# MySQL Practical

## Part 1: Using the MySQL Command Line Tool

a) Log in to mysql using its (SQL) Command-Line Tool

(**Note for users using their own laptops for this practical**: first SSH into a gateway server in order to get access to the other servers - **ssh ph-cplab.ph.ed.ac.uk using your guest account**.)

Type the following command into your unix shell, replacing the <mysql\_username> with your guest account name (note that visit1 to visit9 should use visit**0**1 to visit**0**9 - that is put a zero prior to the single digit):

mysql -h archerdb1.ph.ed.ac.uk -u <mysql\_username> -p

This will prompt you to enter your password; use the following password:

password: **jxfap3pi**

Once you are connected you should see the “mysql>” prompt.

1a) To change your password type the following command from the MySQL command line tool:

SET PASSWORD = PASSWORD('***cleartext password***');

b) Create a database

For example if your username is visit01:

create database visit01\_testdb;

c) Then switch to the database you have created.

For example,

use visit01\_testdb;

d) Create a table

CREATE TABLE addresses(address\_id INT NOT NULL AUTO\_INCREMENT, number VARCHAR(20), street VARCHAR(100), city VARCHAR(100), PRIMARY KEY(address\_id));

e) To view the table schema, type:

describe addresses;

f) Insert data into the table:

INSERT INTO addresses VALUES(NULL, "1", "Lygon St", "Edinburgh");

INSERT INTO addresses VALUES(NULL, "54", "James St", "Edinburgh");

INSERT INTO addresses VALUES(NULL, "11", "Crammond Rd", "London");

g) View all the data in the table:

select \* from addresses;

## Part 2:

Lets try a SQL JOIN operation between two tables.

a) Create a new table called customers and insert few customers with the corresponding address ID from the above addresses table:

CREATE TABLE customers(customer\_id INT NOT NULL AUTO\_INCREMENT, name VARCHAR(100), DoB DATE, address\_id INT, PRIMARY KEY( customer\_id ));

b)Insert few customers:

INSERT INTO customers VALUES (NULL, "Larry Henderson", "1970-01-29", 3);

INSERT INTO customers VALUES (NULL, "Margaret Jackson", "1950-11-16", 2);

INSERT INTO customers VALUES (NULL, "Tony Smith", "1980-12-11",1);

c)Do a JOIN operation between the customers and addresses tables showing names of customers and their full addresses:

SELECT name, number, street, city FROM customers, addresses WHERE customers.address\_id = addresses.address\_id;

4)How would you list all the customers from Edinburgh with their full addresses?

## Part 3: Accessing the data via a Java application

a) Download the following two files to a directory in your Linux account.

* Download the Java example SqlQueryExecutor.java from the training web site.
* Download the MySQL JDBC driver (mysql-connector-java-5.1.32-bin.jar) from the training web site.

b) Edit the file SqlQueryExecutor.java and at the top of the file edit the fields MYSQL\_USER\_NAME, MYSQL\_PASSWORD, MYSQL\_DATABASE with your values.

Compile the SqlQueryExecutor.java using the following command:

javac SqlQueryExecutor.java

c) Run the SqlQueryExecutor as shown below (type the whole command in one line):

java -classpath mysql-connector-java-5.1.32-bin.jar:. SqlQueryExecutor

## Part 4: A few practice queries in case you have time to spare

a) Select the number of customers in each city

SELECT a.city, count(\*) number\_of\_customers FROM addresses a, customers c WHERE a.address\_id = c.address\_id GROUP BY a.city;

b) Select the oldest customer (for a special treat)

SELECT c.name oldest\_customer, c.DoB FROM customers c WHERE c.DoB = (SELECT min(DoB) FROM customers);

c) Find out whether any customer has their birthday today (for a cake)

SELECT c.name, c.DoB FROM customers c WHERE MONTH(c.DoB) = MONTH(NOW()) AND DAY(c.DoB) = DAY(NOW());

d) Display the names and ages of each customer ordered from youngest to oldest

SELECT c.name, TIMESTAMPDIFF(YEAR, c.DoB, CURDATE()) AS age FROM customers c ORDER BY age;