

About Postgres and the Movie Database

January 2014

Students in INF1300, INF3100, INF5100 get access to the postgres server `dbpg-ifi-kurs.uio.no`. Login from unix systems to postgres via the `psql` interface when on linux machines on the Ifi network:

```
> psql -h dbpg-ifi-kurs -U username -d dbnavn
```

where `username` is your ordinary UiO user name. Each user has a database (work space) with the same name as the user name. The command

```
> psql -h dbpg-ifi-kurs -U username -d username
```

or

```
> psql -h dbpg-ifi-kurs -U username
```

gives access to the work space. Here each can create his/her own database schema.

When logged in the system responds with the prompt `"=>"`. You can then write SQL commands and postgres commands. All sql commands must be terminated with semicolon. If forgotten, you get the prompt `"->"`, i.e. `psql` believes that the command continues on the next line.

Alternatively, prepare your command in a text file, e.g. `filename.sql`, and run it with the command

```
=> \i filename.sql
```

If you get error messages like `ERROR: character 0xc2bd of encoding "UTF8" has no equivalent in "LATIN9"`, try issuing the command

```
=> \encoding UTF-8
```

Exit `psql` with the command

```
=> \q
```

Postgres documentation: The current version of Postgres is 9.2. You can find postgres documentation on

<http://www.postgresql.org/docs/9.2/interactive/>

For more about `psql`, see

<http://www.postgresql.org/docs/9.2/interactive/app-psql.html>

Information about postgres from windows or from home: The simplest way is to use `ssh` to Ifi's login-cluster (`login.ifi.uio.no`) and then use `psql`. Use for instance PuTTY. You can also use VPN.

There are Windows installers for those who use Windows:

<http://www.postgresql.org/download/windows/> [Open URL]

The installation program from EnterpriseDB installs among others pgAdminIII and psql.

The Movie Database

There are two ways to get access to the movie database, each is described below.

A. Access to the work space of the movie database

You get direct access to the work space of the movie database by logging in as follows:

```
> psql -h dbpg-ifi-kurs -U username -d fdb
```

or with '`\c fdb`' when you are in your own work space. (This can be used the other way around too; when you are in 'fdb' you return to your own work space by issuing the command '`\c username`'.) You then get read access to the database and can perform queries against it.

Example: "How many cinema movies does the database contain?"

```
fdb=> select count(*) as Ccount
fdb-> from filmitem
fdb-> where filmtyp = 'C';
  ccount
-----
  549782
(1 row)
```

```
fdb=>
```

The drawback of performing queries in the fdb work space, is that you are not allowed to define auxiliary tables nor views.

B. Access to the movie database from your own work space

You can execute queries towards the movie database from your personal work space (`> psql -h dbpg-ifi-kurs -U username -d username`). The advantage is that in your personal work space you are allowed to create local tables and views.

To access fdb from your personal work space you must send the query as argument to a function named `dblink`. It performs a remote call to the movie database, and the result set is returned to the local work space. Example:

```
select *
from dblink('dbname=fdb host=dbpg-ifi-kurs.uio.no user=ifikurs_ro
password=ifikurs_ro',
$$select count(*) as Ccount
from filmitem
where filmtyp = 'C';$$)
as t1(ccnt integer);
```

The query (parenthesized by double dollar signs - \$\$select count(*) as Ccount from filmitem where filmttype = 'C';\$\$) is then executed in the fdb work space and thus allows the database management system to do local optimizations for making the execution as efficient as possible. Notice that you have to define an alias (here named t1, you can however choose an alias at your own leisure) and supply the alias with the attributes and corresponding data types that dblink is expected to return (here: ccnt which is an integer). Beware that if the result set is large, the streaming of the result set from fdb to your personal work space can take a *very* long time. Therefore queries involving dblink are only recommended if the result set is fairly small.

Movie database segment

To allow you to experiment with the movie database and queries towards it, we have made a segment¹ of the movie database of which you get your private version. The tables in the segment have names similar to those in the movie database. These tables contain essentially all films for which language in the table filmlanguage contains the value 'France'². There are few restrictions on what you are allowed to do with tables in your work space, like changing or deleting table definitions and changing contents, this applies to the French segment as well(!).

It may be wise to first try out more complex queries towards the segment, and then, when you are fairly certain that a query is correctly designed, try it towards the full movie database version - via dblink from your personal work space or directly from the fdb work space. You can in psql change between the two work spaces by issuing the command \c fdb or \c username (depending on the present work space).

¹ The segment contains approximately 8% of the full version. It is fair to assume that the query time consumption is reduced correspondingly, if not more.

² Hommage á Igor Rafienko, David Ranvig and Rune Aske

