Lab 2

CREATE, INSERT, SELECT

# Introduction

The purpose of this lab is to gain practice writing DDL to create a database and to review basic INSERT and SELECT commands.

# Submission

* You may work on this lab in groups of 2
* Submit a single solution for your group on LMS
* You must submit the following files:
  + ddl.sql
    - An SQL/DDL file that creates your database and all its tables
  + data.sql
    - An SQL file that INSERTs all sample data you created for your database
  + queries.sql
    - Your Task 3 queries
* If you have done the lab correctly, the instructor should be able run the following sequence of files using a MySQL DBMS without any errors occurring:
  + ddl.sql (to create the database)
  + data.sql (to populate the database with your sample data)
  + queries.sql (to run your reports)

# Project Description

This is an open-ended lab where you may choose to track information for any simple system using a database. Your database must contain at least 3 tables (try to keep it below 10). Pick appropriate column names and data types, add appropriate constraints to your tables, and use foreign key constraints to relate the tables to one another.

# Lab Instructions

## Task 1 — Write a DDL script [50%]

* Create a file called ddl.sql in which you will **manually** write the necessary DDL to create your database.
  + The file must create both the tables AND the database for a MySQL DBMS
  + The file needs to be able to be run as a whole without errors. (Ie, use appropriate DROP and IF [NOT] EXISTS statements so that the database and its tables are dropped and re-created every time the script is run.)
  + You do NOT need to submit a MySQL Workbench model, but if you find it helpful to create one you may; however, **you may not use the Forward Engineering tool from Workbench to create your ddl.sql file**. You must write the DDL script by hand.
* Your database must have AT LEAST the following\*:
  + 3 tables
  + 2 **named** foreign key constraints
  + A **named** compound unique key
  + A **named** index
  + A check constraint (yes, even though MySQL will ignore it)
  + A field that has a default value
  + A non-nullable field
  + An ENUM field
  + An integer-based field
  + A decimal-based field **that is appropriate for use in precise calculations**
  + A date-based field
  + It must support emoji characters in any text-based fields

\* Be creative and use your imagination, but if you are having a hard time coming up with a ‘reason’ to include some of these aspects in your database then feel free to include fields or constraints that don’t really ‘fit’ into your system. **But be sure to document that this is the case using SQL comments.**

## Task 2 — Insert Sample Data [10%]

* Create a file called data.sql and in this file write insert statements that will fill your database tables with enough useful sample data so that you can create meaningful queries in Task 3

## Task 3 — Write Queries [40%]

* Create a file called queries.sql and in this file write a number of queries to fulfill the requirements below.
* You should include queries that demonstrate the following

1. Get all data from a single table
2. Get data from only specific columns
3. A report that includes a column with calculated values (either using an arithmetic operator, a string function, or a date function)
4. A report that uses the DISTINCT keyword
5. Queries using the following operators in a WHERE clause:
   1. AND and OR in the same clause
   2. LIKE
   3. IN
   4. BETWEEN
6. An ORDER BY clause
7. A LIMIT clause.

* Each query should only refer to a single table (ie, no JOINs)
* One query might fulfill multiple of the above requirements (eg. One query might have an ORDER BY, a LIMIT, and a WHERE clause that uses the LIKE operator, which would fulfill all of 5b, 6, and 7 using one query).
* Please include comments in your queries that describe what the query does AND which of the above requirements the query fulfills.