

Individual Homework 1: SQL tasks in MySQL

0816160 翁垿豪

A. Part1 table screenshots

```
mysql> describe patient_info;
```

Field	Type	Null	Key	Default	Extra
patient_id	varchar(10)	NO	PRI	NULL	
sex	varchar(10)	YES		NULL	
age	int	YES		NULL	
province	varchar(20)	YES		NULL	
city	varchar(20)	YES		NULL	
infection_case	varchar(100)	YES		NULL	

```
mysql> describe search_trend;
```

Field	Type	Null	Key	Default	Extra
date	date	NO	PRI	NULL	
cold	float	YES		NULL	
flu	float	YES		NULL	
pneumonia	float	YES		NULL	
coronavirus	float	YES		NULL	

```
mysql> describe time;
```

Field	Type	Null	Key	Default	Extra
date	date	NO	PRI	NULL	
test	int	YES		NULL	
negative	int	YES		NULL	
confirmed	int	YES		NULL	
released	int	YES		NULL	
deceased	int	YES		NULL	

```
mysql> describe time_age;
```

Field	Type	Null	Key	Default	Extra
date	date	NO	PRI	NULL	
age	int	NO	PRI	NULL	
confirmed	int	YES		NULL	
deceased	int	YES		NULL	

```
mysql> describe time_gender;
```

Field	Type	Null	Key	Default	Extra
date	date	NO	PRI	NULL	
sex	varchar(10)	NO	PRI	NULL	
confirmed	int	YES		NULL	
deceased	int	YES		NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> describe time_province;
```

Field	Type	Null	Key	Default	Extra
date	date	NO	PRI	NULL	
province	varchar(20)	NO	PRI	NULL	
confirmed	int	YES		NULL	
released	int	YES		NULL	
deceased	int	YES		NULL	

```
5 rows in set (0.00 sec)
```

```
mysql> describe region;
```

Field	Type	Null	Key	Default	Extra
code	int	NO	PRI	NULL	
province	varchar(20)	YES		NULL	
city	varchar(20)	YES		NULL	
elementary_school_count	int	YES		NULL	
kindergarten_count	int	YES		NULL	
university_count	int	YES		NULL	
elderly_population_ratio	float	YES		NULL	
elderly_alone_ratio	float	YES		NULL	
nursing_home_count	int	YES		NULL	

```
9 rows in set (0.00 sec)
```

```
mysql> describe weather;
```

Field	Type	Null	Key	Default	Extra
code	int	NO	PRI	NULL	
date	date	NO	PRI	NULL	
avg_temp	float	YES		NULL	
most_wind_direction	int	YES		NULL	
avg_relative_humidity	float	YES		NULL	

```
5 rows in set (0.00 sec)
```

Question answering

1. 變數型態 “char” 和 “varchar” 有什麼不同？

char 採用固定長度的儲存方式，在資料儲存時，系統會為其分配最大儲存空間。而 varchar 則是可變長度，設定時我們只是設定它的最大值，實際它會依照所需長度分配儲存空間，可減少空間消耗。

2. “tinyint” , “smallint” , “mediumint” , “int” 各需要多少 bytes 來儲存？

- Tinyint：能表示 2^8 個值，需要 1byte。
- Smallint：能表示 2^{16} 個值，需要 2bytes。
- Mediumint：能表示 2^{24} 個值，需要 3bytes。
- Int：能表示 2^{32} 個值，需要 4bytes。

3. 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？

我認為 time_age 和 time_gender 其欄位 confirmed 和 deceased 並無法被有效統計利用，其缺少可辨識其所屬的欄位。若修改架構，我會選擇在 patient_info 表格新增 date，刪除 time_age 和 time_gender 會讓架構比較整理與統計。

SQL tasks

8. 在求取隊伍隊員平均身高時，某些隊伍的某幾位隊員為 Null，我使用 IF 和 IFNULL 計算各隊伍隊員的平均身高和隊伍隊員人數，使得某些不滿 11 人的隊伍也能準確求出其平均。

10. 同樣對隊員的 Null 做了過濾。先計算隊員們個人的整體分數後，再對所有人求平均，以讓每個人在計算隊伍平均時的占比合理。

11. 在此我對所有比賽，取其隊伍前五名球員末次測量的 overall_rating 為平均程度參考，來計算 last_prob，再對比上各場比賽，主場隊伍的勝率。由結果可得知：相較於主場隊伍，平均程度較高的隊伍比較容易贏。

```
mysql> source 11.sql
+-----+-----+
| home_prob | last_prob |
+-----+-----+
| 0.45870000 | 0.4998 |
+-----+-----+
```

12. 我以對 B365 下注為例，用 SQL 判斷我該不該對該主場隊伍下注。

首先，計算隊伍平均整體分數較競爭隊伍較高的隊伍在己方主場的勝率、平手率、敗率。分別為 0.5835、0.2350、0.1815。

```
+-----+ +-----+ +-----+
| Win_prob | | Tie_prob | | Loss_prob |
+-----+ +-----+ +-----+
| 0.5835 | | 0.2350 | | 0.1815 |
+-----+ +-----+ +-----+
```

最後用 $(\text{勝率} \times (\text{B365H}-1) + (1-\text{勝率}) \times (-1))$ 做期望值求得總和為正。

```
+-----+
| Expected_value |
+-----+
| 167.73592205590032 |
+-----+
```

證明在此條件下多次下注，最終可有正面收益。