

---

# Advanced Data Base (8trd157)

## Lab7 Introduction to Distributed Data Base

(no report)

Paul Girard, Ph.D.

---

### Objectives:

Part I: Overview of Open/Ingres DBMS and compare some Oracle and Ingres SQL

Part II: Experiment a distributed database with Ingres/Net and Ingres/Star on 2 Sun servers  
(*sunensingres1.uqac.ca*, *sunensingres2.uqac.ca*). A synchronous replication will be explained using a two-phase commit.

---

## PART I Overview of Open/Ingres DBMS

### Methodology

#### A) First Method on Physical Server

##### A1 Creating an Open/Ingres database on the physical server sunensb.uqac.ca

```
sunensb:sun*****>createdb db***
```

*(replace \*\*\* by the 3 last digits of your user id sun\*\*\*\*)*

The command **destroydb bd\*\*\*** will destroy it

##### Creating the database *dbtest*

```
sunensb:pgirard> createdb dbtest
Creating database 'dbtest' . . .

    Creating DBMS System Catalogs . . .
    Modifying DBMS System Catalogs . . .
    Creating Standard Catalog Interface . . .
    Creating Front-end System Catalogs . . .

Creation of database 'dbtest' completed successfully.
```

##### Destroying the database *dbtest*

```
sunensb:pgirard> destroydb dbtest
Destroying database 'dbtest' . . .

Destruction of database 'dbtest' completed successfully.
```

##### A2. Creating and loading tables

Copy all files from my own directory. These files will create 4 tables and will load data from data files. Some files will be used in Part II.

---

```
sunensb: sun*****>cp /nfs/sunensc/professeurs/pgirard/tut_trd157/lab7/* .
                                     (some files will not be copied in your directory)
sunensb:sun*****> ls                                     check these files
```

### A3. How to use Open/Ingres interactive SQL

Interactive SQL (or terminal monitor) is the equivalent of Oracle SqlPlus

```
sunensb:sun*****> sql db***
```

#### Some useful commands for interactive SQL

(see Appendix B of *ingres\_sql.pdf* manual in the web site)

<b>*i file</b>	insert a <i>file</i> from the current directory in the query buffer
<b>*\p</b>	print on the screen the content of the query buffer
<b>*\g</b>	execute all commands in the query buffer
<b>*\r</b>	reset the query buffer
<b>*\a</b>	append to the query buffer
<b>*\q</b>	quit SQL et retour à Unix)
<b>*commit;\g</b>	commit all previous transactions and free table and page locks.
<b>*rollback;\g</b>	rollback all previous transactions & free table and page locks.
<b>*help;\g</b>	displays a summary of all tables and views in your database
<b>*help table;\g</b>	displays the characteristics of a table in your database

### A4. Creation of 4 tables in your database:

The file **cretab** contains similar tables definition than those used with Oracle (*except some datatypes [integer2] and the use of not default*)

#### cretab

```
-- Open/Ingres : Table creation for TUT-Lab7 8trd127
-- Paul Girard Ph.D. UQAC

drop table component;
drop table responsible;
drop table part;
drop table pa_agent;

create table pa_agent
    (emp_num      integer2      not null      not default      primary key,
     pa_name      char(15)      not null      not default);
```

```

create table responsible
  (emp_number integer2      not null      not default,
   part_number integer2     not null      not default,
   CONSTRAINT uni_responsible unique (emp_number, part_number));

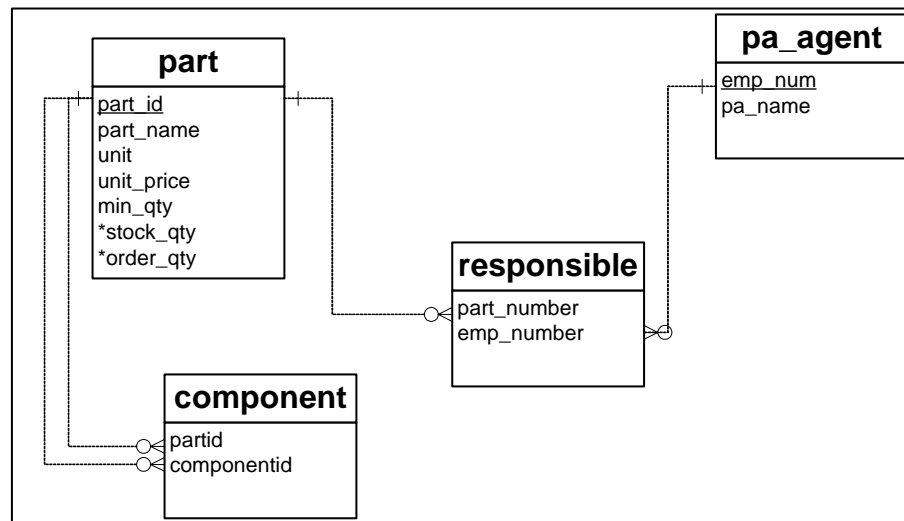
create table part
  (part_id      integer2      not null      primary key,
   part_name    char(15)     not null      not default,
   stock_qty    integer2      default 0,
   order_qty    integer2      default 0,
   min_qty      integer2      default 0,
   unit         char(10)     default 'unit',
   unit_price   money         default 0);

create table component
  (partid       integer2      not null      not default references part(part_id),
   componentid  integer2      not null      not default references part(part_id),
   CONSTRAINT uni_component unique (partid,componentid));

```

### Graphics Schema of these 4 Relational Tables

(do not forget: there is NO RELATION, only tables)



Load the file *cretab* in the query buffer and execute it

```

sunensb:sun*****>more cretab
sunensb:sun*****>sql bd***
*i cretab                                insert cretab in the query buffer
*p
*g                                        execution
*help;g                                the 4 tables should be displayed

```

---

## A5 Load data files into tables

The file **load** contains the command **copy from** (equivalent of Oracle *SQL\*Loader*) which can read a sequential data file into a table. There is also a command **copy to** to export tables data to files. c0 mean a comma is the delimiter between fields.

**load**

```
copy table pa_agent
(emp_num = c0,
 pa_name = c0) from 'pa_agentdat';
copy table part
(part_id = c0,
 part_name = c0,
 stock_qty = c0,
 order_qty = c0,
 min_qty = c0,
 unit = c0,
 unit_price = c0) from 'partdat';
copy table responsible
(emp_number = c0,
 part_number = c0) from 'responsibledat';
copy table component
(partid = c0,
 componentid = c0) from 'componentdat';
```

The sequential files contain the following data :

pa_agentdat	responsibledat	componentdat	part
100,John	100,1001	1001,1003	1001,motor 1,0,1,0,unit,2500
101,Luciano	100,1002	1001,1005	1002,motor 2,1,0,0,unit,2785
102,Bobbie	101,1003	1002,1003	1003,batteries AA,120,0,10,unit,4.95
	101,1004	1002,1006	1004,batteries 90C,10,0,1,unit,85.95
	102,1005		1005,alternator 1,4,0,0,unit,69.50
	102,1006		1006,alternator 2,2,1,1,unit,110.35

### Execution of load

*i load	(load file)
*p	(print the contents of load)
*g	(execution)

## 6. Test your transaction

Test this database with the same basic SQL transaction (*only pure SQL*) as your previous labs : *lisagent, lispart, quant, respon, modresp, explosion, implosion, crepart, invent, value*.

### Examples of an Ingres SQL session

```
sunensb:pgirard> sql bctest
```

```
INGRES TERMINAL MONITOR Copyright 2003 Computer Associates Intl, Inc.
```

```
Ingres SPARC SOLARIS Version II 2.6/0305 (su9.us5/00) login
```

```
Mon Mar 17 16:02:18 2008
```

```
continue
```

```
* select * from pa_agent;\g
```

```
Executing . . .
```

emp_nu	pa_name
100	John
101	Luciano
102	Bobbie

```
(3 rows)
```

```
continue
```

```
* \i explosion
```

```
continue
```

```
* \g
```

```
Executing . . .
```

part n	compon	name
1001	1003	batteries AA
1001	1005	alternator 1

```
(2 rows)
```

```
continue
```

```
* \i respon
```

```
continue
```

```
* \g
```

```
Executing . . .
```

pa_name	part_name
John	motor 1
John	motor 2
Luciano	batteries AA
Luciano	batteries 90C
Bobbie	alternator 1
Bobbie	alternator 2

```
(6 rows)
```

```
continue
```

```
* \q
```

```
Your SQL statement(s) have been committed.
```

```
Ingres Version II 2.6/0305 (su9.us5/00) logout
```

---

## PART II

### Distributed Database

#### Creation & Access of a Distributed Database with Two-Phase Commit

---

In this part, we will create 2 databases : **bd1a** on sunensa.uqac.ca and **bd1b** on sunensb.uqac.ca. Each database will have 2 different tables.

- **bd1a** will have the tables **part** and **component**,
- **bd1b** will have the tables **pa\_agent** and **responsible**.

In sunensa.uqac.ca, we will create a distributed database called **bddist**. This database will register the tables from **bd1a** and **bd1b** and will also have its own local table (*store*). Transactions of Part I will be tested in **bddist** with joints over tables owned by 2 different databases. A client program will also be used to gather some statistics. A two-phase commit will be tested.

### Methodology

#### 1. Access to Open/Ingres via TCP ports

- If the work is done outside UQAC buildings, a TCP port used by Ingres must be unlocked with a special security server OR specify in the UQAC external router a remote valid IP address.

valideur.uqac.ca

LOGIN Authentication

Username: ???????

Password: \*\*\*\*\*

Authentication Successful

- At the end of the session, TCP access ports must be secured by logging again to the same server using telnet. The first telnet does an authentication and the second telnet automatically does a logout.

valideur.uqac.ca

LOGOUT Authentication

Username: ???????

Password: \*\*\*\*\*

Logout Successful

#### 2. Creating *bd1a* and *bd1b* on 2 DBMS servers

*This work has been explained in Part I.*

sunensa: sun*****> <b>createdb bd1a</b>	(sunensa.uqac.ca, Solaris)
sunensb: sun*****> <b>createdb bd1b</b>	(sunensb.uqac.ca, Solaris)

### 3. Creating and loading tables in bd1a and bd1b

(*cretab* ==> *cretaba* & *cratabb*,      *load* ==> *loada*, *loadb*)

#### Creating and loading bd1a : *part* and *component*

```
sunensa:pgirard> sql bd1a
INGRES TERMINAL MONITOR Copyright 2003 Computer Associates Intl, Inc.
Ingres SPARC SOLARIS Version II 2.6/0305 (su9.us5/00) login
Tue Mar 18 15:23:10 2008

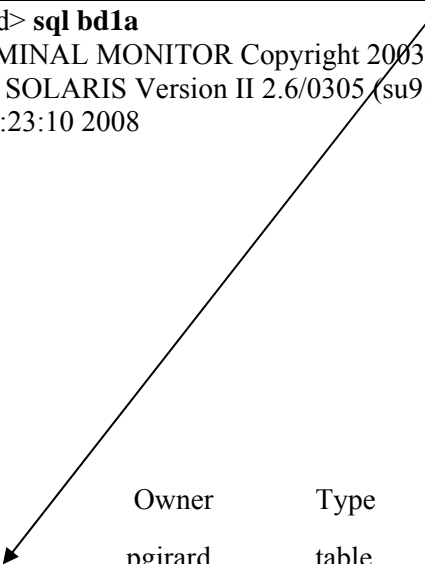
continue
* \i cretaba
continue
* \g
Executing ...

continue
* help;\g
Executing ...

Name                Owner      Type
-----
component            pgirard   table
part                 pgirard   table
(2 rows)

continue
* \i loada
\g
Executing ...

(6 rows)
(4 rows)
continue
* \q
Your SQL statement(s) have been committed.
```



#### Creating and loading bd1b : *pa\_agent* and *responsible*

```
sunensb:pgirard> sql bd1b
INGRES TERMINAL MONITOR Copyright 2003 Computer Associates Intl, Inc.
Ingres SPARC SOLARIS Version II 2.6/0305 (su9.us5/00) login
Tue Mar 18 15:30:36 2008

continue
* \i cretabb
continue
* \g
Executing ...

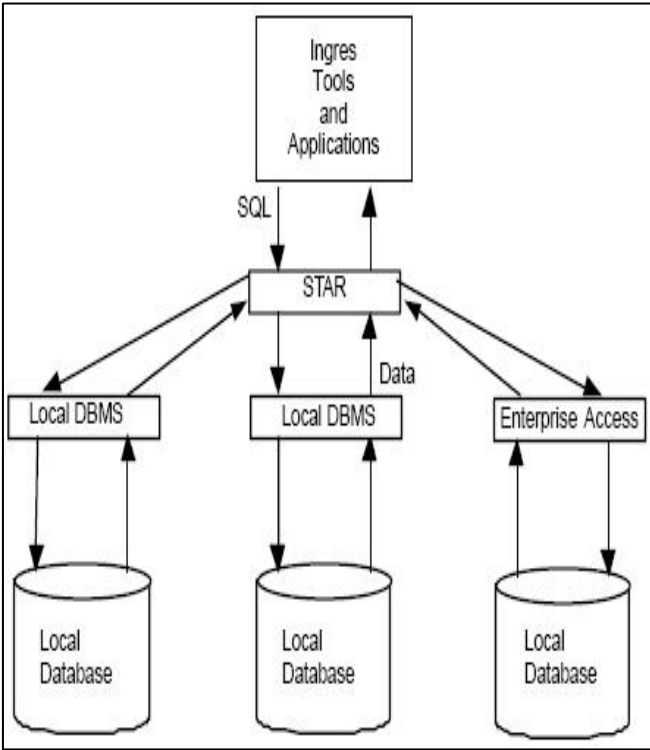
continue
* help;\g
Executing ...
```

Name	Owner	Type
<u>pa_agent</u>	pgirard	table
<u>responsible</u>	pgirard	table
(2 rows)		
continue		
* \i loadb		
\g		
Executing . . .		
(3 rows)		
(6 rows)		
continue		
* \q		
Your SQL statement(s) have been committed.		

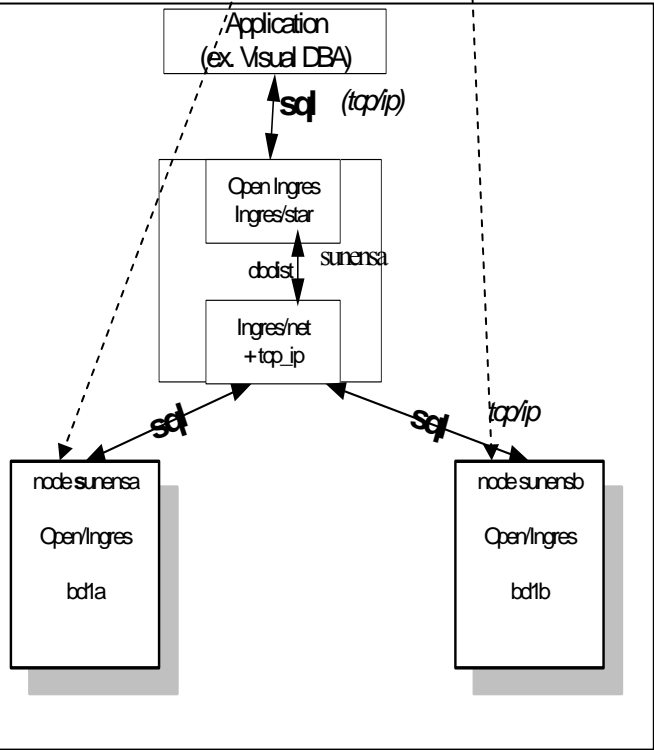
#### 4. Creating a distributed database using Ingres/Star

The general architecture of a **Star** system is shown in these 2 figures **Figure 1** shows a Star database receiving requests from clients and dispatching each request to local databases (*one server*). **Figure 2** shows a Star database controlling requests to databases installed on 2 DBMS servers identified by a name (*node*) controlled by **Ingres/Net** (*node sunensa, node sunensb*). Access rights on databases and tables must be given to Ingres for these 2 nodes.

**Fig. 1 : Ingres/Star with one DBMS servers**



**Fig. 2: Ingres/Star with 2 DBMS servers**





#### 4.1 Creation of a Star distributed database *bddist* on the master server *sunensa.uqac.ca*

```
sunensa:pgirard> createdb bddist/star bddistcdb
Creating distributed database 'bddist' . . .
    Creating DBMS System Catalogs . . .
    Modifying DBMS System Catalogs . . .
    Creating Standard Catalog Interface . . .
    Creating STAR System Catalogs . . .
    Initializing STAR System Catalogs . . .
    Modifying STAR System Catalogs . . .
    Creating Front-end System Catalogs . . .
Creation of distributed database 'bddist' successfully completed.
```

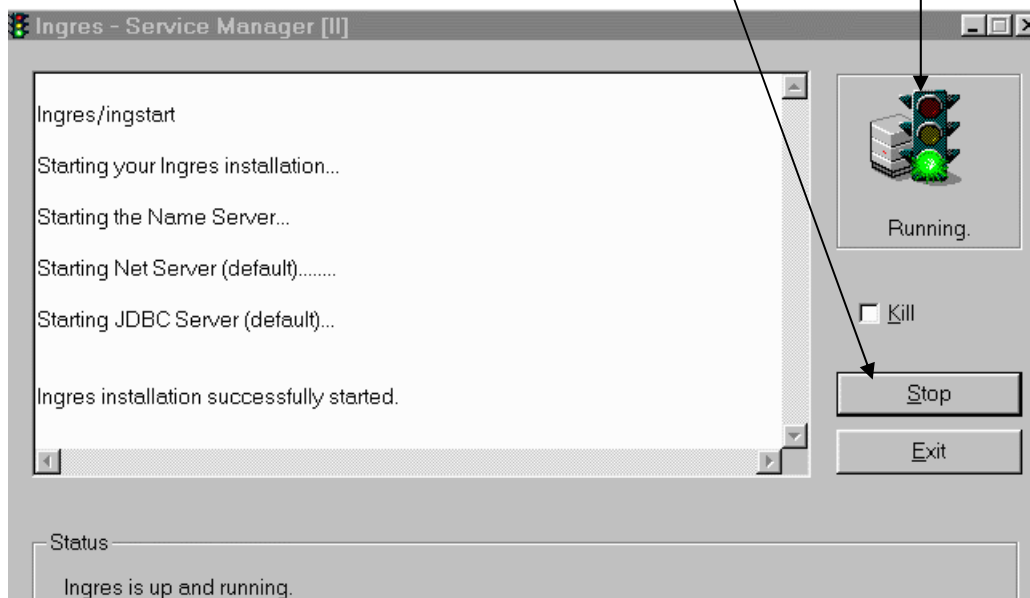
**Result of Ingres command *catalogdb* on *sunensa*  
showing all databases owned by *pgirard***

Database	Owner	Type
bddist	pgirard	local
bddistcdb	pgirard	star-CDB ( <i>coordinator database</i> )
bddist	pgirard	distributed

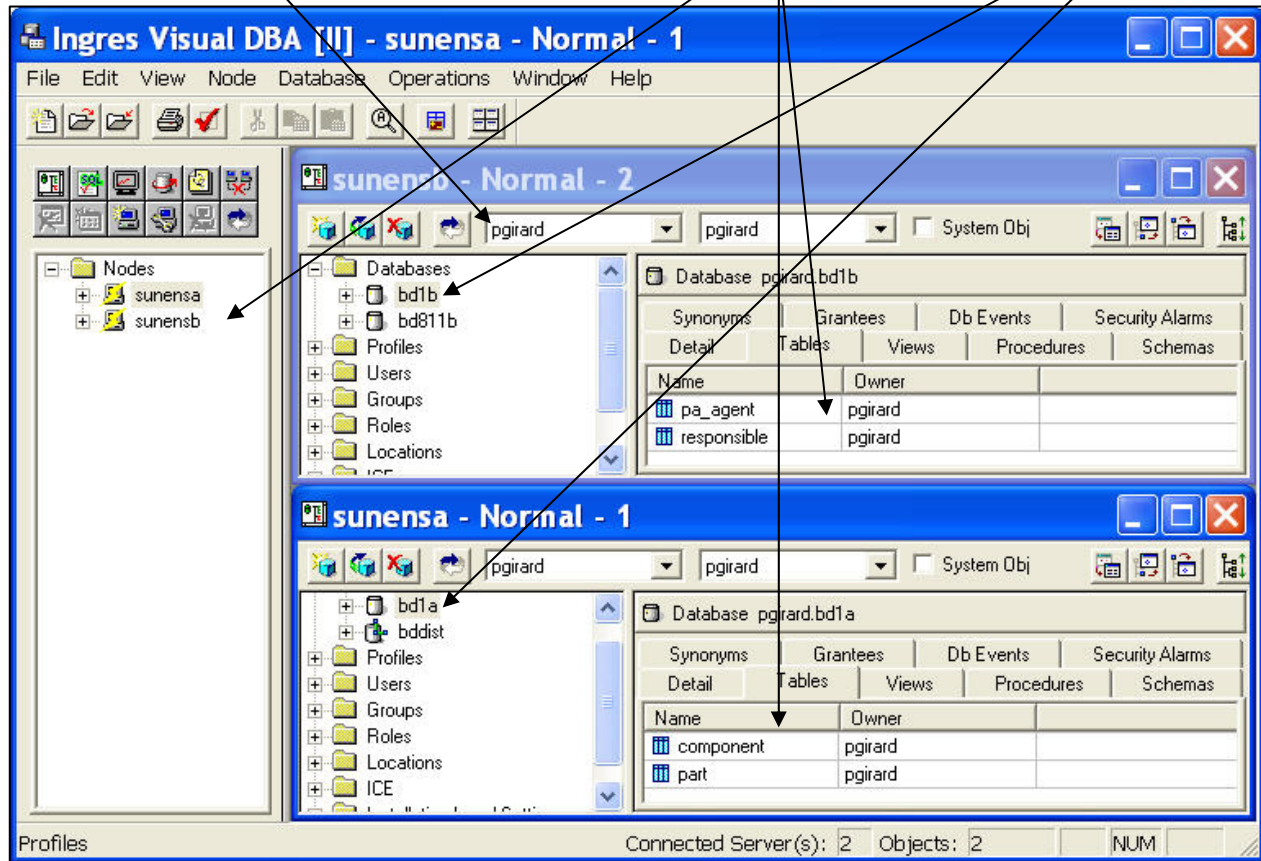
A **coordinator database (CDB)** contains the catalogs that the Star server uses to keep track of distributed objects. When a user requires information, the Star server accesses the coordinator database and associated local databases via the local DBMS server to get the information.

#### 4.2 Creation of access rights for the "Ingres user" for *bd1a* and *bd1b* by using Ingres clients (the Ingres server *NetUtil* may also be used)

- Start the Ingres client ***Ingres Service Manager*** and click on **Start**. The green light specifies that all local Ingres services are started. After the session **Stop** will end those services.



- Start the Ingres client **Visual DBA to control the access right.** *sunensa* and *sunensb* are the names given by the client owner to identify *sunensa.uqac.ca* and *sunensb.uqac.ca*. The name server of the preceding ingres client started the name server for this reason. The databases owned by the owner are identified in the window. Tables in each one are shown.



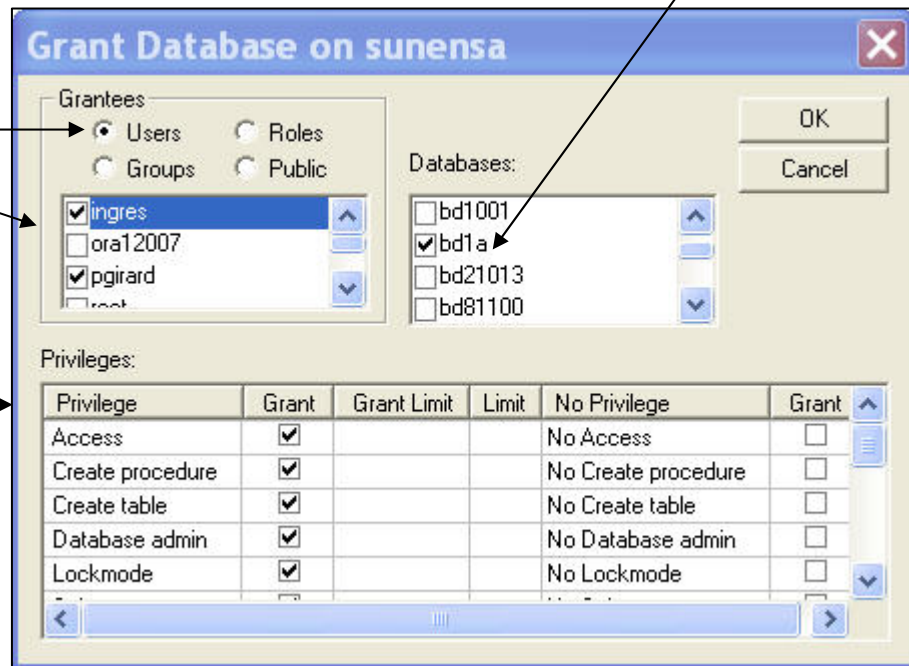
- To create a node, activate **Node** => **Add** for each server having a database and fill the required information.

**Virtual Node:** *sunensa (or sunensb)*  
**User name:** your user code to login on the server *sunensa (or sunensb)*  
**Password:** your password  
**Remote Node:** *sunensa.uqac.ca (sunensb.uqac.ca)*  
**Protocol:** *wintcp*  
**Listen address:** *ii*

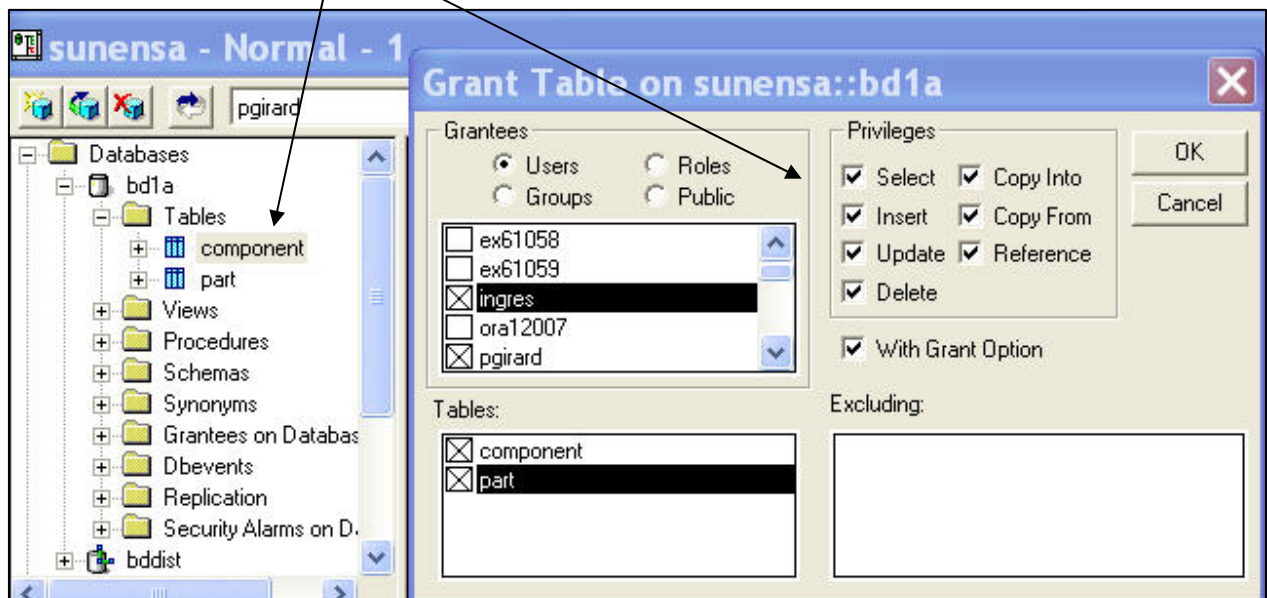
- Connect the client to a virtual node with the menu **Node** => **Connect/DOM**

- To give the required access rights to a database, select the node, the database and activate **Edit => Grant**

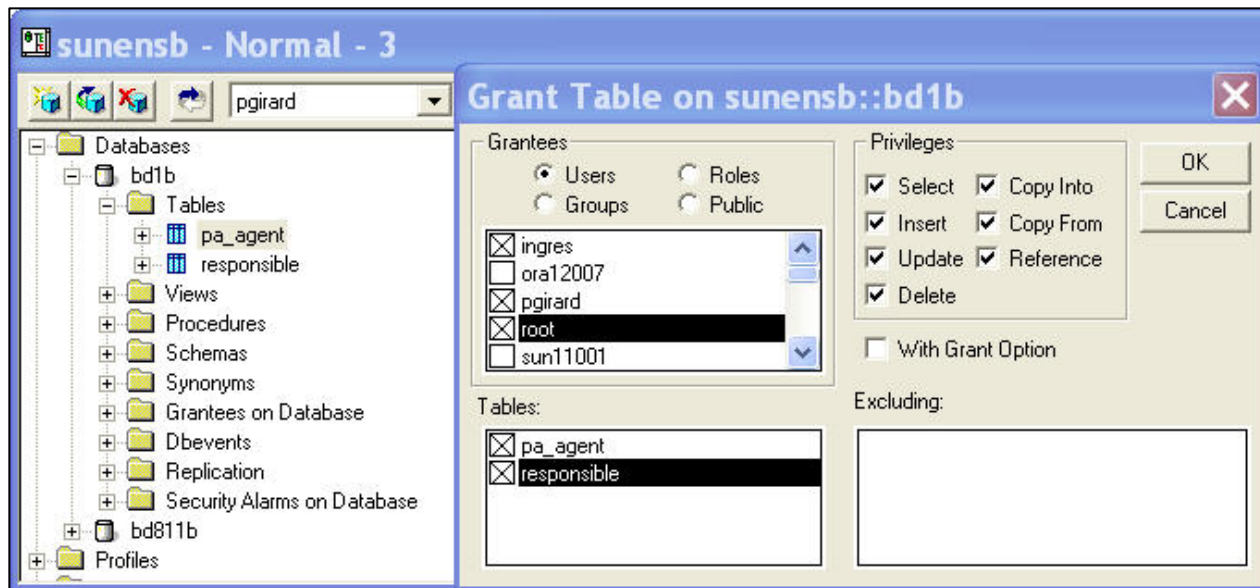
- Click Users
- Click all users to give those rights (*yourself, Ingres*)
- Click the required access rights
- and click OK



- The same procedure must be done for the database **bd1b** on **sunensb**.
- The access rights must now be given to the tables used by the distributed database (*actually 2 tables in each database*) by clicking on any table in one database and activate **Edit ==> Grant**. Click on all tables used in **bddist** and give all privileges necessary to support the transactions.



The same procedure is used for the tables of **bd1b** (*pa\_agent, responsible*)



### 4.3 Register tables in the distributed database

All tables needed par *bddist* and owned by a local database (*bd1a*) or a remote databases (*bd1b*) must be defined in the distributed database *bddist*. The file *linkdis* contains those links..

*linkdis* defines a link for each table owned by a database on a virtual node

```
remove pa_agent;
remove responsible;
remove part;
remove component;

register table pa_agent as link
with node=sunensb,
database=bd1b;
register table responsible as link
with node= sunensb,
database= bd1b;

register table part as link
with node=sunensa,
database= bd1a;
register table component as link
with node= sunensa,
database= bd1a;
```

### Registering distributed tables in the distributed database *bddist*

```
sunensa:pgirard> sql bddist/star
INGRES TERMINAL MONITOR Copyright 2003 Computer Associates Intl, Inc.
Ingres SPARC SOLARIS Version II 2.6/0305 (su9.us5/00) login
Wed Mar 19 09:29:53 2008
continue
* \i linkdis
```

```

continue
* \g
Executing . . .
continue
* help;\g
Executing . . .

Name                                Owner                                Type
ddx_1089_1090                       pgirard                             index
ddx_1091_1092                       pgirard                             index
ddx_1093_1094                       pgirard                             index
ddx_1095_1096                       pgirard                             index
ddx_1095_1097                       pgirard                             index
ddx_1095_1098                       pgirard                             index
component                           pgirard                             table
pa_agent                           pgirard                             table
part                               pgirard                             table
responsible                         pgirard                             table
(10 rows)
continue
* \q
Your SQL statement(s) have been committed.
Ingres Version II 2.6/0305 (su9.us5/00) logout

```

#### 4.4 SQL Test of the distributed database *bddist*

respon has a joint between *part* of *bd1a* in *sunensa* and  
*pa\_agent* & *responsible* of *bd1b* on *sunensb*

```

sunensa:pgirard> sql bddist/star
INGRES TERMINAL MONITOR Copyright 2003 Computer Associates Intl, Inc.
Ingres SPARC SOLARIS Version II 2.6/0305 (su9.us5/00) login
Wed Mar 19 09:36:33 2008

* \i respon
continue
* \p

select pa.pa_name,p.part_name
from pa_agent pa, part p, responsible r
where r.part_number=p.part_id and r.emp_number=pa.emp_num;
continue
* \g
Executing . . .

```

pa_name	part_name
John	motor 1
John	motor 2
Luciano	batteries AA
Luciano	batteries 90C
Bobbie	alternator 1
Bobbie	alternator 2

---

## 4.5 Two-Phase Commit

The term **two-phase commit** comes from the fact that there are two phases to committing a distributed transaction where two or more databases are updated. The two phases consist of:

- Agreement between all sites to commit
- Committing the updates

**Ingres/Star** manages these two phases in the following way:

**Phase 1** begins when the user issues a commit statement. **Star** sends a **prepare-to-commit** notice to each database involved in the distributed transaction. If all databases indicate that they are prepared to commit, Star makes the decision to commit the transaction. The local databases remain in the prepare-to-commit state and wait for Star's instruction to commit.

**Phase 2** Star sends a commit to all sites involved in the transaction. **Star** guarantees that all sites will commit. If the connection to a local database is lost between the time that Star decides to commit and the time the local database actually obeys that instruction, Star keeps trying to complete the transaction until the connection is restored and the commit is made. Star does not return control to the end user until all nodes have committed.

If any part of Phase 1 fails, for example, if Star loses a network connection to a node before all databases are prepared to commit, Star rolls back the transaction at all sites, including those that are already prepared to commit.

If any part of Phase 2 fails, Star still eventually commits the transaction.

### Example of a Two-Phase Commit

*responsible* in *db1b (sunensb)* and *part* in *db1a (sunensa)* are updated. If Star comes back to the client, that means that the commit has been done by each node properly.

```
sunensa:pgirard> sql bddist/star
INGRES TERMINAL MONITOR Copyright 2003 Computer Associates Intl, Inc.
Ingres SPARC SOLARIS Version II 2.6/0305 (su9.us5/00) login
Wed Mar 19 09:44:10 2008

* update responsible set emp_number=100 where part_number=1003;\g
Executing . . .
(1 row)
continue

* update part set min_qty=5 where part_id=1003;\g
Executing . . .
(1 row)
continue

* commit;\g
Executing . . .
continue ▲
```

---

## 4.6 Creating a local table in a distributed database

Local tables may be created in a distributed database like any standard database. The following example creates the table **store** in the distributed database **bddist** then load this table with data from the sequential file **storedat**. SQL transactions can be done using local tables in the distributed database, registered local tables in local or remote databases.

### Example of a local table creation in *bddist*

```
sunensa:pgirard> sql bddist/star
INGRES TERMINAL MONITOR Copyright 2003 Computer Associates Intl, Inc.
Ingres SPARC SOLARIS Version II 2.6/0305 (su9.us5/00) login
Wed Mar 19 10:12:38 2008
```

```
continue
* create table store
*   (name      char(15)      not null      not default,
*   partid    integer2      not null      not default);\g
Executing . . .
```

```
continue
* copy table store
*   (name = c0,
*   partid = c0) from 'storedat';\g
Executing . . .
```

```
(15 rows)
continue
* select * from store;\g
Executing . . .
```

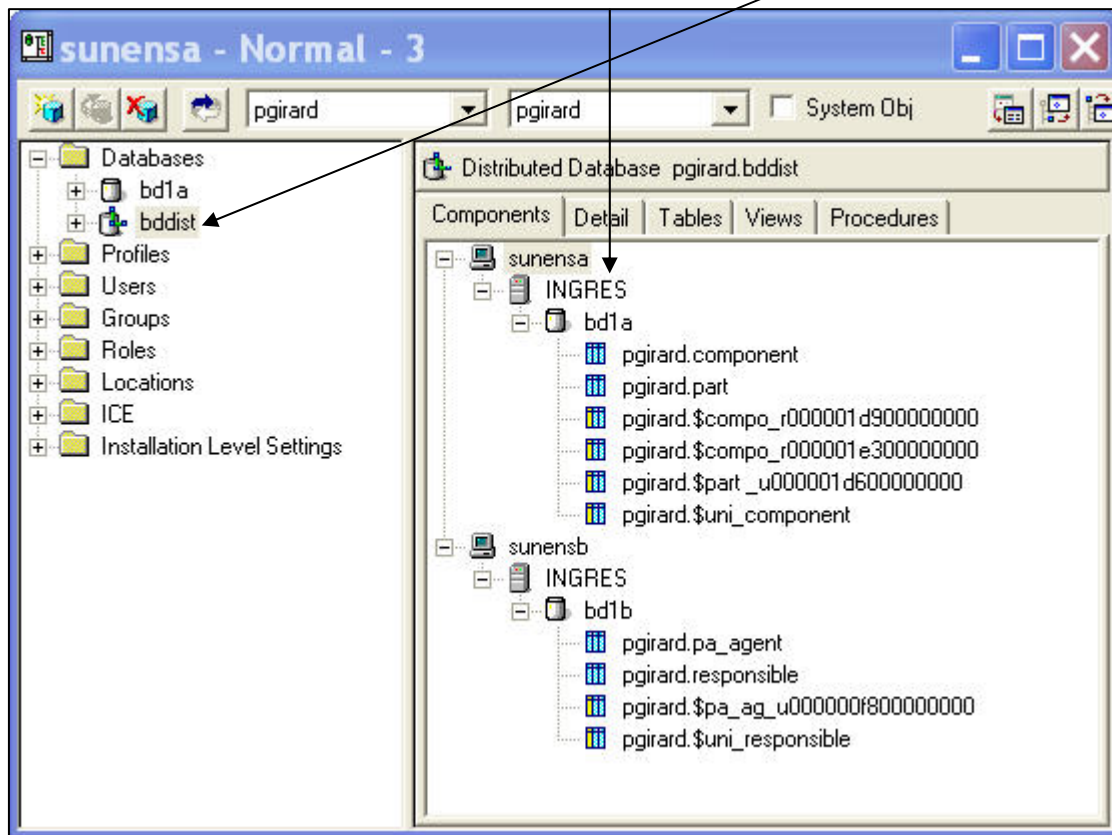
name	partid
Beijing	1001
Beijing	1002
Beijing	1003
Beijing	1004
Tianjin	1005
Tianjin	1006
Tianjin	1001
Tianjin	1002
Tianjin	1003
Tianjin	1004
Shanghai	1001
Shanghai	1002
Shanghai	1003
Shanghai	1004
Shanghai	1005

```
(15 rows)
continue
```

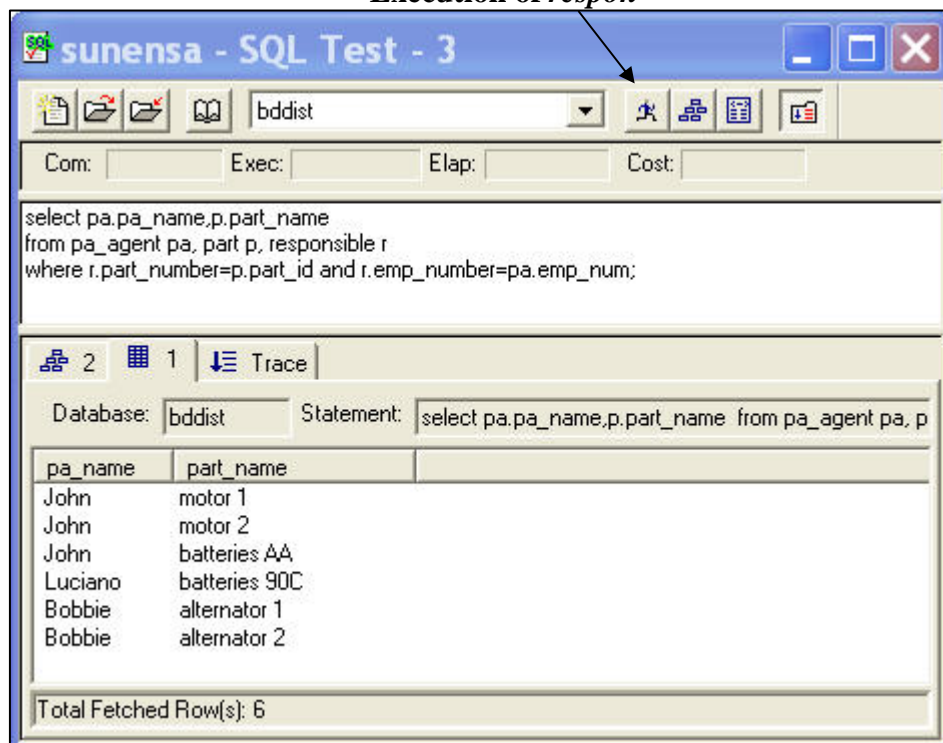


## 5 SQL test on databases *bd1a* and *bddist* using *Visual DBA*

Figure showing the contents of *bddist*

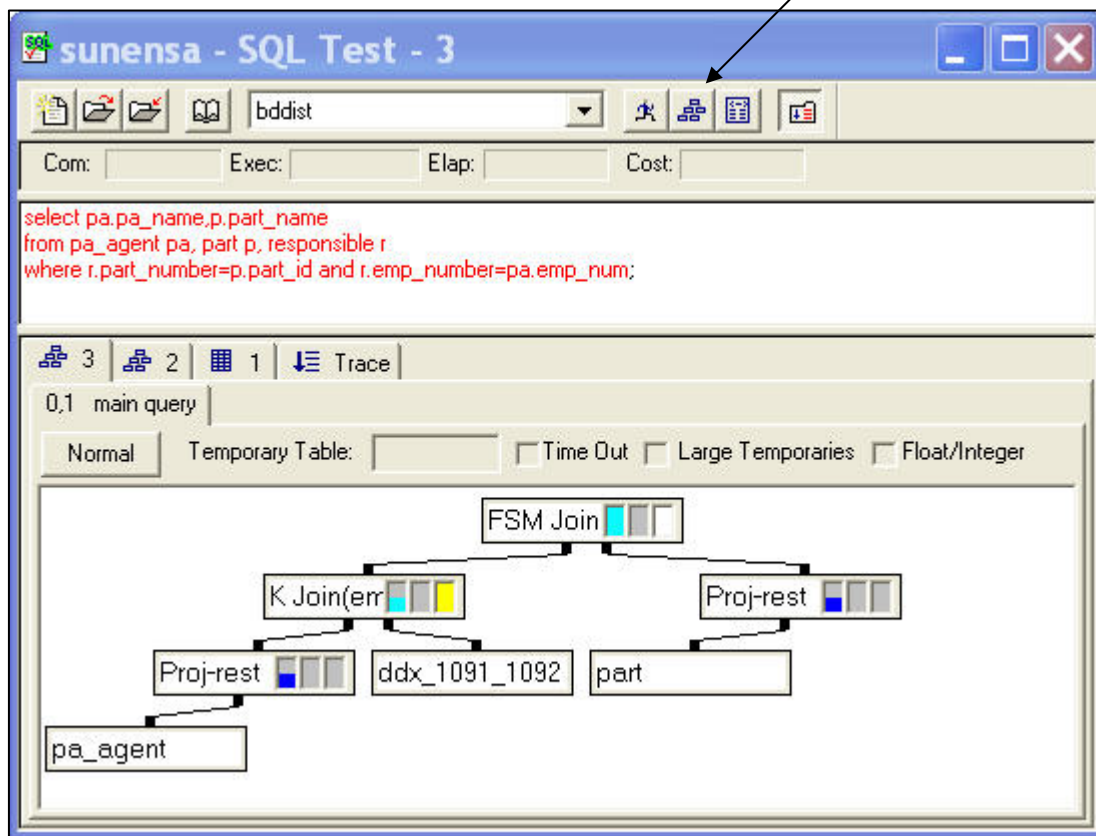


Execution of *respon*





### Execution plan for the transaction *respon*



### Cost analysis of this execution

**Com:** compilation time

**Exec:** cpu time

**Elap:** Elapsed time

**Cost:** cost of the request in terms of disk I/O

part number	component	name
1001	1003	batteries AA
1001	1005	alternator 1

Total Fetched Row(s): 2

Conversion of the SQL request to an XML format (*partial display*)

The screenshot shows a window titled "sunensa - SQL Test - 3". The SQL query is:

```
select pa.pa_name,p.part_name
from pa_agent pa, part p, responsible r
where r.part_number=p.part_id and r.emp_number=pa.emp_num;
```

The result set is displayed in XML format:

```
<?xml version="1.0" encoding="windows-1252" ?>
<!DOCTYPE resultset (View Source for full doctype...)>
- <resultset>
- <row>
  <column column_name="pa_name" a-
    dtype="is_null boolean">John</column>
  <column column_name="part_name" a-
    dtype="is_null boolean">motor 1</column>
</row>
- <row>
  <column column_name="pa_name" a-
    dtype="is_null boolean">John</column>
  <column column_name="part_name" a-
    dtype="is_null boolean">motor 2</column>
</row>
- <row>
  <column column_name="pa_name" a-
    dtype="is_null boolean">John</column>
  <column column_name="part_name" a-
```

Below this, another window shows the same data in a table format:

pa_name	part_name
John	motor 1
John	motor 2
John	batteries AA
Luciano	batteries 90C
Bobbie	alternator 1
Bobbie	alternator 2

Execution of the XML  
transaction respon

---

### Important Note

1. At the end of a session, activate **Stop** with the program **Ingres Service Manager**.
2. Then Destroy the distributed database with the command with a telnet session on sunensa, then on *sunensb*.

<b>sunensa: sun***</b>	<b>destroydb</b> <i>name_of_distributed_database</i>
<b>sunensa: sun***</b>	<b>destroydb</b> <i>name_of_local_database_on_sunensa</i>
<b>sunensb: sun***</b>	<b>destroydb</b> <i>name_of_local_database_on_sunensb</i>

3. Lock the Ingres TCP port by executing valideur.uqac.ca with a given username and password.