

## Lab # 1

You will be using MySQL as a database management system during the labs. The goal of this first lab is to familiarize you with MySQL.

The reason you are using MySQL is twofolds. First, MySQL is a very successful open source database system (ver 50 000 downloads a day – see [www.mysql.com](http://www.mysql.com)). It is easy to install and simple to manipulate. But MySQL is a simple and efficient system well suited for this class. You will hate some of its limitations though, like the meaningless error messages. Second, we have good contacts with MySQL because of previous joint research activities. So we can feed any feedback you will have directly to the MySQL developers.

All the documentation you need about MySQL is on the MySQL web site: <http://dev.mysql.com/doc/>. In particular, the searchable reference manual is your friend <http://dev.mysql.com/doc/mysql/en/index.html>. One of the problems with this documentation is that it mixes up features from different versions of the software. So, note the version of MySQL that you are using (4.1 is the version installed at ITU) and be careful to read the right pages on the MySQL docs.

The goal of this first lab is (A) to present you with the environment we will use during the labs and (B) to get you started using this environment in a very simple way. When you are done with the lab we will be able to check that you have performed the tasks that are assigned to you. This way we can be sure that you are ready to go on with the forthcoming labs.

Those of you that have experience using MySQL will be done very quickly.

### A – Environment

You are very much welcome to setup MySQL on your laptop. Follow the instructions at [http://dev.mysql.com/usingmysql/get\\_started.html](http://dev.mysql.com/usingmysql/get_started.html).

The IT department at ITU provides a self-service MySQL access. Follow the instructions at <http://www1.itu.dk/sw114891.asp>.

I assume in the rest of this document that you are using the ITU setup. If you are using a version you installed on your computer, simply replace `mysql.itu.dk` by your server name (probably `localhost`).

In order to launch the MySQL client software, you need access to a shell on the machine on which the MySQL client is installed, i.e., `ssh.itu.dk`:

```
yourPC ~ > ssh ssh.itu.dk
```

The MySQL client software is named *mysql*.

If you simply run it, you will get a error message:

```
ssh.itu.dk ~ > mysql
```

*ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/tmp/bonnet.mysqlsock' (2)*

The reason is that mysql is a client software that tries to open a network connection to a server process. No need to go into the details of network connections. The fact is that the mysql client needs to connect to a server. There is no server on the bach-1 machine, so no connection can be opened and the call to mysql fails. In order to connect to the server, you have to (1) indicate the host on which the server is running, i.e. fp-www – this is done with the `-h` option, (2) your user name, e.g., group1 – this is done with the `-u` option, and (3) for security reason you cannot access the database server without giving a password – this is done using the `-p` option to signal that the server is welcome to ask for your password.

```
ssh.itu.dk ~ > mysql -h mysql.itu.dk -u YourUserName -p
```

*Enter password:*

*Welcome to the MySQL monitor. Commands end with ; or \g.*

*Your MySQL connection id is 12 to server version: 4.1.8*

*Type 'help;' or '\h' for help. Type '\c' to clear the buffer.*

```
mysql>
```

You are now logged in the database server using the mysql client. You can issue commands that will be interpreted by the database server.

The first thing you should do is set your own password. Please choose a password which is not sensitive (not the one you use for your digital signature), but one you will remember. In order to change the password, use the command *set password* with your own password as follows:

```
mysql> set password = password('YOUR OWN PASSWORD');
```

*Query OK, 0 rows affected (0.00 sec)*

Note that the *set password* command is followed by a semicolon (;) if you do not type it, the mysql client does not know that your command is complete. The parser waits for a semicolon

```
mysql> set password = password('YOUR OWN PASSWORD')
```

```
->
```

You need to type a semicolon to complete the command.

Once you have set your password, you can issue commands.

The very first command you should type each time you log into mysql is *use*. This command allows you to specify the database you are going to access. At ITU, this database is created when you register in the self-service (on your own installation, you need to create a database, see <http://dev.mysql.com/doc/refman/5.0/en/create-database.html>):

```
mysql> use db1;  
Database changed
```

Now you can work with the contents of this database. First, let us find out what is the contents of your database. Use the *show* command (show is a very helpful command to inspect the contents of your database – type in *help show*; to find out more details about the *show* command):

```
mysql> show tables;  
Empty set (0.00 sec)
```

Your database is empty, but you know how to access it. You will populate your database in the next part of the lab. Let us first see how you quit the mysql client. Just type quit

```
mysql> quit  
Bye  
bach-1 ~ >
```

## B – Getting Started

You will populate your database in three steps:

1. You will use a script to create a database schema.
2. You will use a script to load data into the database
3. You will type in a query to check that the database is correctly populated

The database is an example of airline database.

To do that you will need to download the following files (from the isis web site):

- airline-schema.sql
- insert-airline.sql

### Step 1

Download the file airline-schema.sql.

This file contains a series of SQL commands that create tables.

Log in the database server using your user name (see A)

Use the command *source* to interpret the contents of the airline-schema.sql file. This is an efficient alternative to typing in commands in an interactive way.

```
mysql> source airline-schema.sql  
Query OK, 0 rows affected (0.00 sec)
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
...
```

```
Query OK, 0 rows affected (0.00 sec)
```

11 tables have been created. Use the show command to display their names:

```
mysql> show tables;
```

```
+-----+
| Tables_in_db1 |
+-----+
| Aircraft      |
| Assigned_To  |
| Booked_On    |
| Can_Fly      |
| Departure    |
| Employee     |
| Equipment    |
| Flight       |
| Person       |
| Pilot        |
| Plane        |
+-----+
```

11 rows in set (0.00 sec)

You can also use the show command to find out the attribute names of those tables:

```
mysql> show columns from Aircraft;
```

```
+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key| Default | Extra |
+-----+-----+-----+-----+-----+
| Serial_No  | smallint(6) |    | PRI | 0       |      |
| Model_No   | varchar(15) |    | PRI |         |      |
+-----+-----+-----+-----+-----+
```

2 rows in set (0.00 sec)

## What are the attribute names for the tables Flight and Plane?

### Step 2

Download the file insert-airline.sql.

Same procedure using the insert-airline.sql file that inserts rows into the tables you have just created.

```
mysql> source insert-airline.sql
```

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

.....

Query OK, 1 row affected (0.00 sec)

You can select all the tuples of a table using the following SQL command:

```
mysql> select * from Aircraft;
```

```
+-----+-----+
| Serial_No | Model_No |
+-----+-----+
|    10    | B747    |
|    11    | B727    |
|    13    | B727    |
|    13    | B747    |
|    21    | DC10    |
|    21    | DC9     |
|    22    | B757    |
|    22    | DC9     |
|    23    | DC9     |
|    24    | DC9     |
|    70    | A310    |
|    80    | A320    |
|    93    | B757    |
+-----+-----+
```

```
13 rows in set (0.00 sec)
```

**How many rows are there in table Flight?**

**What is the earlier arrival time in table Flight?**

### Step 3

The airline database contains information about airplanes, personel (including pilots) as well as flight routes.

The following SQL query finds out the employee number and name of the pilots booked (as passengers) on a flight they are assigned to (as pilots). Ideally, this should not happen in a consistent database. Is it the case for the database you have created? The way to find out is to run the following SQLquery that finds the employee number, name of those employees who are booked and assigned to a same flight. The result of this query should be the empty set:

```
select Employee.Emp_No, Employee.name
from Employee, Booked_On, Assigned_To, Pilot
where Employee.Name = Booked_On.Name
and Employee.Emp_No = Pilot.Emp_No
and Assigned_To.Emp_No = Employee.Emp_No
and Assigned_To.Dep_date = Booked_On.Dep_Date;
```

**What result do you get? Try and find out why it is not the empty set...**