



The Grid Relational Catalog Project Tutorial

Sandro Fiore (sandro.fiore@unile.it)

Alessandro Negro (alessandro.negro@unile.it)

SPACI Consortium & University of Salento, Lecce

Authorization

Select Queries

grelc-das-query-memory

grelc-das-query-dime

grelc-das-query-stream

GRelC Portal

<https://grid.ct.infn.it/twiki/bin/view/GILDA/GRelCDataAccessService>

<https://grid.ct.infn.it/twiki/bin/view/GILDA/GRelCPortal>

<https://grid.ct.infn.it/twiki/bin/view/GILDA/GRelCPortalAdvanced>

Before starting be sure you have a valid proxy ...

To create a valid proxy, run the following command

```
grid-proxy-init
```

With this proxy you will be able to access the sakila database on grelc02.unile.it (only select statements)

To carry out SELECT queries you can use the following commands

grenc-das-query-memory

grenc-das-query-dime

grenc-das-query-stream

These kinds of query were designed to address high performance

play help

das-query-memory -h

query-das-memory [options]

- s VAL Sets GReIC DAS IP or hostname (default: localhost)
- p NUM Sets GReIC DAS port (default: 18500)
- D database VAL Sets grid database name
- Q query VAL Sets query value
- c chunk NUM Sets chunk dimension (default:0 no chunk)
- t Prints result in table format (default)
- x Prints result in xml format
- l Prints result in html format
- u Shows usage

ES:

mission without chunking

das-query-memory -s hostname -p 18500 -D database -Q 'select * from table'

mission with chunking

das-query-memory -s hostname -p 18500 -D database -Q 'select * from table' -c 50

Submits a query and displays its result in tabular format. The result is transferred one-shot.

```
as-query-memory -s grelc02.unile.it -p 18500 -D sakila -Q "select * from actor" -t
```

Submits a query and displays its result in XML format. The result is transferred one-shot.

```
as-query-memory -s grelc02.unile.it -p 18500 -D sakila -Q "select * from actor" -x
```

Submits a query and displays its result in XML format. The result is transferred in chunks of 50 lines. The result shown is exactly the same of example no. 3

```
as-query-memory -s grelc02.unile.it -p 18500 -D sakila -Q "select * from actor" -x -c 50
```

play help

das-query-dime -h

query-das-dime [options]

er VAL Sets GReIC DAS IP or hostname (default: localhost)

NUM Sets GReIC DAS port (default: 18500)

abase VAL Sets grid database name

ry VAL Sets query value

nk NUM Sets chunk dimension (default:0 no chunk)

Compresses result file

Prints result in table format (default)

Prints result in xml format

Shows usage

ES:

mission with compression

s-query-dime -s hostname -p 18500 -D database -Q 'select * from table' -z

mission with chunking

s-query-dime -s hostname -p 18500 -D database -Q 'select * from table' -c 50

omits a query and displays its result in tabular format. The result is transferred one-shot.

```
as-query-dime -s grelc02.unile.it -p 18500 -D sakila -t -Q "select * from actor"
```

omits a query and displays its result in XML format. The result is transferred one-shot.

```
as-query-dime -s grelc02.unile.it -p 18500 -D sakila -x -Q "select * from actor"
```

omits a query and displays its result in XML format. The result is transferred in chunks of 50 files.

```
as-query-dime -s grelc02.unile.it -p 18500 -D sakila -x -Q "select * from actor" -c 50
```

w, let's look at the files in your current folder. Each file is a resultset stored client-side.

's see what's inside one of the resultsets stored client-side.

`grelc02.unile.it_18500_171-singlequery.xml`

omits a query and displays its result in XML format. The result is compressed (using `-z` option) and transferred in chunks of 50 tuples.

`as-query-dime -s grelc02.unile.it -p 18500 -D sakila -x -Q "select * from actor" -z`

w, let's look at the files in your current folder again. You can find a compressed and an compressed file for each resultset stored client-side.

Submits a query and displays its result in tabular format. The result is transferred by using streaming.

```
das-query-stream -s grelc02.unile.it -p 18500 -D sakila -t -Q "select * from actor"
```

Submits a query and displays its result in XML format. The result is transferred one-shot.

```
lc-das-query-stream -s grelc02.unile.it -p 18500 -D sakila -x -Q "select * from actor"
```

Before starting be sure you have a valid proxy with the correct GRelC DAS Role

To create a valid proxy, run one of the following commands

grid-proxy-init

With this proxy you will be able to access the sakila database on grelc02.unile.it...



GRELC

AN EASY WAY TO MANAGE GRID DATABASES

Query

[Synchronous Query](#)

GRELC DASs

[Servers List](#)

Operations

[System Users](#)

Hosts

[Instances](#)

Grid databases

[Grid Database Users](#)

Virtual Organizations

[Association](#)

Maps

Session

[Logout](#)

GRELC Portal (BETA)

The GRELC Portal (BETA) is intended for administrators and end users of the GRELC DAS Server. It offers a direct method of management and it really takes a few seconds to:

- manage several GRELC DASs at the same time;
- submit queries;
- manage the enterprise grid;
- manage users and their privileges;
- define grid databases and their properties;
- manage VOs;
- view metadata information concerning the schema of your database/tables.

Please share your opinion, comments and experience on the GRELC Portal with the GRELC community. It will help us increasing the features of the service provided, trying to satisfy your requirements as much as possible.

Upload your Proxy

No valid proxy for the user was found. Please run grid-proxy-init in order to create a valid proxy and upload it by means of the form provided below. After that you'll be able to **login** and interact with your GRELC DAS servers until the expiration of your proxy.



Sat Oct 13 09:53:24 CEST 2007



GRELC

AN EASY WAY TO MANAGE GRID DATABASES

- Home
- Components
- Downloads
- Documentation
- News
- Events
- Publications
- Staff
- Deployment
- Portal (BETA)

Query

Synchronous Query

GRELC DASs

Servers List

greclgilda

Operations

System Users

Hosts

Instances

Grid databases

Grid Database Users

Virtual Organizations

Association

Maps

Session

Logout

GRELC DAS Servers

<input type="checkbox"/>	Alias	Hostname	Port
<input type="checkbox"/>	greclgilda	grecl02.unile.it	18500

Sat Oct 13 09:53:44 CEST 2007



GRELC

AN EASY WAY TO MANAGE GRID DATABASES

- Home
- Components
- Downloads
- Documentation
- News
- Events
- Publications
- Staff
- Deployment
- Portal (BETA)

Query

Synchronous Query

GRELC DASs

Servers List

grelegilda

Operations - grelcgilda

System Users

Hosts

Instances

Grid databases

Grid Database Users

Virtual Organizations

Association

Maps

Session

Logout

Grid Databases Management - grelcgilda

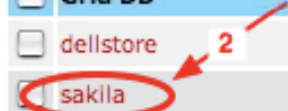
<input type="checkbox"/>	Grid DB	Physical DB	DBMS	Instance	Hostname
<input type="checkbox"/>	dellstore	dellstore	postgres	postgres_on_...	localhost
<input checked="" type="checkbox"/>	sakila	sakila	mysql	mysql_on_loc...	localhost
<input type="checkbox"/>	uniutrd	uniutrd	postgres	postgres_on_...	gandalf.unil...

Register Remove Selected

Please click on the checkbox just behind this message, if you want to create a grid DB and then register an already existent legacy database.

<gridDB> - HOST - - INSTANCE - <physicalDB> +

? Reset Apply



Sat Oct 13 09:54:05 CEST 2007



GRELC

AN EASY WAY TO MANAGE GRID DATABASES

- Home
- Components
- Downloads
- Documentation
- News
- Events
- Publications
- Staff
- Deployment
- Portal (BETA)

Query

Synchronous Query

GRELC DASs

Servers List

grelcgilda

Operations - grelcgilda

System Users

Hosts

Instances

Grid databases

Grid Database Users

Virtual Organizations

Association

Maps

Session

Logout

Tables Management - grelcgilda - sakila

Tables
<input checked="" type="radio"/> store
<input type="radio"/> staff
<input type="radio"/> rental
<input type="radio"/> payment
<input type="radio"/> language
<input type="radio"/> inventory
<input type="radio"/> film_text
<input type="radio"/> film_category
<input type="radio"/> film_actor
<input type="radio"/> film
<input type="radio"/> customer
<input type="radio"/> country
<input type="radio"/> city
<input type="radio"/> category
<input type="radio"/> address
<input type="radio"/> actor

Back

Sat Oct 13 09:54:38 CEST 2007



GRELC

AN EASY WAY TO MANAGE GRID DATABASES

[Home](#)
[Components](#)
[Downloads](#)
[Documentation](#)
[News](#)
[Events](#)
[Publications](#)
[Staff](#)
[Deployment](#)
[Portal \(BETA\)](#)

Query

[Synchronous Query](#)

GRELC DASs

[Servers List](#)
[grelcgilda](#)

Operations - grelcgilda

[System Users](#)
[Hosts](#)
[Instances](#)
[Grid databases](#)
[Grid Database Users](#)
[Virtual Organizations](#)
[Association](#)
[Maps](#)

Session

[Logout](#)

Your proxy is valid until 23/12/2007 17:11:42

Tables Management - grelcgilda

Database - sakila

Table - actor

fields	types	length	primary key	isNullable	precision	scale	unique	autoincrement
actor_id	smallint	5	✓					
first_name	varchar	45						
last_name	varchar	45						
last_update	timestamp			✓				

[Back](#)

Sat Oct 13 09:54:53 CEST 2007



GRELC

AN EASY WAY TO MANAGE GRID DATABASES

[Home](#)
[Components](#)
[Downloads](#)
[Documentation](#)
[News](#)
[Events](#)
[Publications](#)
[Staff](#)
[Deployment](#)
[Portal \(BETA\)](#)

Query

[Synchronous Query](#)

GRELC DASs

[Servers List](#)
[grelcgilda](#)

Operations - grelcgilda

[System Users](#)
[Hosts](#)
[Instances](#)
[Grid databases](#)
[Grid Database Users](#)
[Virtual Organizations](#)
[Association](#)
[Maps](#)

Session

[Logout](#)

Synchronous Query Result

Server - grelcgilda

Database - sakila

Query - select * from actor

actor_id	first_name	last_name	last_update
1	PENELOPE	GUINNESS	2006-02-15 04:34:33
2	NICK	WAHLBERG	2006-02-15 04:34:33
3	ED	CHASE	2006-02-15 04:34:33
4	JENNIFER	DAVIS	2006-02-15 04:34:33
5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
6	BETTE	NICHOLSON	2006-02-15 04:34:33
7	GRACE	MOSTEL	2006-02-15 04:34:33
8	MATTHEW	JOHANSSON	2006-02-15 04:34:33
9	JOE	SWANK	2006-02-15 04:34:33
10	CHRISTIAN	GABLE	2006-02-15 04:34:33
11	ZERO	CAGE	2006-02-15 04:34:33
12	KARL	BERRY	2006-02-15 04:34:33
13	UMA	WOOD	2006-02-15 04:34:33
14	VIVIEN	BERGEN	2006-02-15 04:34:33
15	CUBA	OLIVIER	2006-02-15 04:34:33
16	FRED	COSTNER	2006-02-15 04:34:33
17	HELEN	VOIGHT	2006-02-15 04:34:33
18	DAN	TORN	2006-02-15 04:34:33
19	BOB	FAWCETT	2006-02-15 04:34:33
20	LUCILLE	TRACY	2006-02-15 04:34:33

Previous 1 2 3 4 5 6 7 8 9 10 Next

Back

Sat Oct 13 09:55:15 CEST 2007



GRELC

AN EASY WAY TO MANAGE GRID DATABASES

- Home
- Components
- Downloads
- Documentation
- News
- Events
- Publications
- Staff
- Deployment
- Portal (BETA)

Query

[Synchronous Query](#)

GRELC DASs

[Servers List](#)

grelcgilda

Operations

[System Users](#)

[Hosts](#)

[Instances](#)

[Grid databases](#)

[Grid Database Users](#)

[Virtual Organizations](#)

[Association](#)

[Maps](#)

Session

[Logout](#)

Synchronous Query Submission

Server	<input type="text" value="grelcgilda"/>
Grid database	<input type="text" value="sakila"/>
Number of tuples for page	<input type="text" value="20"/>
Query	<input type="text" value="select * from film"/>

's start with a simple select query...

select * from film

This query retrieves all of the films available in the "sakila" database

's now retrieve a list of films whose length is longer than 90 minutes...

ect * from film where length > 90

pretty cool! The GReIC Portal gives us a list of films

Why do not we search a list of actors involved in the films listed before?

```
select first_name, last_name, title, length from (film join film_actor on  
film.film_id=film_actor.film_id) join actor on actor.actor_id=film_actor.actor_id  
length > 90
```

ables are now involved in our select query

What if we want to know the film category of the films listed before?

```
select category.name as category_name, first_name, last_name, title, length from ((  
join film_actor on film.film_id=film_actor.film_id) join actor on  
actor.actor_id=film_actor.actor_id) join film_category on film.film_id =  
film_category.film_id) join category on  
category.category_id=film_category.category_id where length > 90
```

ables are now involved in our select query

us now know who sells the films listed before, together with the category and the actor

```
select category.name as category_name, first_name, last_name, title, length from ((  
join film_actor on film.film_id=film_actor.film_id) join actor on  
actor.actor_id=film_actor.actor_id) join film_category on film.film_id =  
film_category.film_id) join category on  
category.category_id=film_category.category_id where length > 90
```

ables are now involved in our select query

u can enrich the query as you want, no matter how many tables are involved. If the DBL
derstands your query, the GRelC DAS understands it too!