SQL Warm-up – Brief intro to SQL by examples

The exercises below are intended for experiments with SQL.
To get started you'll need MySQL Workbench. Follow the instructions in the guide "**MySQL Installation"** (for Windows or Mac as appropriate). For the purpose of these exercises you can choose to install Workbench only, but we recommend to install the Community Server as well (two separate procedures on Mac, same procedure on Windows).
The **MySQL Installation** guide also include instructions to make a connection to the **islb.ruc.dk**-server. Follow these.

The database to be target for these exercises is called **movie**. It contains an extract from "The Internet Movie Database" (imdb.com) condensed in 3 tables (**`mid`** and **`pid`** are id's for movies and persons respectively):

```
movie(mid, title, production_year)
casting(mid, pid)
person(pid, name, gender)
```

Open your connection to the islb.ruc.dk-server and start your session with the command "use movie". Write SQL queries to answer the following using the 3 tables **`movie, person`** and **`casting`**.

1) List all titles of movies produced in year 1888
2) List title and year of movies produced in year 2019 or later
3) List title and year of movies produced in the 1930'ies.
   (Hint: you can combine conditions using AND)
4) Find the names of all female actors. (Hint: gender should be "f")
5) Find the title and production year of all movies with a title starting with 007. Hint: rather than "=" you can use the operator "like" and by specifying e.g. "An%" you will match all values that start with "An"
6) Count the number of movies. Hint: rather than selecting an attribute you can simply select using a function "count(*)"
7) Count the number of movies produced in 2004. Hint: use "count(*)" again, but now in a query with a where-clause
8) Find the name and gender of all persons with a name starting with "Mikkelsen"
9) Find the name of the person with 7767 as pid
10) Find all mid's for movies that have casted the person with pid=7767
11) Try the following query and discuss what it retrieves – most importantly, what is the meaning of "natural join"
    ```
    select title from movie natural join casting where pid=7767;
    ```
12) By taking a natural join of all 3 tables you can express queries that involve attributes from all three tables. Express a query that joins all 3 tables using by natural join (you can select all attributes by specifying *) and show only those where name is starting with "Mikkelsen"
13) Find names of actors that have acted (been casted) in movies produced in 1888
14) Find the title of all movies that have casted the actor "Mads Mikkelsen" (take imdb's convention for writing names into consideration).
15) Try now to continue your investigation of the imdb-content. Use what's introduced above to express you own SQL-queries and, if you get inspired, look (google) for other possibilities. One of many useful resources is http://www.w3schools.com/sql/