Introduction to Database Systems

Individual Homework 1: SQL tasks in MySQL

1. Introduction

In this homework you need to practice some basic usages of MySQL, including creating databases, creating tables, loading csv files, loading SQL files, and using MySQL command to find the answer of tasks. After this homework, you will be capable of querying and analyzing your data by MySQL from zero to one.

There will be two datasets for this homework. The first one is COVID-19 data in South Korea, and the second one is European Soccer Dataset, both of these are downloaded from Kaggle Dataset (feel free to google them).

For the first dataset, you need to create a database based on our setting, and load the csv files into your created database. There are 6 easier questions you need to solve by SQL. For the second dataset, you need to load our sql file directly. There are 6 advanced problems you will meet. Read the following content for more details.

2. Tasks

◈ Part 1 - Data Science for COVID-19 in South Korea

In this part, you need to create tables based on the provided DB schema, and load csv files into the database.

A. Create Tables

First, download the COVID-19 data from here.

You can refer to this page to see the meaning of the columns.

Then you should create tables based on the following setting. Notice that you **must** make the detail of your tables the same as our description, including 'table name', 'attribute name', 'attribute type', 'primary key', 'foreign key', 'null'.

Please **paste the screenshot of your tables by using the `describe` command to your report**, it will take 5% of your grades in this homework.

Table Name	Attribute Name	Type	Primary Key	Foreign Key	NULL
patient_info	patient_id	varchar(10)	YES		NO
	sex	varchar(10)			
	age	int			

	province	varchar(20)			
	city	varchar(20)			
	infection_case	varchar(100)			
search_trend	date	date	YES		NO
	cold	float			
	flu	float			
	pneumonia	float			
	coronavirus	float			
time	date	date	YES		NO
	test	int			
	negative	int			
	confirmed	int			
	released	int			
	deceased	int			
time_age	date	date	YES		NO
	age	int	YES		NO
	confirmed	int			
	deceased	int			
time_gender	date	date	YES		NO
	sex	varchar(10)	YES		NO
	confirmed	int			
	deceased	int			
time_province	date	date	YES		NO
	province	varchar(20)	YES		NO
	confirmed	int			
	released	int			
	deceased	int			

region	code	int	YES		NO
	province	varchar(20)			
	city	varchar(20)			
	elementary_school_count	int			
	kindergarten_count	int			
	university_count	int			
	elderly_population_ratio	float			
	elderly_alone_ratio	float			
	nursing_home_count	int			
weather	code	int	YES	region (code)	NO
	date	date	YES		NO
	avg_temp	float			
	most_wind_direction	int			
	avg_relative_humidity	float			

Please **answer the following question in your report**, it will take 10% of this homework.

1. (3%) What is the difference between type “char” and type “varchar”?
   譯:變數型態 “char” 和 “varchar” 有什麼不同？
2. (3%) How many bytes it should take for “tinyint”, “smallint”, “mediumint”, “int”? (e.g. 8 bytes for “bigint”)
   And what’s the range they can express? (e.g. from -1000 to 1000)
   譯:“tinyint”, “smallint”, “mediumint”, “int” 各需要多少bytes來儲存？(e.g. 8 bytes for “bigint”)
   還有他們的表示範圍可以從哪裡到哪裡？(e.g. from -1000 to 1000)
3. (4%) What do you think about this DB schema? If you can change this table architecture, how would you modify it and why?
   譯:你對這資料庫架構有什麼想法？如果你可以修改這架構, 你會怎麼改？為什麼？

B. Load CSV Data

After creating the database, you need to load the downloaded csv files into your database.

Here we don't restrict the method you use, but you have to check the data is loaded successfully by yourself. The following number is the data records for each table.

Table Name	# of Data Records
patient_info	5164
search_trend	1642
time	163
time_age	1089
time_gender	242
time_province	2771
region	243
weather	26271

C. Query Tasks

In this part, here are 6 query tasks you need to write. **Please read the following rules carefully**.

You are only allowed to use **one** query (**one delimiter**) to find the answer, and you **don't** have to explain your SQL. Noted that the **column names** of your query answers should be **the same as our examples**.

For homework submission, please write every query task into a single `sql` file, named as "1.sql", "2.sql", etc.

1. (5%) How many days have the word "cold" been searched over 2000 times in one day?

   譯:請找出有多少天, 一天內的 "cold" 搜尋次數大於2000次。

cnt
5566

2. (5%) How many ways are there for those men under 30, living in Seoul Gangnam-gu being infected? List out in the alphabetical order as the following example.

   譯:找出所有不到三十歲且住在首爾(Seoul)江南區(Gangnam-gu)的男性有因為哪些方式感染covid-19, 並依照字典順序排序, 如下方範例。

infection_case
eating
talking
sleeping

3. (5%) Find out the province, city, and the elementary school count in which the elementary school count is the top three most **and the name of the province is different from the city**. List out in decreasing order of the count.
譯:找出各個區域中, **province**跟**city**名稱不同, 且擁有小學數量為前三高的區域及數量, 並從數量最多到第三多的方式排序。

province	city	cnt
Apple	Taipei	5566
Banana	Hsinchu	3344
Cherry	Taichung	1122

4. (5%) Find out the provinces whose days count of the average relative humidity larger than 70 are the top three most in May of 2016, and list their days count in decreasing order.
譯:請找出2016年5月中, 相對濕度(avg_relative_humidity)大於70前三多天的省(province), 並列出他的天數, 從最多天到最少天排序。

province	cnt
Apple	30
Banana	20
Cherry	10

5. (5%) Find out the province where the elderly population ratio is larger than the average and the date with maximum confirmed in one day. List out in the order of date increasingly. **Notice that the average of the elderly population ratio is the average of those provinces which have the same name as the city.**
譯:找出老年人口比例(elderly_population_ratio)超過平均的省(province)中, 有最大單日確診人數的日期。省的老年人口平均請選擇**province**跟**city**一樣的直接做平均, 不用擔心人口數量不同的問題。

province	date
Apple	2020-01-01

Banana	2020-02-02

6. (5%) How many “accumulated-confirmed”, “added-confirmed”, “accumulated-dead”, “added-dead” are there while the search number of the word “coronavirus” is larger than two standard deviations? List your answer in ascending order by date and round coronavirus to second decimal place.
   The standard deviation should be calculated by the period from 2019-12-15 to 2020-06-29.
   The added-confirmed and the added-dead should be calculated by “the accumulated count of that day minus the accumulated count of the previous day“.
   譯:coronavirus 的搜尋次數大於其平均兩個標準差時, 該天檢測陽性人數(累積)、檢測陽性增加幅度、死亡人數(累積)、死亡人數增加幅度各有多少人？依照時間由小到大排序, 並將coronavirus四捨五入到小數點後第二位
   (coronavirus的標準差請利用2019-12-25至2020-06-29的數值來計算)
   (增加幅度請用該天累積人數減去前一天的累積人數)

date	coronavirus	confirmed_accumulate	confirmed_add	dead_accumulate	dead_add
2020-01-01	66.00	3	0	12	0
2020-01-02	55.12	23	20	34	22

◈ Part 2 - European Soccer Database

In this Part, instead of creating tables and loading csv files, you need to load the provided DB file directly. Please download the file from here.

D. Load SQL File

Here we provide simple steps for the Linux environment. You can also load the SQL file by other methods, like execute the sql file using the “source” command.

1. Firstly, create a database
2. Then, back to your shell and enter command
   `mysql -u {user_name} -p {DB Name} < hw1_part2.sql`
   (You can google “IO Redirection” for more detail of the above mechanism)

E. Query Tasks

In this part, you are also only allowed to use **one** query (**one delimiter**) to find the answer, and you **don’t** have to explain your SQL, **except task 11 and task 12**.
Noted that the **column names** of your query answers should be **the same as our examples**.

For **task 11 and 12**, take **screenshots** of your queries, and **write your analysis** into the report. Try to explain what're your queries doing, why you write these queries, what's the meaning of the result, what's your conclusion, etc.
For submission, also write every query task into a single `.sql` file, named as "7.sql", "8.sql" …, "11.sql", "12.sql"
For the meaning of each table and column in part2, please refer to this page.

7. (10%) List the average long_shots score(**round to the second decimal place**) of the players who had participated in the Italy Serie A league during 2015/2016 season with respect to the preferred foot. You should calculate the long_shots score by the newest data of each player.
   譯:分別列出在2015/2016賽季(season)中，義大利甲級聯賽(Italy Serie A)不同慣用腳(preferred_foot)的選手的平均遠射分數(long_shots)（取該選手最新的測量紀錄)，四捨五入到小數點後第二位。

preferred_foot	avg_long_shots
left	30.87
right	20.87

8. (10%) During the 2015/2016 season, for each of the leagues, if we have known that the average height of members in one team is over 180, what is the probability that the team can win?
   The numerator is the winning count of those over-180-teams, and the denominator is the count of those over-180-teams. Round the probability to fourth decimal place. List out in the alphabetical order.
   (e.g. In two matches, A and B, one team of A is over-180-teams and wins the game. Both team of B are over-180-teams and one of the team win the game, then the win probability is ⅔)
   (e.g. In two matches, A and B, one team of A is over-180-teams and ties the game. Both team of B are not over-180-teams, then the win probability is 0)
   (You need to calculate the record of the same team in different matches. For example, team a1 and team a2 take match A, and team a1 and team b1 take match B, then you need to consider a1 multiple times)
   在2015/2016的季賽中，對於各聯賽，已知某隊伍隊員平均身高大於180，求該隊獲勝概率為何？
   (分子:獲勝隊伍為平均身高大於180隊伍次數，分母:平均身高大於180的隊伍次數，四捨五入至小數點後第4位)
   (請依照聯賽名稱字典順序輸出)
   (例:A, B兩場賽事中，A的其中一隊平均身高大於180且贏了場次，B的兩隊

平均身高皆大於180且其中一隊贏了場次, 得獲勝概率為2/3。）
（例:A, B兩場賽事中, A的其中一隊平均身高大於180且平手, B兩隊平均身高皆不足180, 得獲勝概率為0）
（需重複計算相同隊伍在不同賽事的紀錄。A場為a1與a2兩隊比賽, B場為a1與b1兩隊比賽, 若a1平均身高大於180, 需重複計算）

name	prob
AppleLeague	0.5566
BananaLeague	0.3344

9. (10%) The “win point” can be calculated by the following rule: “For each match, the winning team will get two points. The loser will get zero point. If the match is a draw, both of the teams will get one point. The win points of each team is the point divided by the match count the team participating in during the whole season.” The top five teams with the highest win points are called “the greats of the season”. Find out the average winning score (**round to the second decimal place**) and the team's long name of the greats of the season during the 2015/2016 season. List out in the decreasing order of their win points.
我們定義一種勝利分數, 計算方式為:「對該賽季每一場比賽, 獲勝的那一隊得兩分, 輸的隊伍得零分, 若平手的話兩隊各得一分。每一隊的勝利分數即為其得到的分數除以其該賽季參加過的比賽場數」。勝利分數最高的前五隊稱為年度強權。請找出2015/2016年賽季的年度強權, 參加的每一場比賽平均可以贏對手幾分(這裡的分是指比賽的分數)(四捨五入到小數點後第二位)？請由「勝利分數」最高至第五的方式排序。

team_long_name	avg_win_score
AppleTeam	5.55
BananaTeam	4.44
CherryTeam	3.33
JellyTeam	2.22
OreoTeam	1.11

10. (15%) We call it a landslide victory if there is a “larger than or equal to five” score gap between two teams in a match. And we call it an upset if any one of the sports betting companies has a higher betting odds on a team with landslide victory. Find out that for an upset, what is the average age of the player at that time and the average rating of the players of each team from the previous six months? Round the score to second decimal place and list out in the order of

match id increasingly.
(If a player has scores multiple times, please average all of them)
(Here we want to see the average data of players on the home side and away side in the upset matches. You don't need to consider which team the team belongs to)
當兩隊比賽時, 最後的得分數相差大於等於5, 我們說這樣叫做大贏，在某隊大贏的時候, 任一家賭商開的賠率較輸的隊伍高時, 我們說這場比賽爆冷門。試問爆冷門賽事的兩隊,「當時」各隊選手平均年紀與「當時算起前六個月內（不含）」的各隊選手平均整體分數各是多少？請四捨五入至小數點後第二位, 並依照賽事id由小到大排序。
(若某選手有多次分數時, 請將他們全部都參與平均)
(這邊我們希望看到在爆冷門的賽事中, 主場方與客場方的一些選手平均數據, 不用再考慮該隊伍屬於哪隊)

id	home_player_avg_age	away_player_avg_age	home_player_avg_rating	away_player_avg_rating
1	22.22	23.23	60.60	58.58
2	23.23	24.24	62.62	75.75
3	24.24	21.21	55.55	99.99

11. (10%) Do the home team with home advantage has much more opportunity to win the game, or the team with a higher average score(which can be one of the overall_rating, dribbling, strength, interceptions, or the average of the four scores, calculated by the latest attribute before the player participating in the match) of the whole team players? **Answer by your own view with one SQL query.**
比賽通常都會有所謂的「主場優勢」, 但是主場優勢也只是優勢, 不能保證為隊伍帶來勝利。你認為擁有主場優勢的隊伍比較容易贏, 還是隊伍選手依照「其參加比賽前最後一次測量attribute」的平均程度較高的隊伍比較容易贏？(此題為開放式答案, 請用一個SQL找出的結果闡述你的觀點。程度可以是整體分數、運球分數、強度分數、截攔分數四選一, 或是將這四種分數平均當作這個選手當時的程度)

12. (10%) You are a gambler of sport lottery. Analyzing this dataset with SQL and finding out the better way to place a bet. **You can answer by your own view with multiple SQL queries. Focus on observation to the dataset and explain your analysis.**
你是一名運彩賭徒, 請利用SQL分析這份資料, 提出什麼情況下進行怎樣的下注會是比較建議的(此題為開放式答案, 可以使用多個SQL, 重點請著重在對資料的分析與發想)

3. Grading

TA will run your ".sql" file automatically. The environment will be **Ubuntu 20.04** and **MySQL 8.0.23**, which is the same as the environment you had prepared in Hw0. Make sure your code can run in this environment correctly.
Following is the grading policy.

Plagiarism is not allowed! You will get a huge **penalty** if we find that.

Description	Score(%)
Part1 table screenshots	5
Question answering	10 (bonus)
SQL tasks	95
Total	100 + 10

4. Discussion

TAs had opened a channel **HW1 討論區** on New E3 forum of the course, you can post questions about the homework on the forum. TAs will answer questions as soon as possible.

Discussion rules:

1. Do not ask for the answer to the homework.
2. Check if someone has asked the same question before asking.
3. We encourage you to answer other students' questions, but again, do not give the answer of the homework. Reply the messages to answer questions.
4. Since we have this discussion forum, do not send email to ask questions about the homework unless the questions are personal and you do not want to ask publicly.

5. Submission

1. The deadline of this homework is **4/14 (Wed.) 23:55:00**.
2. You should put your `pdf` and `sql` files into one folder, each should be named as "**HW1_XXXXXXX.pdf**", "1.sql", "2.sql" …. And the folder should be named as "**HW1_XXXXXXX**" where XXXXXXX is your student ID.

   Then compress your folder into one `zip` file. Submit it to New E3 System with the format **HW1_XXXXXXX.zip** where XXXXXXX is your student ID.

   We **only accept one zip file**, wrong format or naming format cause -10 points to your score (after considering late submission penalty).
3. Late submission lead to score of (original score)*0.85^{days}, for example, if you submit your homework right after the deadline, you'll get (original score)*0.85 points.
4. If there is anything you are not sure about submission, ask in the discussion forum.