The objective of this lab practice is to become familiar with simple database operations

1-Design a database

Design a database that uses at least two tables.

Decide which columns must be keys

One example is a closet inventory, for several people and several types of clothes.

2-Basic database usage

2.1 Create a database using MySQL commands or HeidiSQL interface

Here are some example commands to create the tables:

```
CREATE DATABASE closet;
USE closet;
CREATE TABLE person (
    id INT NOT NULL AUTO_INCREMENT,
    name CHAR(60) NOT NULL,
    PRIMARY KEY (id)
) ENGINE=InnoDB;

CREATE TABLE shirt (
    id INT NOT NULL AUTO_INCREMENT,
    style ENUM('t-shirt', 'polo', 'dress') NOT NULL,
    color ENUM('red', 'blue', 'orange', 'white', 'black') NOT NULL,
    owner INT NOT NULL,
    FOREIGN KEY owner REFERENCES person(id) ON DELETE cascade,
    PRIMARY KEY (id)
) ENGINE=InnoDB;
```

(By default ON DELETE is RESTRICT)

2.2 Insert some data in the tables

The typical Insert command is:

```
INSERT INTO person (name) VALUES ('Peter');
```

Retrieve inserted data to check if it is correct:

```
SELECT * from person;
SELECT * from shirt;
```

2.3 Delete data

Try to delete a person and check if his/her clothes are deleted as well:

```
DELETE from person WHERE id=123;
```

Check for cascading delete.

3 Performance improvements

3.1 Load enough data to be able to measure search times

Generate lots of data and load them into one of the tables, for example into table person.

You may use large random numbers as names, created with Matlab or Excel. Save those numbers in a text file and load the file using HeidiSQL or the command:

LOAD DATA INFILE xxx INTO TABLE person FIELDS TERMINATED BY '\t';

3.2 Measure time for executing a search

Make a search by name, using SELECT and measure the time required.

You may use MySQL profiler for the measurements:

SET profiling = 1;

[Your Query]

SHOW PROFILES;

3.3 Improve performance by creating an index

Use HeidiSQL to set the person name as an index in the table

Repeat the previous search and compare time.