Imperial College of Science, Technology and Medicine	University of London
Computer Science (CS) / Software Engineering (SE)	BEng and MEng Examinations Part I
Department of Computing	Integrated Laboratory Course

Laboratory work is a continuously assessed part of the examinations
and is a required part of the degree assessment.
Laboratory work must be handed in for marking by the due date.
Late submissions may not be marked.

Exercise: 15	Working: Individual
Title: SQL Queries	
Issue date: 23rd February 2004	Due date: 1st March 2004
System: Linux	Language: SQL

Aim

- To use the query language SQL to extract information from tables in the relational database system PostgreSQL.

The Problem

- This exercise requires that you use **SQL** to query a database of films. When you have tested thes queries you should write them into the file **queries.sql** which you then submit.
- The database **films** contains three tables with information about some films, the casting and the actors. The definitions of tables are listed here: the names and types of the columns in each table. A column which is 'not null' must have a value. In the case of the current database these fields can be considered as the primary keys for their respective tables.

Films

This contains information about films - their titles, their directors, the country where they were made, the year they were made and the running times. Two films with the same title may exist so another field (the director) must be used with the title 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   /* the time the film lasts  */
);
```

Casting

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

```
CREATE TABLE  casting (
/* Assumes no actor plays more than one part in any film */
    name     VARCHAR(40) NOT NULL,
    title    VARCHAR(40) NOT NULL,  /* the film title */
    director VARCHAR(40) NOT NULL,
    part     VARCHAR(40)            /* who the actor plays */
);
```

Actors

If the date when an actor was born is not known, the value is set to NULL.

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

Submit by Monday 1st March 2004

What to Do

The PostgreSQL sever is set up on a single machine, **db** . Client programs and libraries are installed on all teaching linux machines. Use the interactive client **psql** to run **SQL** queries on any linux machines.

In this case:

```
psql  -U lab films
```

The password is **lab**

Type an SQL query into psql terminating the command with a semicolon. For example

```
SELECT * FROM casting;
```

The result will be shown on the screen.

Lines you type in can be edited in the same way as the Unix shell. After a line is entered and run it is stored in a buffer and can be recoved and edited using the arrow-keys.

Psql Commands

Psql also has a set of commands which start with \. You can see them all by typing \?. You may find the following ones useful.

```
\?                              Help on psql commands
\h                              Help on sql statements
\q                              Quit psql
\e <filename>                   Edit the query buffer or the contents of <filename>
\E <filename>                   Edit the query buffer or the contents of <filename>
                                and run the saved result
\g <filename>                   Run the query input buffer and optionally
                                save the output in <filename>
\i <filename>                   Read queries from <filename> into the query input buffer
\p <filename>                   Print the query buffer
\s <filename>                   Print or save the command line history to <filename>
```

The Queries

1. Show the film titles and directors of all films, sorted by director (1 mark).
2. Show the titles, running times and date of all non American films (1 mark).
3. Which films have been made after 1997 and last two hours or more? Show the titles, directors, years and running times (1 mark).
4. List the cast of Hitchcock's Psycho (or at least those actors in the database) (1 mark).
5. Which actors appeared in one or more films with titles starting with 'D'? Show the actor's name, the character's name and the film's title (1 mark).
6. Which films contain actors born before 1900 ? Show the actor's name, the character's name and the film's title (1 mark).
7. Which other actors appeared in the same films as Bruce Willis? Show the actor's name, the character's name and the film's title (1 mark).
8. How many actors made their last film before 1990 (1 mark) ?
9. Which actors have appeared in more than one film, and what are the films each such actor played in? Show the actor's name and the film's title and director, sorted by actor (1 mark).
10. List any actors that appeared in a film directed by Alfred Hitchcock and also in a film with Ian Holm (1 mark).

Unassessed

- Look at the "http://uk.imdb.com/" internet film database which was used to provide data for this exercise.

- CSG will create special accounts to that will allow you to create and your own database. The accounts will contain a copy of the films database with write permission on the tables. We will send you an email with details about these accounts.

Useful Sources for information about psql

Here are some URLs for some pages on psql

- An Introduction to PostgreSQL and SQL
(http://www.doc.ic.ac.uk/lab/labman/postgresql/)
- Printable (pdf) version of the above.
http://www.doc.ic.ac.uk/lab/labman/postgresql/postgresql.pdf

Submission

Save your SQL queries as a single file called **queries.sql** in the order specified, with a comment marking the start of each query and a semicolon ending each query. The file should not contain the results of the queries. Submit the file using the command **submit 15** at your Linux prompt.

Assessment

Queries 1 - 10	1 (mark each)
Total	10