

# CS360 Introduction to Database – Midterm

## (Guidelines for SQL Programming)

[Note] '\$' means the shell command line.

### Part 1. Connect to Virtual Machine

Step 0) Confirm your information related to Virtual Private Network & Virtual Machine.

You can find your VPN information (ID, PW) and VM information (NAT IP, Username, PW) in '[CS360] VM\_VPN\_info.pdf'. Please check your information. OS of every VM is Ubuntu 18.04.

- We strongly recommend changing your informed password after connecting to the VM.

Step 1) Download VPN Client & Connect to VPN

You can download the manual file 'KCloud-VPN-Tutorial.pdf' from the link below. According to the manual file, download the appropriate VPN client for your OS and connect to the VPN.

- Link:  
<https://kcloud.kaist.ac.kr/index.php/board/?uid=141&mod=document&pageid=1#kboard-document>
- [Note1] If VPN client or config file requires VPN IP or URL, just type <https://192.249.18.254> or <https://kcloudvpn.kaist.ac.kr>.
- [Note2] If you have any problems, especially for MacOS, please update your OS and download client again.

Step 2) Connect to VM

After KCloudVPN connection, connect to VM via Mac terminal, Window terminal, PUTTY , etc., with the command 'ssh'.

```
$ ssh {username}@{NAT IP}
```

- It is OK to follow this instruction instead of the last page ('Connect to VM with pem file') in the manual file.

## Part 2. Connect to MySQL and Load database.

### Step 0) Change your password

We strongly recommend you changing your informed password with the below command.

```
$ passwd
```

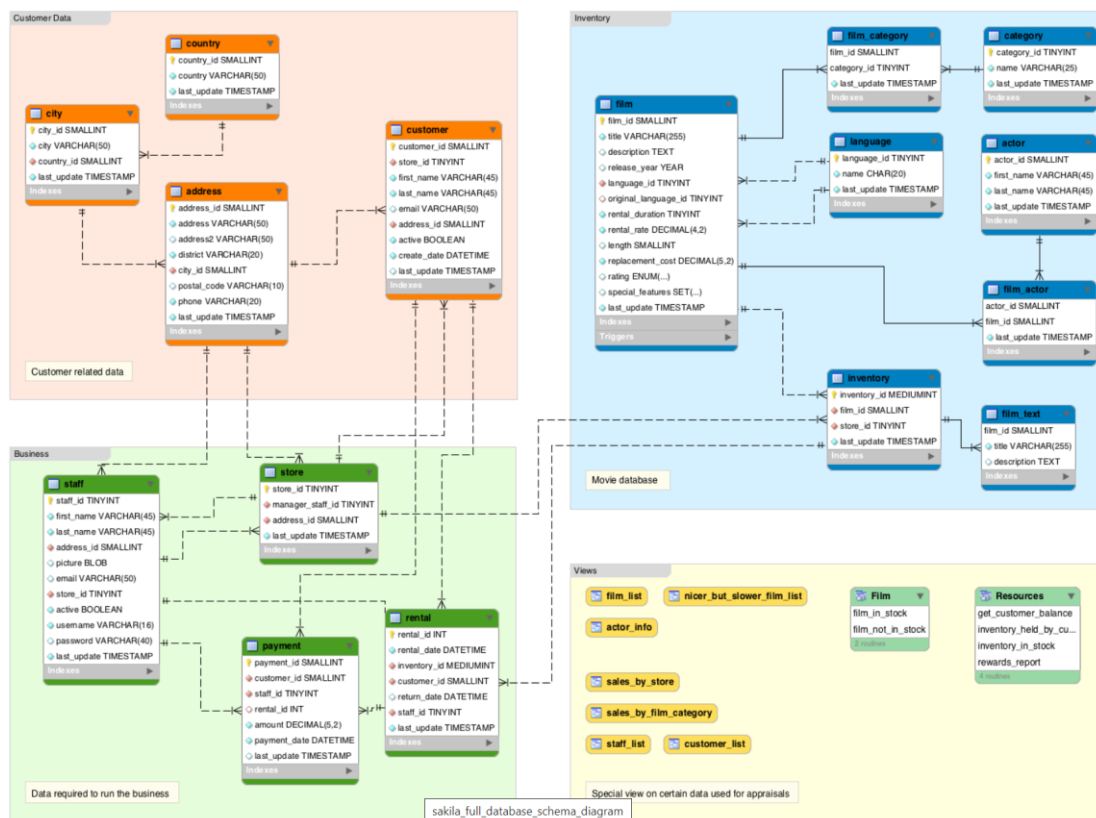
### Step 1) Download the file for creating the database 'sakila' & Extract the files

You can download and extract the files with the commands below.

```
$ cd ~  
$ wget https://downloads.mysql.com/docs/sakila-db.tar.gz  
$ tar -zxvf sakila-db.tar.gz
```

- The figure below shows the schema of the Sakila database. This database has 23 tables: 16 normal tables and 8 view tables. You can use view tables, but we recommend that you use only normal tables. You can find the information related to the schema in the following link:

<https://dev.mysql.com/doc/sakila/en/sakila-structure.html>



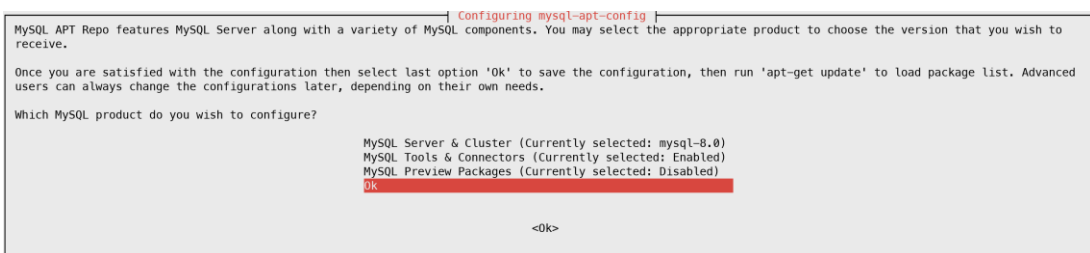
ref: [https://databaseguide/what-is-a-database-schema/sakila full database schema diagram/](https://databaseguide/what-is-a-database-schema/sakila%20full%20database%20schema%20diagram/)

## Step 2) Install MySQL on Ubuntu 18.04

You can install MySQL server 8.0 with the commands below. In this process, the configuration windows appear as shown below. Just select the red parts like the figures.

- Download & Install repository for MySQL

```
$ cd ~  
  
$ wget https://dev.mysql.com/get/mysql-apt-config_0.8.14-1_all.deb  
  
$ dpkg -i mysql-apt-config_0.8.14-1_all.deb
```

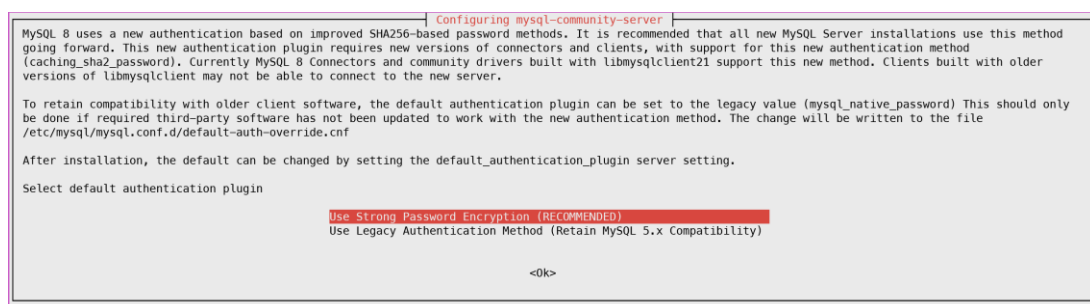
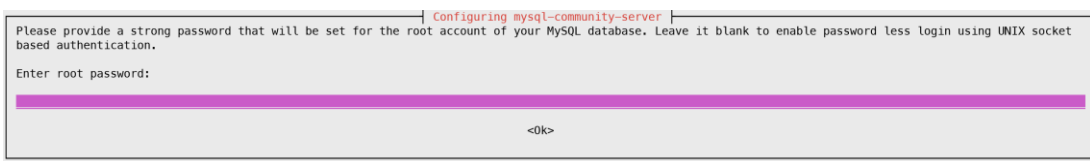


- Update repository

```
$ apt-get update
```

- Install MySQL server

```
$ apt-get install -y mysql-server
```



- (Optional) Automatically restart MySQL on VM reboot

```
$ systemctl enable mysql
```

### Step 3) Connect to MySQL

If you have completed all of the above, you will be able to connect to MySQL with the command below. Check your version (red underlined in the below figure).

```
$ mysql -u root -p
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.24 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> |
```

### Step 4) Load the database 'sakila'

- Create database

```
mysql> CREATE DATABASE sakila;
```

- Create database structure

```
mysql> SOURCE ~/sakila-db/sakila-schema.sql;
```

- Insert data

```
mysql> SOURCE ~/sakila-db/sakila-data.sql;
```

- Check tables

```
mysql> USE sakila;
```

```
mysql> SHOW TABLES;
```

Tables_in_sakila
actor
actor_info
address
category
city
country
customer
customer_list
film
film_actor
film_category
film_list
film_text
inventory
language
nicer_but_slower_film_list
payment
rental
sales_by_film_category
sales_by_store
staff
staff_list
store

[Note1] After connecting to MySQL, for querying, you have to select a database once with the command 'USE' on MySQL command line. However, you can select a database at the same time as you connect with the command below.

```
$ mysql sakila -u root -p
mysql>
```

[Note2] You can disconnect to MySQL with the command 'exit' in the MySQL command line.

```
mysql> exit;
$
```

## Part 3. Execute queries

There are several ways to query like the below. The ways not covered below are also fine, such as accessing the database remotely using MySQL Workbench on your PC. However, no matter how you write and execute queries, query files (.sql) and result files (.csv) MUST be submitted.

### Method 1 Just type SQL on MySQL command line

```
mysql> SELECT COUNT(*) FROM customer;
```

### Method 2 Write SQL file, and Execute the SQL file in MySQL command line

You can write SQL files with editors, such as vim, nano, atom, vscode, etc, on the terminal. After you write the SQL file like the below, the query can be executed with the command 'SOURCE' on MySQL command line.

Example: 'test.sql'

```
SELECT COUNT(*)
FROM customer
~
-- INSERT --          2,14      All
```

```
mysql> SOURCE test.sql;
```

- If you are not familiar with terminal, we recommend using the vim editor. You can use the vim with the command vi (or vim). Following link would be helpful to use vim:  
<https://coderwall.com/p/adv71w/basic-vim-commands-for-getting-started>

```
$ cd ~
```

```
$ vi {file_name}.sql
```

### Method 3 Write SQL file, and Execute the SQL file in shell command line

After writing SQL files like the Method 2, you can also execute the query with the command below on the shell command line.

```
$ mysql sakila -u root -p < test.sql
```

[Note] All of the above methods produce the same result.

```
+-----+
| COUNT(*) |
+-----+
|      599 |
+-----+
```

## Part 4. Submit files

You MUST submit the query file (.sql) and the result file (.csv) for each problem. You will only get points if these two files are submitted for each problem. The name of SQL file and CSV file for each problem MUST be 'problem' + Problem Number. For example, your submission files for Problem 1 should be 'problem1.sql' and 'problem1.csv', respectively.

The result files (.csv) can be obtained with the MySQL command 'INTO OUTFILE' as 'Step 0'. Keep in mind that you MUST submit the results obtained through MySQL command 'INTO OUTFILE'. If the format of the output files is different (e.g., terminators for field and line are different, or not csv file), you may not get desirable points.

### Step 0) Output query execution result to file

- '/var/lib/mysql-files/' should always be used for the output directory path (Otherwise, an authority-related error will occur). It is recommended that you visually check an output of query as much as possible and output the result to a file because the file cannot be overwritten automatically. If you want to change the output file, first use the command 'rm' in the shell command line to delete the file (e.g., \$ rm /var/lib/mysql-files/problem0.csv') and execute the query again.

```
mysql> SELECT COUNT(*) FROM customer  
  
      INTO OUTFILE '/var/lib/mysql-files/problem0.csv'  
      FIELDS TERMINATED BY '|'   
      LINES TERMINATED BY '\n';
```

### Step 1) Make the file (.tar.gz) for submission

- Create a directory

```
$ cd ~  
$ mkdir {studentID}
```

- Move result files (.csv) (One-by-One)

```
$ mv /var/lib/mysql-files/{filename}.csv ~/ {studentID}
```

- Move query files (.sql) (One-by-One)

```
$ mv ~/ {filename}.sql ~/ {studentID}
```

- Check your directory with the command 'ls'. If you have written the queries and obtained the results for all problems, you can see the same result as the figure below.

```
$ ls ~/ {studentID}
```

```
problem1.csv  problem2.sql  problem4.csv  problem5.sql  problem7.csv  problem8.sql
problem1.sql  problem3.csv  problem4.sql  problem6.csv  problem7.sql  problem9.csv
problem2.csv  problem3.sql  problem5.csv  problem6.sql  problem8.csv  problem9.sql
```

- Compress your directory

```
$ cd ~
```

```
$ tar -zvcf {studentID}.tar.gz {studentID}
```

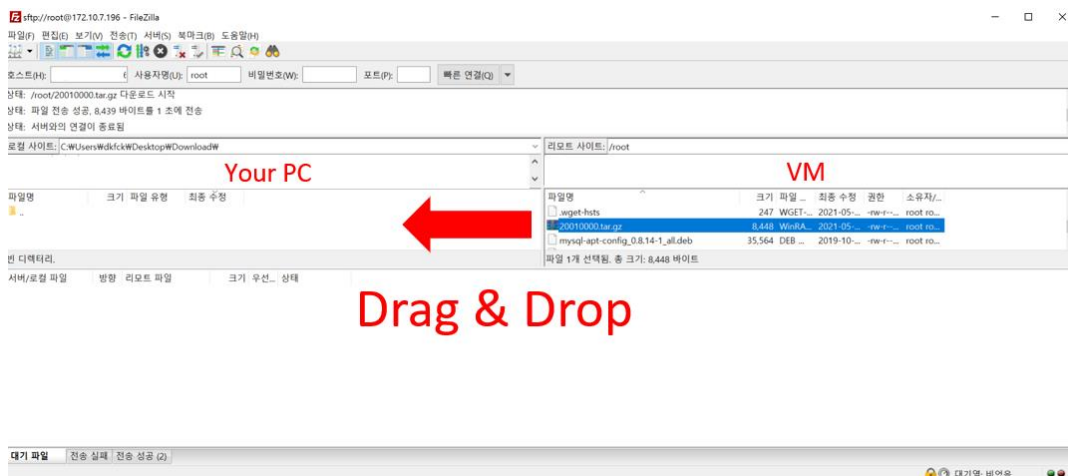
## Step 2) Transfer the file in the VM to your PC

In this guide, we use the software 'FileZilla' to transfer the file. If you have another familiar way to transfer the file in the remote server, you can just use it.

- Download the FileZilla: [https://filezilla-project.org/download.php?show\\_all=1](https://filezilla-project.org/download.php?show_all=1)
- Connect to VM with FileZilla



- Transfer the file



## Step 3) Upload your file (.tar.gz) on KLMS