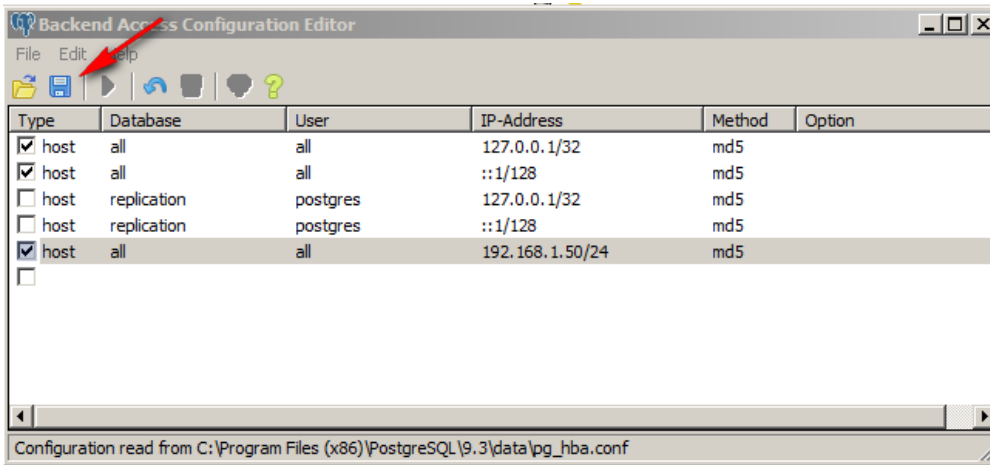


1.4 Enter the following values:

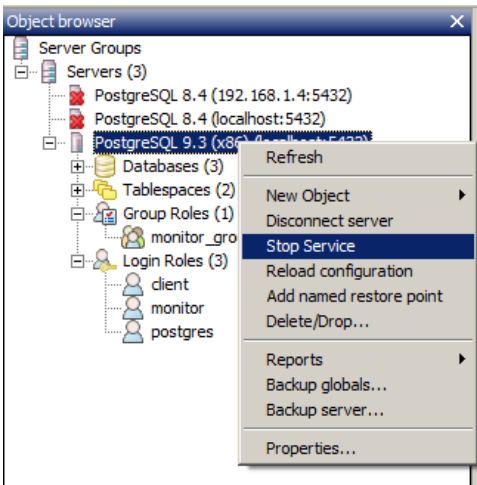


Note: The IP address is your local Computer IP Address.

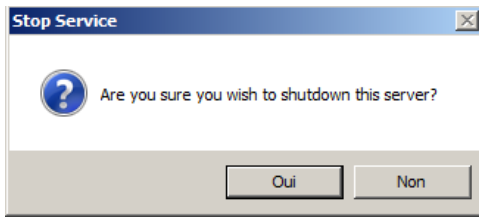
1.5 Click on the **Save** icon and then exit.



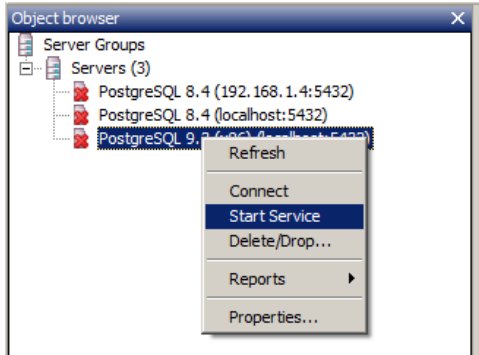
1.6 Restart PostgreSQL 9.3 Service by right-clicking on “PostgreSQL 9.3” Server and select “Stop”.



1.7 Click on **“Yes”** to confirm you want to stop the service.



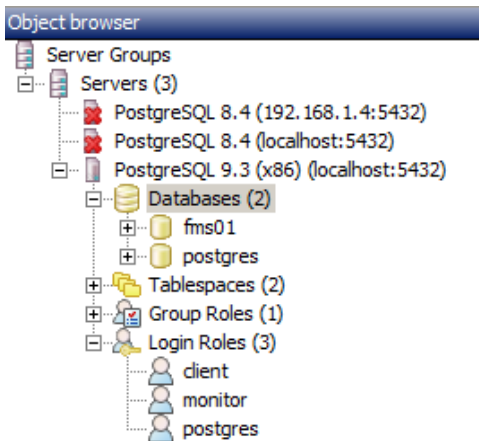
1.8 Right-click on **“PostgreSQL 9.3”** Server and select **“Start Service”**.



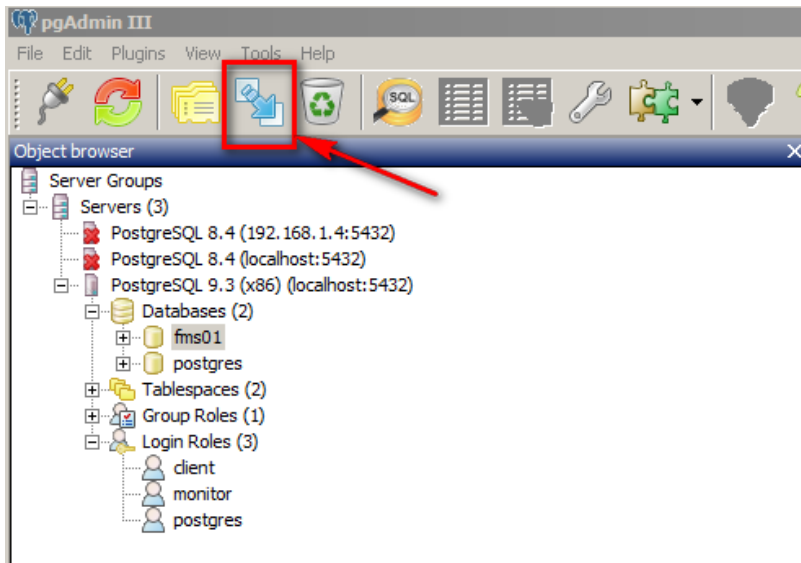
2. Create Database **“fms02”**

2.1 Start pgAdmin III.

2.2 Expand **“PostgreSQL 9.3”** then expand **“Databases (2)”**.

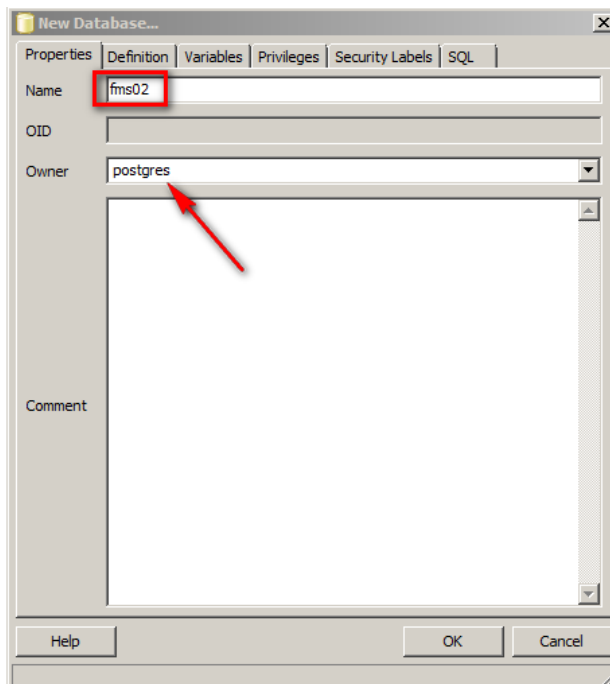


2.3 Click on “fms01” database then click on the following icon:



2.4 Enter the following values:

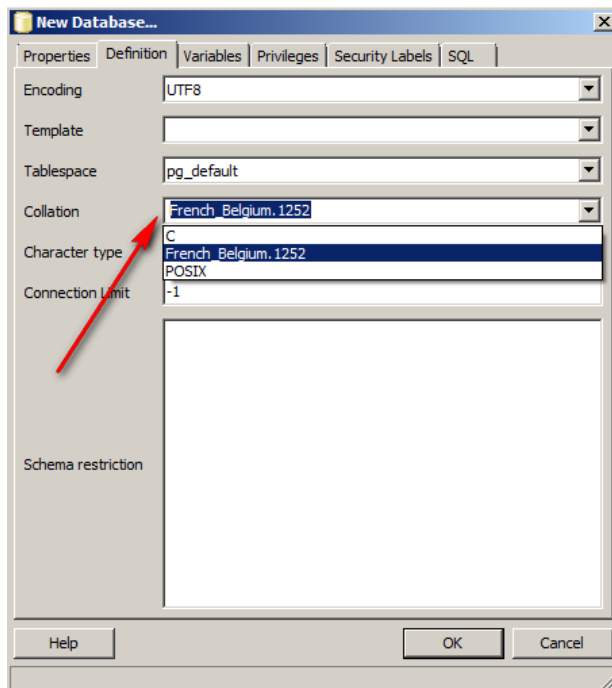
- Name= “fms02”
- Owner= “postgres”



Then click on the “**Definition**” tab.

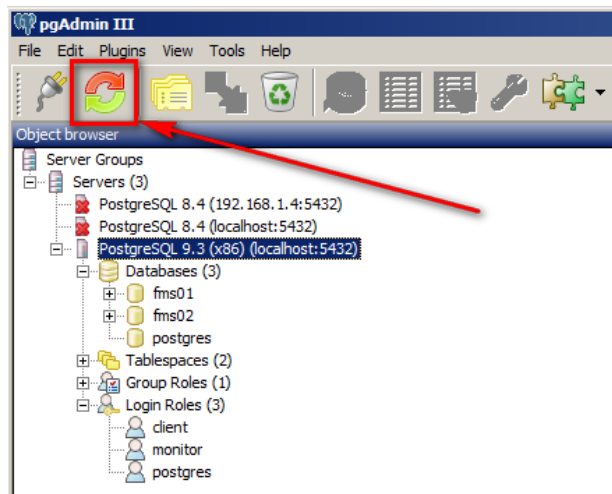
2.5 From the drop-down list “Tablespace” select “**pg_default**”.

From the drop-down list “Collation” and “Character Type” select your language code page. This value depends on the Windows® Operating System language.



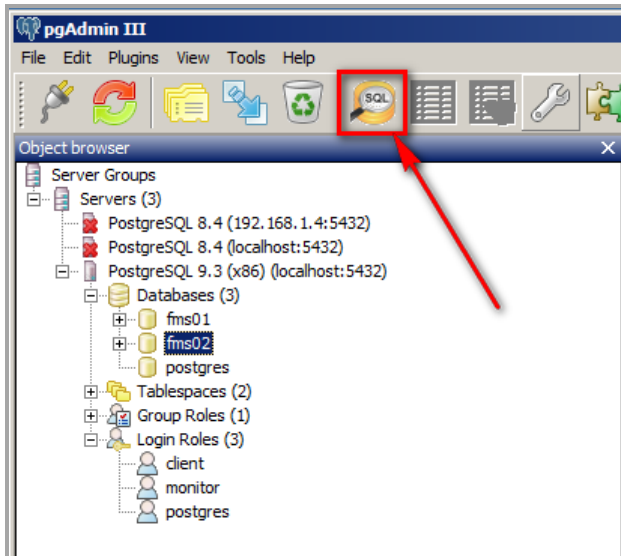
2.6 Click “**OK**”.

2.7 Click on “**PostgreSQL 9.3**” then click on the icon shown below. This will refresh your screen and will show the newly created database “fms02”.

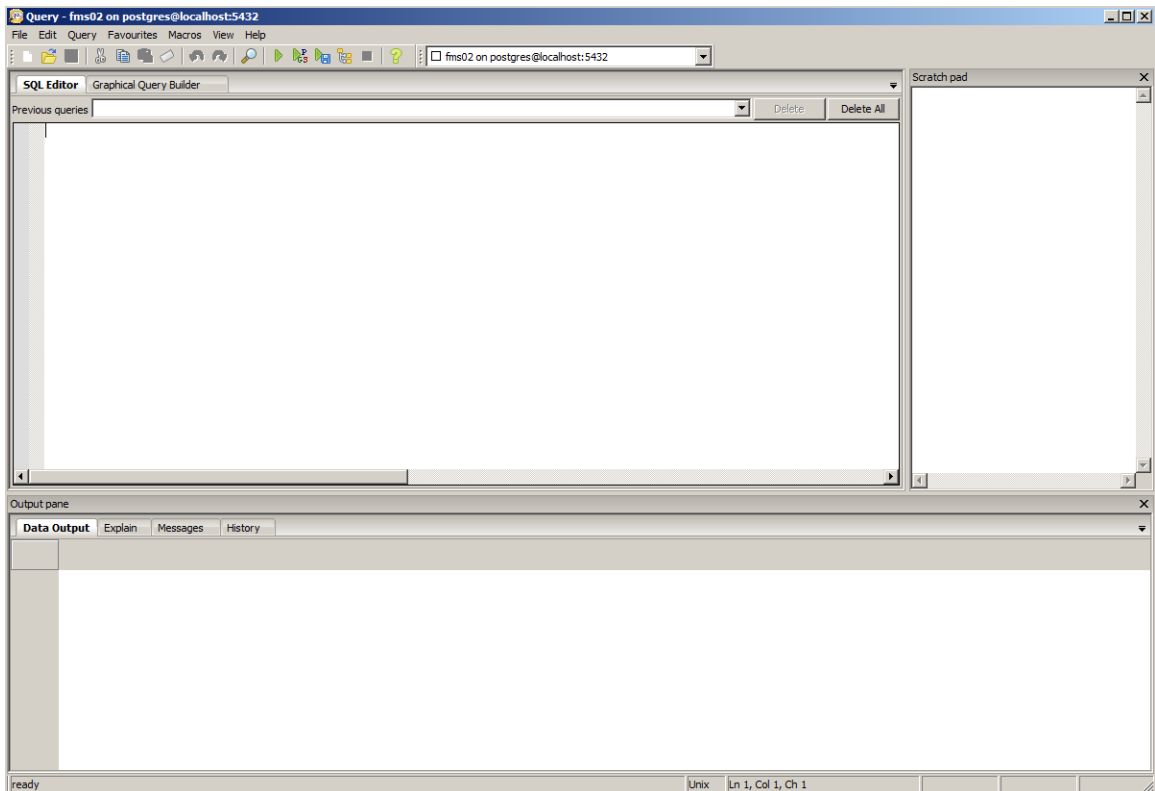


3. Set Privileges on “fms02” Database

3.1 Click on “fms02” then click the following icon:



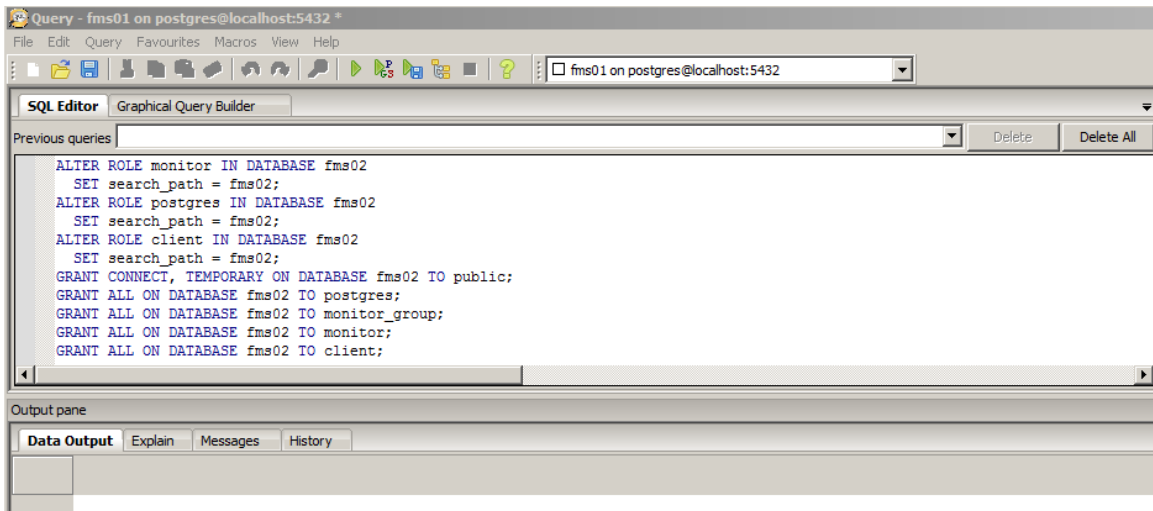
3.2 A new window will show up.



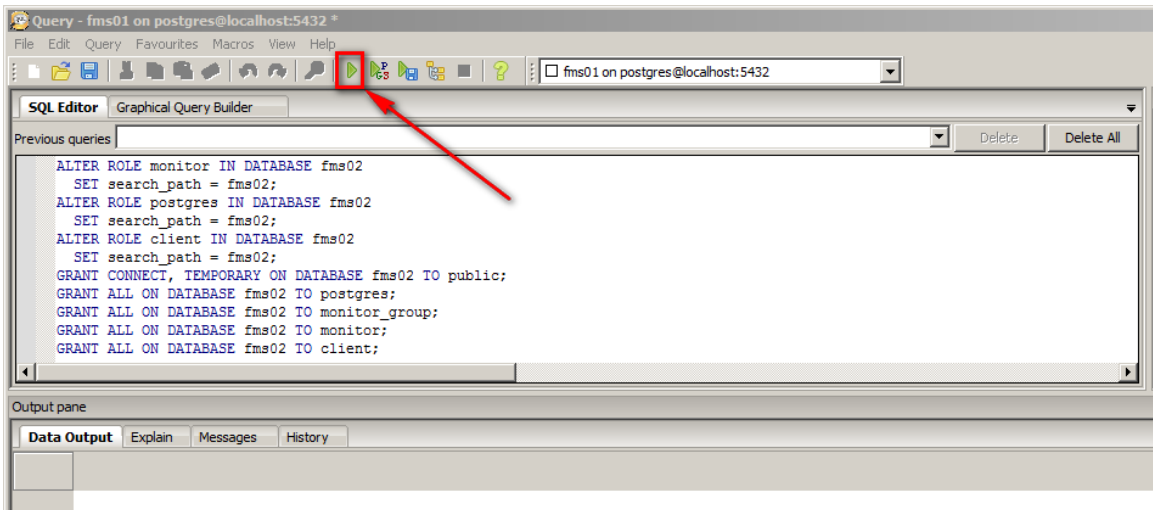
3.3 Copy and paste the following code into this window:

```
ALTER ROLE monitor IN DATABASE fms02
SET search_path = fms02;
ALTER ROLE postgres IN DATABASE fms02
SET search_path = fms02;
ALTER ROLE client IN DATABASE fms02
SET search_path = fms02;
GRANT CONNECT, TEMPORARY ON DATABASE fms02 TO public;
GRANT ALL ON DATABASE fms02 TO postgres;
GRANT ALL ON DATABASE fms02 TO monitor_group;
GRANT ALL ON DATABASE fms02 TO monitor;
GRANT ALL ON DATABASE fms02 TO client;
```

Note: Be careful to copy exactly the code from A to ;



3.4 Click on the following icon to execute the Query:

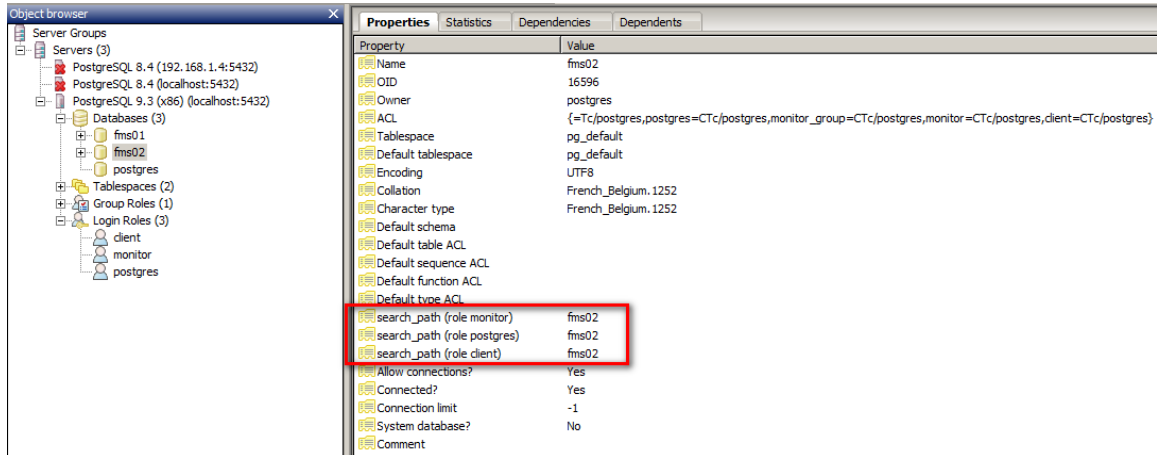


3.5 Close the Query window without saving.

3.6 You **MUST** get a result successful message in the Output pane; otherwise, it means that you did not correctly paste the code in step 3.3.

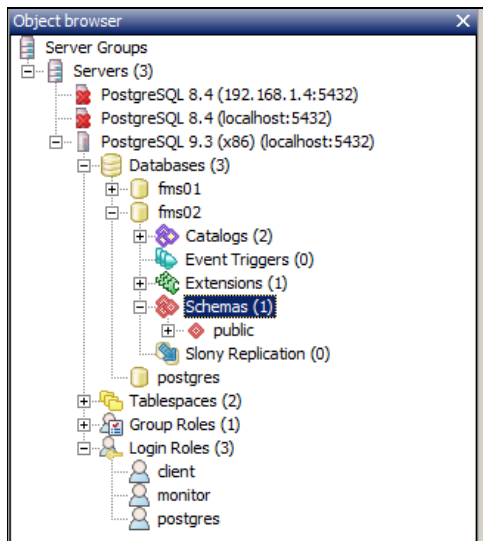
3.7 Click on database “fms02”, followed by clicking on the **Refresh**  icon.


Verify that the “search_path” is set correctly as shown below.



4. Create Schema “fms02”

4.1 Expand Database “fms01” then expand “Schemas (1)”. You should only see a “public” schema.



4.2 Click on the  icon while you are on “Schema (1)”.

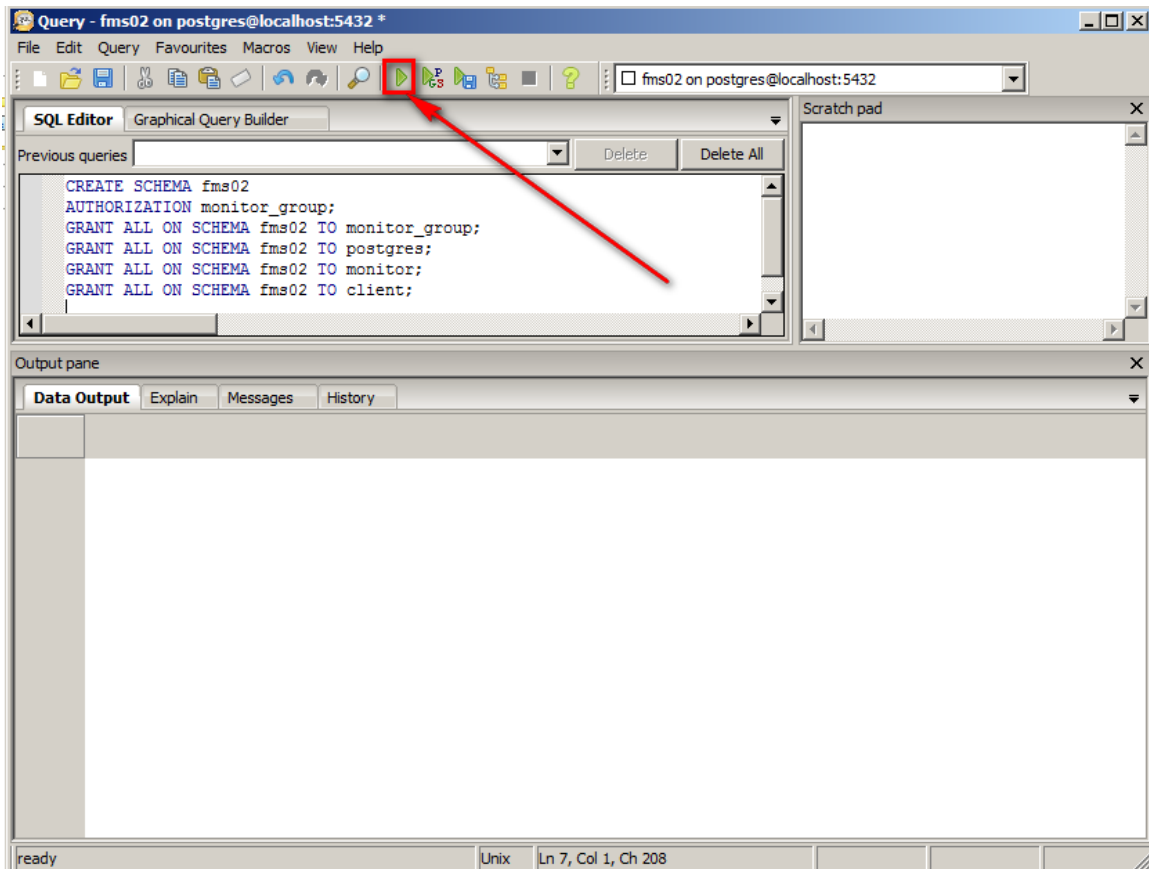
4.3 Copy and paste the following code into the Query window:

```
CREATE SCHEMA fms02
AUTHORIZATION monitor_group;
GRANT ALL ON SCHEMA fms02 TO monitor_group;
GRANT ALL ON SCHEMA fms02 TO postgres;
GRANT ALL ON SCHEMA fms02 TO monitor;
```

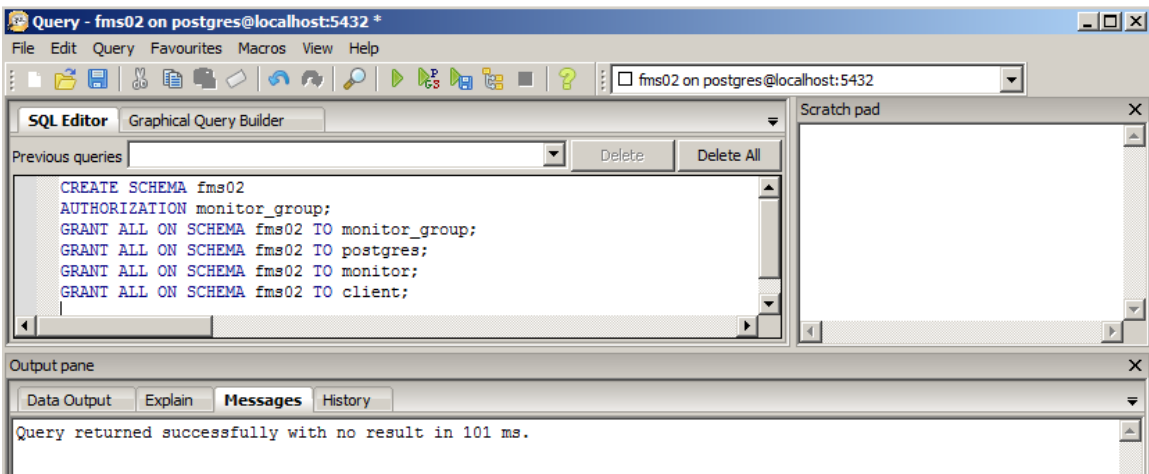
4.4 GRANT ALL ON SCHEMA fms02 TO client;

Note: Be careful to copy exactly the code from C to ;

4.5 Click on the “Execute Query” icon.



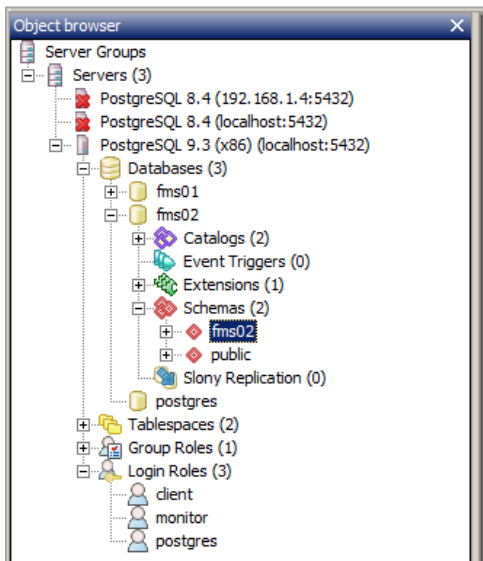
4.6 You **MUST** get a result successful message in the Output pane; otherwise, it means that you did not correctly paste the code in step 4.3.




4.7 Close the Query window without saving.

4.8 Click on schema “fms02”, followed by clicking on the **Refresh**  icon.

4.9 You should now see the newly created schema “fms02”.



5. Create Tables in Schema “fms02”

5.1 While you are on “fms02” schema click on the **SQL**  icon.

5.2 Copy and paste the following code into the Query window :



CreateCommonTables.sql

Note: Be careful to copy exactly the code from -- to ; (*written in blue*)

5.3 You may also edit the file shown above “CreateCommonTables.sql” and copy all the code:

```
-- DROP TABLE fms02.alarm_list;
CREATE TABLE fms02.alarm_list
(
    timedate timestamp without time zone,
    node character varying(64),
    source character varying(64),
    messages character varying(255)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE fms02.alarm_list OWNER TO monitor;

-- Index: fms02.alarm_list_index

-- DROP INDEX fms02.alarm_list_index;

CREATE INDEX alarm_list_index
ON fms02.alarm_list
USING btree
(node, source);

CREATE TABLE fms02.alarm_log
(
    timedate timestamp without time zone,
```

```

    source character varying(32),
    messages character varying(255)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE fms02.alarm_log OWNER TO monitor;

-- Index: fms02.alarm_log_index

-- DROP INDEX fms02.alarm_log_index;

CREATE INDEX alarm_log_index
    ON fms02.alarm_log
    USING btree
    (timedate);

CREATE TABLE fms02.annotations
(
    annotation_id character varying(64),
    timeentry timestamp without time zone,
    timedate timestamp without time zone,
    sample_point character varying(32),
    tagname character varying(32),
    "comment" character varying(255),
    source character varying(32),
    publicview character varying(1)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE fms02.annotations OWNER TO monitor;

-- Index: fms02.annotation_index

-- DROP INDEX fms02.annotation_index;

CREATE INDEX annotation_index
    ON fms02.annotations
    USING btree
    (annotation_id);

CREATE TABLE fms02.audit_log
(
    timedate timestamp without time zone,
    source character varying(32),
    messages character varying(255)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE fms02.audit_log OWNER TO monitor;

-- Index: fms02.audit_log_index

-- DROP INDEX fms02.audit_log_index;

```

```

CREATE INDEX audit_log_index
  ON fms02.audit_log
  USING btree
  (timedate);

CREATE TABLE fms02.batch_events
(
  timedate timestamp without time zone,
  batch_id character varying(64),
  timeevent timestamp without time zone,
  source character varying(32),
  eventname character varying(255),
  eventcomment character varying(255)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE fms02.batch_events OWNER TO monitor;

-- Index: fms02.batch_events_index

-- DROP INDEX fms02.batch_events_index;

CREATE INDEX batch_events_index
  ON fms02.batch_events
  USING btree
  (batch_id);

CREATE TABLE fms02.batch_names
(
  timedate timestamp without time zone,
  batch_name character varying(64),
  source character varying(32)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE fms02.batch_names OWNER TO monitor;

-- Index: fms02.batch_names_index

-- DROP INDEX fms02.batch_names_index;

CREATE INDEX batch_names_index
  ON fms02.batch_names
  USING btree
  (batch_name);

CREATE TABLE fms02.event_log
(
  timedate timestamp without time zone,
  source character varying(32),
  messages character varying(255)
)
WITH (
  OIDS=FALSE

```

```

);
ALTER TABLE fms02.event_log OWNER TO monitor;

-- Index: fms02.event_log_index

-- DROP INDEX fms02.event_log_index;

CREATE INDEX event_log_index
  ON fms02.event_log
  USING btree
  (timedate);

CREATE TABLE fms02.recipe_log
(
  timedate timestamp without time zone,
  recipetrigger character varying(64),
  target character varying(64),
  targettype character varying(64),
  oldrecipe character varying(64),
  newrecipe character varying(64),
  source character varying(64),
  "comment" character varying(255)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE fms02.recipe_log OWNER TO monitor;

-- Index: fms02.recipe_log_index

-- DROP INDEX fms02.recipe_log_index;

CREATE INDEX recipe_log_index
  ON fms02.recipe_log
  USING btree
  (timedate);

CREATE TABLE fms02.spc
(
  samplepoint character varying(64),
  tagname character varying(32),
  timedate timestamp without time zone,
  state smallint,
  meanv double precision,
  maxv double precision,
  minv double precision,
  rangev double precision,
  sdv double precision,
  currentv double precision,
  sumv double precision,
  fdiff double precision,
  nv smallint
)
WITH (
  OIDS=FALSE
);

```

```

ALTER TABLE fms02.spc OWNER TO monitor;

-- Index: fms02.spc_index

-- DROP INDEX fms02.spc_index;

CREATE INDEX spc_index
  ON fms02.spc
  USING btree
  (samplepoint, tagname, timedate);

CREATE TABLE fms02.study_details
(
  timedate timestamp without time zone,
  study_id character varying(64),
  source character varying(32),
  descr character varying(255),
  notes character varying(255),
  alarmgroup character varying(64),
  director character varying(32),
  active character varying(1)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE fms02.study_details OWNER TO monitor;

-- Index: fms02.study_details_index

-- DROP INDEX fms02.study_details_index;

CREATE INDEX study_details_index
  ON fms02.study_details
  USING btree
  (study_id);

CREATE TABLE fms02.study_events
(
  timedate timestamp without time zone,
  study_id character varying(64),
  timeevent timestamp without time zone,
  source character varying(32),
  eventname character varying(255),
  eventcomment character varying(255)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE fms02.study_events OWNER TO monitor;

-- Index: fms02.study_events_index

-- DROP INDEX fms02.study_events_index;

CREATE INDEX study_events_index
  ON fms02.study_events

```

```

        USING btree
        (study_id);

CREATE TABLE fms02.study_names
(
    timestamptimestamp without time zone,
    study_name character varying(64),
    source character varying(32),
    active character(1) DEFAULT '1'::bpchar
)
WITH (
    OIDS=FALSE
);
ALTER TABLE fms02.study_names OWNER TO monitor;

-- Index: fms02.study_names_index

-- DROP INDEX fms02.study_names_index;

CREATE INDEX study_names_index
ON fms02.study_names
USING btree
(study_name);

CREATE TABLE fms02.study_users
(
    timestamptimestamp without time zone,
    study_id character varying(64),
    source character varying(32),
    studyuser character varying(255),
    "role" integer
)
WITH (
    OIDS=FALSE
);
ALTER TABLE fms02.study_users OWNER TO monitor;

-- Index: fms02.study_users_index

-- DROP INDEX fms02.study_users_index;

CREATE INDEX study_users_index
ON fms02.study_users
USING btree
(study_id);

CREATE TABLE fms02.trace_log
(
    timestamptimestamp without time zone,
    source character varying(32),
    messages character varying(255)
)
WITH (

```



```

        OIDS=FALSE
    );
ALTER TABLE fms02.trace_log OWNER TO monitor;

-- Index: fms02.trace_log_index

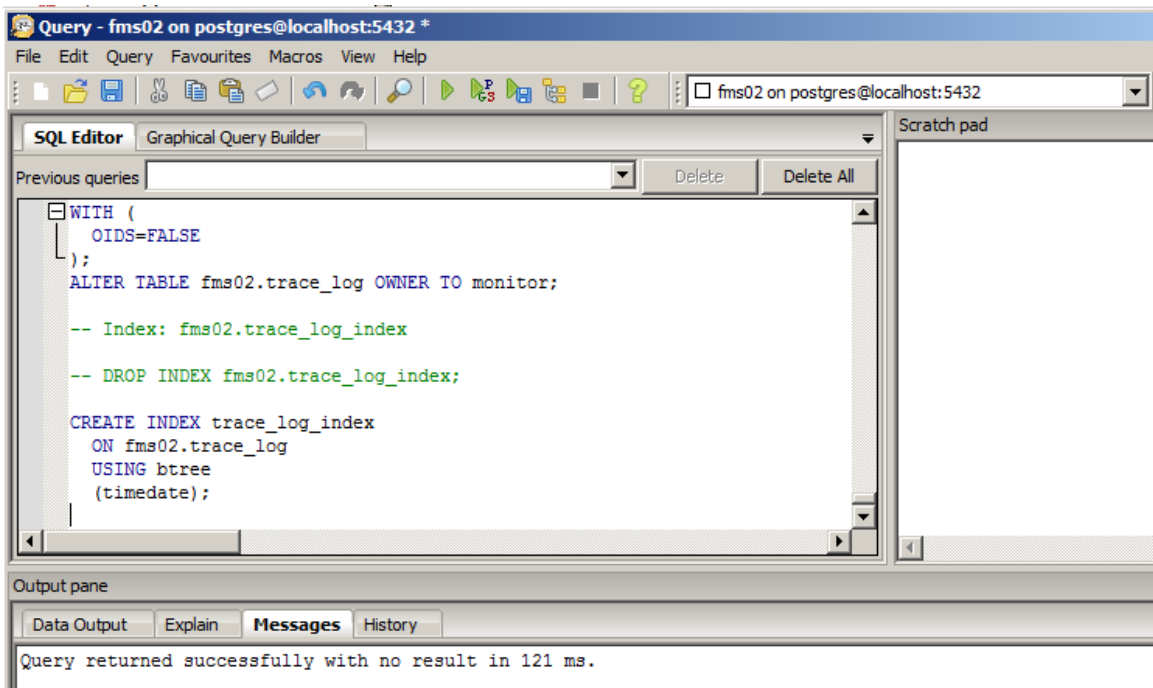
-- DROP INDEX fms02.trace_log_index;

CREATE INDEX trace_log_index
    ON fms02.trace_log
    USING btree
    (timedate);

```

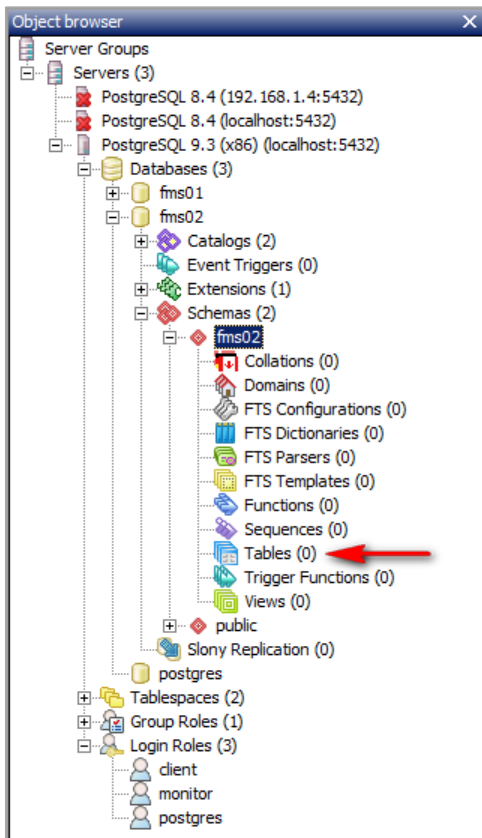
5.4 Run the Query by clicking the “**Execute Query**” icon.


5.5 You **MUST** get a result successful message in the Output pane, otherwise it means that you did not correctly paste the code in step 5.2.



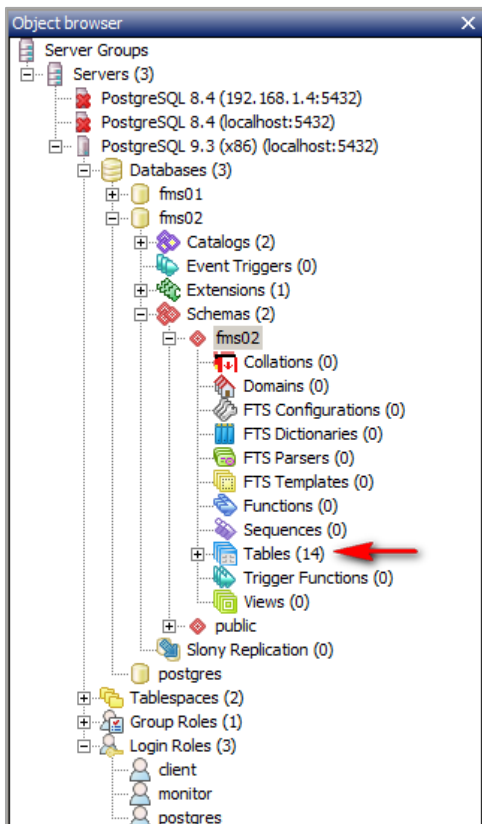
5.6 Close the Query window without saving.

5.7 Expand Schema “fms02”.



5.8 While you are on Schema “fms02” click the **Refresh**  icon.

5.9 You should now be able to see all newly created tables.



6. FMS Configuration Instructions

6.1 Edit the GUARD.INI file located in C:\FMS5\BIN

- Copy the entire first line and append it to the file
- Change the request port
- Change the LocalFile to required name
- Change the PasswordFile to required name

In our case:

```
C:\FMS5\bin\Monitor -node=FMS_DEMO request=4003 -  
localfile=local.xml -passwordfile=password.xml  
C:\FMS5\bin\Monitor -node=FMS_DEMO_1 request=4004 -  
localfile=local1.xml -passwordfile=password1.xml
```

See associated File.

6.2 Save your changes.

6.3 In C:\FMS5\Config copy the NodeLocal.xml to NodeLocal1.xml.

6.4 Edit NodeLocal1.xml file:

- Change License Number.
- Change Requested Port.
- Change DbName to "fms02"

In our case:

```
<TestLocal>  
  <BaseConfiguration Name="Client" >  
    <Item Name="MonitorModules"  
>Generic,PhoenixContact,Calculated,OutputControl,AsciiOutput,Actions</I  
tem>  
    <Item EnableBuddy2="0" EnableBuddy3="0" Buddy1Command="" ExitBuddy=""  
Buddy2Command="" Buddy3Command="" Name="Monitors" Buddy1=""  
EnableExitBuddy="0" Buddy2="" TakeOverTime="1" Buddy3=""  
EnableBuddy1="0" />  
    <Item ClientName="" Reverse="0" Description="" Name="ID"  
LicenseId="LICENSE NUMBER" Language="English" />  
    <Item RebootAt="0" AgePasswords="0" Win32Logins="0"  
RebootTime="00:00:00" AgePasswordDays="0" Name="Security"  
Win32Domain="" />  
    <Item ClientPassword="fms" PasswordServer="0" RequestPort="4004"  
AutoPasswords="0" EnableArchiving="0" Host="localhost"  
InhibitAlarmList="0" Password="fms" DbName="fms02" Type="QPSQL"  
User="monitor" ClientUser="client" Name="MonitorSetup"  
BroadcastPort="4001" NetworkInterface="Intel(R) Ethernet Connection  
I218-LM" ArchiveDirectory="" Port="5432" />  
    <Item EnableArchiving="1" Name="Archive"  
ArchiveDirectory="C:\FMS5\Archive" />  
    <Item UIDPrefix="" ReportTemplateLocation="" UseUID="0"  
UseTemplateLocation="0" Name="Reporting" />  
  </BaseConfiguration>  
</TestLocal>
```

See associated File.

6.5 Start the GUARD Service.

6.6 Start FMS Client.

6.7 Both Nodes are present in FMS.

The screenshot displays the 'FMS Client Client' application window. The interface includes a menu bar (Client, Node, Windows, Help), a toolbar with various icons, and a main workspace divided into several sections:

- Configured:** Shows two tabs labeled 'FMS_DEMO' and 'FMS_DEMO1' above an empty table.
- Messages:** A table with the following data:

Node	Date/Time	Source	Type	Message
Local	14-01-2015 22:16:18	Local	Ok	Monitor Has Connected FMS_DEMO1 from 192.168.1.50
Local	14-01-2015 22:16:18	Local	Ok	Monitor Has Connected FMS_DEMO from 192.168.1.50
Local	14-01-2015 22:16:15	Local	Ok	Monitor Has Connected FMS_DEMO1 from 192.168.1.50
Local	14-01-2015 22:16:15	Local	Ok	Monitor Has Connected FMS_DEMO from 192.168.1.50
- Units Status:** A large empty white box.
- Statistics:** Includes tabs for 'Sample Status', 'Tag Status', and 'Alarm', and a table with 'Name' and 'Value' columns.

The system clock in the bottom right corner shows the date and time as 14-01-2015 22:17:16.

6.8 Complete the configuration of both Nodes with all the Units and, sample points...

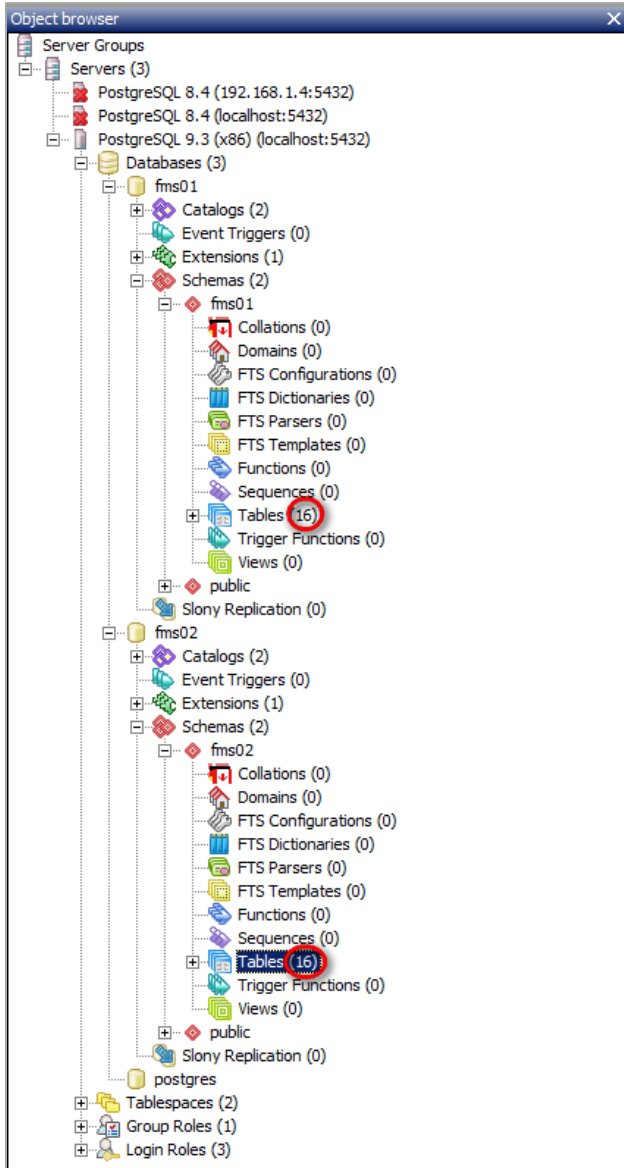
The screenshot shows the FMS Client Client interface. The 'Configured' section displays two nodes: FMS_DEMO01 and FMS_DEMO. Under FMS_DEMO01, the unit DEMO_HUMIDITY is shown with a value of 6.0 %RH. Under FMS_DEMO, the unit DEMO_OPC is shown with a value of 0.5 : 0 C/cuft. The 'Messages' table contains four entries, all indicating that the monitor has connected successfully from 192.168.1.50. The 'Statistics' panel shows various status fields such as Last Failure, Last Warning, Last Alarm, Last Reading, Last Ack., Recipe, Enabled, Ack.State, and Failure, with their respective values.

Node	Date/Time	Source	Type	Message
Local	14-01-2015 22:24:28	Local	Ok	Monitor Has Connected FMS_DEMO01 from 192.168.1.50
Local	14-01-2015 22:24:28	Local	Ok	Monitor Has Connected FMS_DEMO from 192.168.1.50
Local	14-01-2015 22:24:27	Local	Ok	Monitor Has Connected FMS_DEMO from 192.168.1.50
Local	14-01-2015 22:24:27	Local	Ok	Monitor Has Connected FMS_DEMO01 from 192.168.1.50

The screenshot shows the FMS Client Client interface. The 'Configured' section displays two nodes: FMS_DEMO01 and FMS_DEMO. Under FMS_DEMO01, the unit DEMO_1_HUMIDITY is shown with a value of 7.0 %RH. Under FMS_DEMO, the unit DEMO_OPC is shown with a value of 0.5 : 0 C/cuft. The 'Messages' table contains four entries, all indicating that the monitor has connected successfully from 192.168.1.50. The 'Statistics' panel shows various status fields such as Last Failure, Last Warning, Last Alarm, Last Reading, Last Ack., Recipe, Enabled, Ack.State, and Failure, with their respective values.

Node	Date/Time	Source	Type	Message
Local	14-01-2015 22:24:28	Local	Ok	Monitor Has Connected FMS_DEMO01 from 192.168.1.50
Local	14-01-2015 22:24:28	Local	Ok	Monitor Has Connected FMS_DEMO from 192.168.1.50
Local	14-01-2015 22:24:27	Local	Ok	Monitor Has Connected FMS_DEMO from 192.168.1.50
Local	14-01-2015 22:24:27	Local	Ok	Monitor Has Connected FMS_DEMO01 from 192.168.1.50

6.9 Now when looking inside the database “fms01” & “fms02” you can see that all tables are correctly created and data is getting stored in both databases.



TSI and TSI logo are registered trademarks of TSI Incorporated.
Windows is a registered trademark of Microsoft Corporation.



TSI Incorporated – Visit our website www.tsi.com for more information.

USA	Tel: +1 800 874 2811	India	Tel: +91 80 67877200
UK	Tel: +44 149 4 459200	China	Tel: +86 10 8219 7688
France	Tel: +33 1 41 19 21 99	Singapore	Tel: +65 6595 6388
Germany	Tel: +49 241 523030		