

Imperial College of Science, Technology and Medicine  
Department of Computing

Databases (MSc Computing, ISE3, JMC3)

Practical SQL Tutorial

Lab Sessions: Tuesday, 13<sup>th</sup> February 1–3 and Monday, 19<sup>th</sup> February 5–6, both in Rooms 202 and 206.

## 1 Introduction

The objective of this exercise is to give you some practical experience with using the relational database system PostgreSQL.

In this department, there is a single PostgreSQL server, called *kestrel*. This can be accessed through client programs and libraries from all Linux machines. The way to access the server for this exercise will be through the interactive tool **psql**. Your environment variables should be set up in such a way that the only command-line information you need to specify to psql is the database to use. For this exercise, the database is called *films*.

- *Conversion MSc Students:*  
Log into Linux using Exceed. If you have trouble doing so, look at Appendix A.
- Alternatively, use a machine running Linux.
- Type **psql films** to open an interactive query session for this exercise.

## 2 Using psql

You can type SQL queries at the psql prompt (*films=>*). When the query is terminated with a semicolon the result is shown on screen. Lines you type in at the psql prompt can be edited in the same way as Linux shell commands, *i.e.* you can use the arrow keys to edit a line and to recall previously typed lines. **psql** also has a set of commands which start with `\`.

<code>\?</code>	Help on commands
<code>\q</code>	Quit
<code>\e fname</code>	Edit the current query buffer or the contents of file <i>fname</i>
<code>\i fname</code>	Read and execute queries from the file <i>fname</i>
<code>\o fname</code>	Send all query results to <i>fname</i>
<code>\p</code>	Print the current query buffer
<code>\w fname</code>	Write current query buffer to file <i>fname</i>

For the purpose of this exercise, I suggest that you do the following:

- Get used to the environment, typing a few queries.  
Then save your query buffer using `\w file.sql`. Note that this *overwrites* the file — so this ought to be done only once.
- You can then edit the buffer using `\e file.sql`. You can have more than one query in a file.

- If you want to see the result of running the queries in this file, use `\i file.sql`.
- You can write comments in your file `file.sql` using `/* */`.

**Note:** The editor used when opening the query buffer using `\e` is determined by your `EDITOR` environment variable. You should not have to set this; it should default to `edit`. The more enterprising amongst you may want to try using a different editor, see Appendix B.

## 3 The Database

The database `films` contains three tables with information about some films, the casting on those films and the actors who acted in them. This section lists the definition of these tables. A column which is “not null” must contain a value.

### 3.1 Films

This table contains information about films: the title, the director, the country where they were made, the year in which they were made and the running time. Two films may have the same title, so another field (the director) must be used to make a composite key.

```
CREATE TABLE films (  
    title    VARCHAR(40) NOT NULL,  
    director VARCHAR(40) NOT NULL,  
    origin   CHAR(10),           /* Country where the film was made */  
    made    DATE,               /* The year the film was made */  
    length  INTERVAL           /* Running time */  
);
```

### 3.2 Casting

This table contains information about which actors played in which films. An actor’s name can appear in several rows of this table. If the name of the character an actor plays is not known, the value is set to `NULL`.

```
CREATE TABLE casting (  
    name     VARCHAR(40) NOT NULL,  
    title    VARCHAR(40) NOT NULL, /* Title of the film */  
    director VARCHAR(40) NOT NULL,  
    part     VARCHAR(40)           /* Name of character */  
);
```

### 3.3 Actors

This table contains information about actors and their dates of birth. If the date of birth is not known, the value is set to `NULL`.

```
CREATE TABLE actors (  
    name     VARCHAR(40) NOT NULL,  
    born     DATE  
);
```

## 4 Your Task

You should formulate queries in SQL that find out information to answer the following questions.

1. Show the directors and titles of all films, sorted by director in ascending order.
2. Show the titles and running times of all British films, without duplicates.
3. List the cast of Ben Hur.
4. Which actors have been directed in films by both Ridley Scott and Alfred Hitchcock?
5. Which actors appeared in any or all of the films titled 'Alien' or 'Aliens'?  
Show the actors' name and the films' titles.
6. Show all the directors who have either made a British film or who have cast a film titled 'The Terminator'.
7. Which films have been made since 1980 and last 1 hour and 40 mins or more.  
Show the titles, directors, dates, and running times.
8. Which actors appeared in any film as Sigourney Weaver?  
Show the actors' names and the films' titles. Do not show Sigourney Weaver.
9. Show the titles of the films which are longer than at least one of the films made by the director 'Mike Nichols'.

## A Logging into Linux Servers via Exceed from NT

An icon called **Exceed** should be present on your desktop.



Otherwise, use **Start** → **Programs** → **Utilities** → **Exceed** → **Exceed** .

In the “**mchooser**” window which appears after a moment, click on the line starting with “Linux” under **Host Selection** and then click on **Host List ...** . Next, select either **owl** or **emu** and click **accept** .

You will then have to log on to the machine you have selected using your *Linux* password. (You need to hit return after entering both your login and your password to proceed.) After a moment, an **nxterm** window should appear. Use this window to work.

- You can open more windows by typing `nxterm &` at the Linux prompt.
- You can see your directories using `ls`, or `ls -l` to get a listing with more detail.
- You change directory using `cd`.
- You can create a new directory for your SQL work using, e.g. `mkdir sql-tutorial`.

If you need further help with using Linux, ask one of the Lab Helpers.

## B Using psql with Xemacs

Set the editor variable to Xemacs (`setenv EDITOR xemacs`). Then, add the following comment to the end of the file:

```
/** Local Variables: **/  
/** comment-start: "/*" */  
/** comment-end: "*/" */  
/** mode:SQL */  
/** minor-mode: Font */  
/** End: */
```

This should ensure that the file is opened with emacs’ `sql-mode` and that the contents is fontified. If you are working from Exceed, you may find that xemacs complains about some Mod-key re-mapping. If that happens, click onto the window you want to see (the one with your SQL file) and type `Ctrl-X 1`.