

## Part1

### Patient\_info

```
mysql> describe patient_info;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| patient_id | varchar(10) | NO   | PRI | NULL    |       |
| sex        | varchar(10) | YES  |     | NULL    |       |
| age        | int(11)    | YES  |     | NULL    |       |
| province   | varchar(20) | YES  |     | NULL    |       |
| city       | varchar(20) | YES  |     | NULL    |       |
| infection_case | varchar(100) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

### Search\_trend

```
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    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

### Time

```
mysql> describe time;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| date       | date      | NO   | PRI | NULL    |       |
| test       | int(11)   | YES  |     | NULL    |       |
| negative    | int(11)   | YES  |     | NULL    |       |
| confirmed   | int(11)   | YES  |     | NULL    |       |
| released    | int(11)   | YES  |     | NULL    |       |
| deceased    | int(11)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

## Time\_age

```
mysql> describe time_age;
```

Field	Type	Null	Key	Default	Extra
date	date	NO	PRI	NULL	
age	int(11)	NO	PRI	NULL	
confirmed	int(11)	YES		NULL	
deceased	int(11)	YES		NULL	

4 rows in set (0.00 sec)

## Time\_gender

```
mysql> describe time_gender;
```

Field	Type	Null	Key	Default	Extra
date	date	NO	PRI	NULL	
sex	varchar(10)	NO	PRI	NULL	
confirmed	int(11)	YES		NULL	
deceased	int(11)	YES		NULL	

4 rows in set (0.00 sec)

## Time\_province

```
mysql> describe time_province;
```

Field	Type	Null	Key	Default	Extra
date	date	NO	PRI	NULL	
province	varchar(10)	NO	PRI	NULL	
confirmed	int(11)	YES		NULL	
released	int(11)	YES		NULL	
deceased	int(11)	YES		NULL	

5 rows in set (0.00 sec)

## Region

```
mysql> describe region;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code | int(11) | NO | PRI | NULL | |
| province | varchar(20) | YES | | NULL | |
| city | varchar(20) | YES | | NULL | |
| elementary_school_count | int(11) | YES | | NULL | |
| kindergarten_count | int(11) | YES | | NULL | |
| university_count | int(11) | YES | | NULL | |
| elderly_population_ratio | float | YES | | NULL | |
| elderly_alone_ratio | float | YES | | NULL | |
| nursing_home_count | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

## Weather

```
mysql> describe weather;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code | int(11) | NO | PRI | NULL | |
| date | date | NO | PRI | NULL | |
| avg_temp | float | YES | | NULL | |
| most_wind_direction | int(11) | YES | | NULL | |
| avg_relative_humidity | float | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

### 1. Difference between char and varchar

	Char	varchar
動態或靜態	固定大小字串資料，以靜態方式儲存，欄位內不足宣告的長度會以空白的方式填滿	可變大小字串資料，以動態方式儲存
儲存資料大小	Char(n)，n 必須是 1 到 8000 的值	Varchar(n)，n 可以是 1 到 8000 之間的值。max 表示儲存體大小上限， $2^{31}-1$ 個位元組(2 GB)
CPU 計算時間	花費較少的 CPU 計算時間	花費較多的 CPU 計算時間

取資料時差異	取資料時·要用 trim() 去掉多餘空格	儲存資料時長度已經隨之改變
儲存差異	take 1 byte for each character	take 1 byte for each character and some extra bytes for holding length information

## 2. Bytes and range

Tinyint	1 Byte	有符號的範圍是-128 - 127，無符號的範圍是從 0 ~ 255
Smallint	2 Bytes	有符號的範圍是-2 <sup>15</sup> (-32,768) ~ 2 <sup>15</sup> -1 (32,767)，無符號的範圍是 0 到 65535
Mediumint	3 Bytes	有符號的範圍是 -2 <sup>12</sup> (-8388608) 到 2 <sup>12</sup> (8388607)，無符號的範圍是 0 到 2 <sup>24</sup> (16777215)
int	4 Bytes	-2 <sup>31</sup> (-2,147,483,648) ~ 2 <sup>31</sup> -1 (2,147,483,647)

第十一題：

```
mysql> source 11.sql;
+-----+-----+-----+-----+-----+-----+-----+
| homewin_homesc | homewin_awaysc | awaywin_homesc | awaywin_awaysc | home_win_times | tletimes | home_loss_times |
+-----+-----+-----+-----+-----+-----+-----+
|      18343.00 |      18867.09 |      18003.52 |      16859.63 |           11917 |          6596 |           7466 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (3.64 sec)
```

可以看出主場優勢較能為隊伍帶來勝利，我所做的表格分別為，主隊贏時主隊整體分數，主隊贏時客隊整體分數，客隊贏時主隊整體分數，客隊贏時客隊整體分數，後面三個則為主隊贏的次數，平手次數，以及客隊贏的次數，可以發現主隊贏的次數較多，但是整體分數卻是輸給客隊的整體分數，另外，客隊贏的時候客隊的整體分數也比主隊的整體分數小，由此可知主場優勢較能帶來勝利

在 code 的部分，前四筆都是做相同的東西，where 給的條件是要選取比賽前的時間，然後在 pr 的 query 中得出最後一次測量的時間，最後在主 query 中讓全部的分數加起來看整體分數，後三筆資料單純計算贏的次數

第十二題：

team_long_name	win_rate
KRC Genk	0.46
Beerschot AC	0.26
SV Zulte-Waregem	0.37
Sporting Lokeren	0.35
KSV Cercle Brugge	0.32
RSC Anderlecht	0.64
KAA Gent	0.50
RAEC Mons	0.27
FCV Dender EH	0.26
Standard de Liège	0.50
KV Mechelen	0.37
Club Brugge KV	0.58
KSV Roeselare	0.19
KV Kortrijk	0.39
Tubize	0.21
Royal Excel Mouscron	0.35
KVC Westerlo	0.30
Sporting Charleroi	0.30
Sint-Truidense VV	0.26
Lierse SK	0.17

上圖只截取一部分，共有 229 筆資料 ( 與隊伍數同 )，我的作法是計算每一支隊伍贏的次數除以出場比賽次數，獲得勝率，因此賭者可以根據這個表選擇應該要押哪支隊伍

Code 的部分先計算出場次數以及贏的次數，利用隊伍 id 做分類，計算勝率