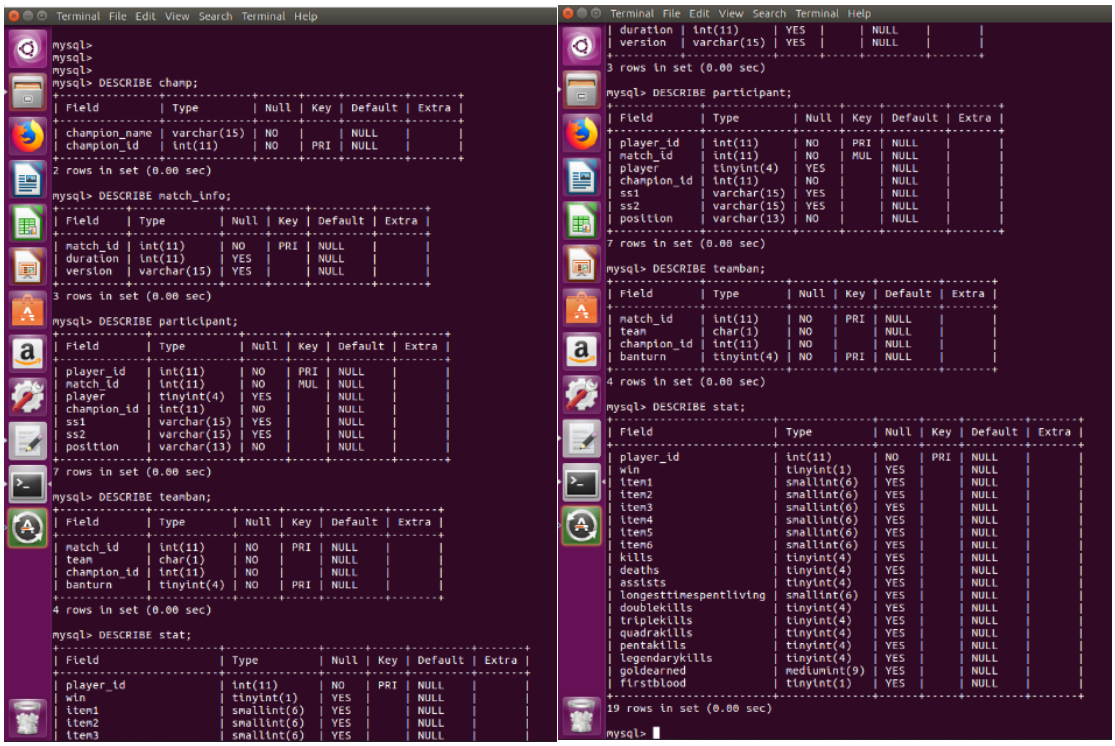
Introduction to Database Systems
Individual Homework 1: SQL tasks in MySQL

資工大三 0616231 彭世丞

A. **Create Tables**



1. (3%) What the difference between type “char” and type “varchar”?
   Answer: char 是固定長度，varchar 是可變長度。

2. (3%) Type “boolean” would be stored as which type in MySQL?
   Answer: 以 tinyint 來儲存。

3. (4%) 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)
   Answer: Tinyint ➔ 1 byte, -128~127
   Smallint ➔ 2 bytes, -32768~32767
   Mediumint ➔ 3 bytes, -8388608~8388607
   Int ➔ 4 bytes, -2147483648~2147483647

4. (5%) What do you think about this table schema? If you can change this table architecture, how would you modify it and why?
   Answer: 我覺得這些 tables 非常的完整而且清楚。如果想要更動的話，因為 champion_name 和 champion_id 是一對一關係，champ 這個 table 其實不需要。而 participant 和 teamban 這兩個 table 的 champion_id 就以 champion_name 替代即可。

B. Load csv Datas

C. Query Tasks

1. (5%) Please list the number of all different champions. You must have “COUNT” syntax in usage of SQL.

```
mysql> SELECT COUNT(*) as cnt
    -> FROM champ;
+-----+
| cnt |
+-----+
| 138 |
+-----+
1 row in set (0.00 sec)
```

2. (5%) Please list the number of different versions. They are same version if the first two numbers of version are same. For example, “7.9.186.1051” and “7.9.186.8155” belong to same version, but different with “7.8.184.113”. You must have “DISTINCT” syntax in usage of SQL.

```
mysql> SELECT COUNT(DISTINCT result.run) as cnt
    -> FROM(
    ->   SELECT substring_index(M.version,'.',2) as run
    ->   FROM match_info M
    ->   GROUP BY run
    -> ) as result;
+-----+
| cnt |
+-----+
|  74 |
+-----+
1 row in set (0.17 sec)
```

3. (5%) Please list the top 3 frequently use of the champion names and counts, which the position summoner choosing is JUNGLE. You must sort counts in decreasing order and have “ORDER BY” syntax in usage of SQL.

```
mysql> SELECT c.champion_name as champion_name, COUNT(*) as cnt
    -> FROM champ as c, participant as p
    -> WHERE c.champion_id = p.champion_id
    -> AND p.position = 'JUNGLE'
    -> GROUP BY c.champion_name
    -> ORDER BY cnt DESC
    -> LIMIT 3;
+---------------+-------+
| champion_name | cnt   |
+---------------+-------+
| Lee Sin       | 56598 |
| Master Yi     | 23385 |
| Graves        | 19767 |
+---------------+-------+
3 rows in set (0.71 sec)
```

4. (5%) Please list the top 5 longest match id and how long the game is taken. You should transfer time format to hh:mm:ss.

```
mysql> SELECT M.match_id as match_id, SEC_TO_TIME(M.duration) as time
    -> FROM match_info M
    -> ORDER BY time DESC
    -> LIMIT 5;
+----------+----------+
| match_id | time     |
+----------+----------+
|   146486 | 01:23:11 |
|    69303 | 01:20:14 |
|      581 | 01:16:59 |
|    70361 | 01:15:06 |
|   176628 | 01:13:34 |
+----------+----------+
5 rows in set (0.10 sec)
```

5. (5%) There are two teams in every match. Please list the number of winning teams and losing teams which average longest time spent living in each team greater than or equals to twenty minutes. You must output win or lose in string as following example. Note that longesttimespentliving only refers to one player's longest time spent living.

```
mysql> SELECT
    -> CASE
    ->      WHEN result.hahawin=1 THEN 'win'
    ->      ELSE 'lose'
    -> END AS win_lose,COUNT(*) cnt
    -> FROM(SELECT P.match_id as hahanum,
    ->      S.win as hahawin,
    ->      AVG(S.longesttimespentliving) as hahatime
    ->      FROM participant P, stat S
    ->      WHERE P.player_id = S.player_id
    ->      GROUP BY hahanum, hahawin
    -> ) as result
    -> WHERE result.hahatime>=1200
    -> GROUP BY result.hahawin;
+----------+-----+
| win_lose | cnt |
+----------+-----+
| lose     | 338 |
| win      | 807 |
+----------+-----+
2 rows in set (24.70 sec)
```

6. (5%) In LoL, some teams will pick champions which have great ability to win matches in earlier or later period. Please list the most appear champions of each position (TOP/MID/JUNGLE/DUO_CARRY/DUO_SUPPORT) which the matches end in forty to fifty minutes (including 40 and 50 minutes). You need to sort position in alphabetical order as following example, and you must have “BETWEEN” syntax in usage of SQL.

```
mysql> SELECT result.position, result.champion_name
    -> FROM(
    ->   SELECT P.position, C.champion_name, COUNT(*) cnt
    ->   FROM champ C, participant P, match_info M
    ->   WHERE C.champion_id = P.champion_id
    ->   AND M.match_id = P.match_id
    ->   AND M.duration BETWEEN 2400 AND 3000
    ->   GROUP BY P.position,C.champion_name
    -> ) result
    -> WHERE NOT EXISTS (
    ->   SELECT *
    ->   FROM(
    ->     SELECT P.position, C.champion_name, COUNT(*) cnt
    ->     FROM champ C, participant P, match_info M
    ->     WHERE C.champion_id = P.champion_id
    ->     AND M.match_id = P.match_id
    ->     AND M.duration BETWEEN 2400 AND 3000
    ->     GROUP BY P.position,C.champion_name
    ->   ) yummy
    ->   WHERE result.position = yummy.position
    ->   AND yummy.cnt > result.cnt)
    ->   AND ( result.position='DUO_CARRY' OR result.position='DUO_SUPPORT'
    ->   OR result.position='JUNGLE' OR result.position='MID'
    ->   OR result.position='TOP' )
    -> ORDER BY result.position ASC;
+-------------+---------------+
| position    | champion_name |
+-------------+---------------+
| DUO_CARRY   | Caitlyn       |
| DUO_SUPPORT | Thresh        |
| JUNGLE      | Lee Sin       |
| MID         | Ahri          |
| TOP         | Riven         |
+-------------+---------------+
5 rows in set (30.08 sec)
```

7. (10%) Please list the champion names with highest KDA (KDA = (sum_of_Kills + sum_of_Assists) / sum_of_Deaths) and its corresponding KDA of each position. Note that you should not take into account if the total number of deaths of a champion is zero. You need to sort position in alphabetical order as following example. Hint: GROUP BY

```
mysql> SELECT result.position, result.champion_name, result.up/result.down as kda
    -> FROM(
    ->   SELECT P.position, C.champion_name, AVG(S.kills)+AVG(S.assists) as up, AVG(S.deaths) as down
    ->   FROM champ C, participant P, stat S
    ->   WHERE C.champion_id = P.champion_id
    ->   AND P.player_id = S.player_id
    ->   AND ( P.position='DUO_CARRY' OR P.position='DUO_SUPPORT'
    ->   OR P.position='JUNGLE' OR P.position='MID'
    ->   OR P.position='TOP' )
    ->   GROUP BY P.position,C.champion_name
    -> ) result
    -> WHERE NOT EXISTS (
    ->   SELECT yummy.position, yummy.champion_name, yummy.up/yummy.down as kda
    ->   FROM(
    ->     SELECT P.position, C.champion_name, AVG(S.kills)+AVG(S.assists) as up, AVG(S.deaths) as down
    ->     FROM champ C, participant P, stat S
    ->     WHERE C.champion_id = P.champion_id
    ->     AND P.player_id = S.player_id
    ->     AND ( P.position='DUO_CARRY' OR P.position='DUO_SUPPORT'
    ->     OR P.position='JUNGLE' OR P.position='MID'
    ->     OR P.position='TOP' )
    ->     GROUP BY P.position,C.champion_name
    ->   ) yummy
    ->   WHERE result.position = yummy.position
    ->   AND yummy.up/yummy.down > result.up/result.down
    ->   AND yummy.down>0)
    -> AND result.down>0
    -> ORDER BY result.position ASC;
+-------------+---------------+-------------+
| position    | champion_name | kda         |
+-------------+---------------+-------------+
| DUO_CARRY   | Shaco         | 19.00000000 |
| DUO_SUPPORT | Janna         |  3.83303972 |
| JUNGLE      | Ivern         |  3.87635729 |
| MID         | Ivern         |  3.70143114 |
| TOP         | Sona          |  3.15384615 |
+-------------+---------------+-------------+
5 rows in set (38.08 sec)
```

8. (5%) Please list the champion names which are not banned in version 7.7. You need to sort champion names in in alphabetical order, and you must have “NOT IN” syntax in usage of SQL.

```
mysql> SELECT DISTINCT C.champion_name
    -> FROM champ C
    -> WHERE C.champion_name NOT IN(SELECT C.champion_name
    ->                              FROM match_info M, teamban T, champ C
    ->                              WHERE M.match_id = T.match_id
    ->                              AND C.champion_id = T.champion_id
    ->                              AND substring_index(M.version,'.',2)=7.7)
    -> ORDER BY C.champion_name ASC;
+---------------+
| champion_name |
+---------------+
| Kayn          |
| Ornn          |
| Rakan         |
| RekSai        |
| Sion          |
| Xayah         |
+---------------+
6 rows in set (0.13 sec)
```

9. (10%) Please list the number of win, lose counts and its winning ratio (#win / #(win+lose)) in each version which definition is same as Q2 when Lee Sin and Teemo are in same teams in the match.

```
mysql> SELECT substring_index(M.version,'.',2) as version, SUM(S.win) as win_cnt, COUNT(*)-SUM(S.win) as lose_cnt, SUM(S.win)/COUNT(*) as win_ratio
    -> FROM match_info M, participant P1, champ C1, stat S,
    ->   participant P2, champ C2
    -> WHERE M.match_id=P1.match_id AND P1.champion_id=C1.champion_id
    ->   AND M.match_id=P2.match_id AND P2.champion_id=C2.champion_id
    ->   AND (C1.champion_name = 'Lee Sin' AND C2.champion_name = 'Teemo')
    ->   AND P1.player_id=S.player_id
    ->   AND ((P1.player>5 AND P2.player>5) OR (P1.player<=5 AND P2.player<=5))
    -> GROUP BY substring_index(M.version,'.',2);
+---------+---------+----------+-----------+
| version | win_cnt | lose_cnt | win_ratio |
+---------+---------+----------+-----------+
| 4.10    |       2 |        1 |    0.6667 |
| 4.12    |       0 |        1 |    0.0000 |
| 4.15    |       1 |        1 |    0.5000 |
| 4.17    |       0 |        1 |    0.0000 |
| 4.18    |       0 |        1 |    0.0000 |
| 4.19    |       0 |        1 |    0.0000 |
| 4.21    |       1 |        1 |    0.5000 |
| 4.9     |       1 |        0 |    1.0000 |
| 5.1     |       1 |        2 |    0.3333 |
| 5.12    |       1 |        0 |    1.0000 |
| 5.13    |       0 |        1 |    0.0000 |
| 5.15    |       0 |        1 |    0.0000 |
| 5.19    |       1 |        0 |    1.0000 |
| 5.20    |       2 |        0 |    1.0000 |
| 5.21    |       0 |        2 |    0.0000 |
| 5.24    |       1 |        1 |    0.5000 |
| 5.5     |       1 |        0 |    1.0000 |
| 5.6     |       0 |        1 |    0.0000 |
| 5.7     |       1 |        0 |    1.0000 |
| 6.1     |       0 |        1 |    0.0000 |
| 6.13    |       1 |        0 |    1.0000 |
| 6.14    |       1 |        0 |    1.0000 |
| 6.18    |       1 |        1 |    0.5000 |
| 6.19    |       1 |        0 |    1.0000 |
| 6.2     |       1 |        1 |    0.5000 |
| 6.20    |       3 |        2 |    0.6000 |
| 6.21    |       0 |        2 |    0.0000 |
| 6.22    |       2 |        1 |    0.6667 |
| 6.23    |       3 |        2 |    0.6000 |
| 6.24    |       4 |        3 |    0.5714 |
| 6.5     |       1 |        0 |    1.0000 |
| 6.6     |       0 |        1 |    0.0000 |
| 6.8     |       1 |        0 |    1.0000 |
| 6.9     |       1 |        1 |    0.5000 |
| 7.10    |     282 |      304 |    0.4812 |
| 7.2     |       2 |        1 |    0.6667 |
| 7.3     |       0 |        1 |    0.0000 |
| 7.4     |       1 |        1 |    0.5000 |
| 7.5     |       2 |        2 |    0.5000 |
| 7.6     |       2 |        5 |    0.2857 |
| 7.7     |      32 |       29 |    0.5246 |
| 7.8     |     210 |      237 |    0.4698 |
| 7.9     |     527 |      464 |    0.5318 |
+---------+---------+----------+-----------+
43 rows in set (20.02 sec)
```

10. (15%)Please list the top 5 winning ratio of champion names, KDA which is defined as Q9 and average gold earned (goldearned) of both sides and battle records when summoners select TOP position and the opposite champion is Renekton. Note that you only need to consider the number of matches of each champion facing Renekton larger than 100.

```
mysql> SELECT result.self_champ_name, result.win_ratio,  result.up2/result.down2 as self_kda,
    -> result.self_avg_gold, result.enemy_champ_name,  result.up1/result.down1 as enemy_kda,
    -> result.enemy_avg_gold, result.battle_record
    -> FROM ( SELECT C2.champion_name as self_champ_name, SUM(S2.win)/COUNT(*) as win_ratio,
    ->   AVG(S2.kills)+AVG(S2.assists) as up2, AVG(S2.deaths) as down2,
    ->   AVG(S2.goldearned) as self_avg_gold,
    ->   C1.champion_name as enemy_champ_name,
    ->   AVG(S1.kills)+AVG(S1.assists) as up1, AVG(S1.deaths) as down1,
    ->   AVG(S1.goldearned) as enemy_avg_gold,
    ->   COUNT(*) as battle_record
    ->   FROM match_info M, participant P1, champ C1, stat S1,
    ->   participant P2, champ C2, stat S2
    ->   WHERE M.match_id=P1.match_id AND M.match_id=P2.match_id
    ->   AND P1.champion_id=58
    ->   AND P1.champion_id = C1.champion_id
    ->   AND P1.player_id = S1.player_id
    ->   AND P2.champion_id = C2.champion_id
    ->   AND P2.player_id = S2.player_id
    ->   AND ((P1.player>5 AND P2.player<=5) OR (P1.player<=5 AND P2.player>5))
    ->   AND P1.position='TOP'
    ->   AND P2.position='TOP'
    ->   GROUP BY C2.champion_name ) as result
    -> WHERE result.battle_record>100
    -> AND result.down2>0
    -> AND result.down1>0
    -> ORDER BY result.win_ratio DESC
    -> LIMIT 5;
+-----------------+-----------+------------+---------------+-----------------+------------+----------------+---------------+
| self_champ_name | win_ratio | self_kda   | self_avg_gold | enemy_champ_name | enemy_kda  | enemy_avg_gold | battle_record |
+-----------------+-----------+------------+---------------+-----------------+------------+----------------+---------------+
| Teemo           |    0.5756 | 1.93771605 |    12429.4756 | Renekton        | 1.85897785 |     11838.0422 |           450 |
| Pantheon        |    0.5433 | 2.35779082 |    11710.6325 | Renekton        | 1.68774692 |     11377.3648 |           381 |
| Nautilus        |    0.5399 | 2.52447552 |    10610.6304 | Renekton        | 2.09916641 |     12158.3877 |           276 |
| Jax             |    0.5336 | 1.83328610 |    11920.0637 | Renekton        | 1.98741042 |     11737.3632 |           581 |
| Jarvan IV       |    0.5333 | 2.52045246 |    10889.9185 | Renekton        | 1.93473084 |     11220.7037 |           135 |
+-----------------+-----------+------------+---------------+-----------------+------------+----------------+---------------+
5 rows in set (8.67 sec)
```

11. (10%) Do summoners choosing Flash and Ignite as summoner spells (ss1 and ss2) have more opportunity to win than choosing Flash and Teleport as summoner spells? (Answer by your own view)

```
mysql> SELECT SUM(result.FlashIgnite), SUM(result.FlashTeleport)
    -> FROM(SELECT motor.champion_id, motor.win_ratio as FlashIgnite, cycle.win_ratio as FlashTeleport
    ->   FROM(SELECT P.champion_id, SUM(S.win)/COUNT(*) as win_ratio
    ->     FROM participant P, stat S
    ->     WHERE P.player_id=S.player_id
    ->     AND ((P.ss1='Flash' AND P.ss2='Ignite') OR (P.ss2='Flash' AND P.ss1='Ignite'))
    ->     AND P.position='TOP'
    ->     GROUP BY P.champion_id
    ->   ) as motor, (SELECT P.champion_id, SUM(S.win)/COUNT(*) as win_ratio
    ->     FROM participant P, stat S
    ->     WHERE P.player_id=S.player_id
    ->     AND ((P.ss1='Flash' AND P.ss2='Teleport') OR (P.ss2='Flash' AND P.ss1='Teleport'))
    ->     AND P.position='TOP'
    ->     GROUP BY P.champion_id
    ->   ) as cycle
    ->   WHERE motor.champion_id=cycle.champion_id
    -> ) as result;
+-------------------------+--------------------------+
| SUM(result.FlashIgnite) | SUM(result.FlashTeleport) |
+-------------------------+--------------------------+
|                66.1791 |                  65.4215 |
+-------------------------+--------------------------+
1 row in set (14.82 sec)
```

```
mysql> SELECT SUM(motor.win_ratio>cycle.win_ratio) as FlashIgnite, SUM(motor.win_ratio<cycle.win_ratio) as FlashTeleport
    -> FROM(SELECT P.champion_id, SUM(S.win)/COUNT(*) as win_ratio
    ->   FROM participant P, stat S
    ->   WHERE P.player_id=S.player_id
    ->   AND ((P.ss1='Flash' AND P.ss2='Ignite') OR (P.ss2='Flash' AND P.ss1='Ignite'))
    ->   AND P.position='TOP'
    ->   GROUP BY P.champion_id
    -> ) as motor, (SELECT P.champion_id, SUM(S.win)/COUNT(*) as win_ratio
    ->   FROM participant P, stat S
    ->   WHERE P.player_id=S.player_id
    ->   AND ((P.ss1='Flash' AND P.ss2='Teleport') OR (P.ss2='Flash' AND P.ss1='Teleport'))
    ->   AND P.position='TOP'
    ->   GROUP BY P.champion_id
    -> ) as cycle
    -> WHERE motor.champion_id=cycle.champion_id;
+------------+--------------+
| FlashIgnite | FlashTeleport |
+------------+--------------+
|          77 |           58 |
+------------+--------------+
1 row in set (16.65 sec)
```

Ans:

a. 我想說每個角色 對於使用 Flash+Ignite 還是 Flash+Teleport 比較容易獲勝一定會有差異，所以我的想法是必須要把每個英雄抓出來個別討論 會比較公正。因此，我先計算每個角色對於 Flash+Ignite 的勝率(win_ratio)和 Flash+Teleport 的勝率(win_ratio)，然後再把所有角色的兩種勝率個別加總
   ➔ Flash+Ignite 勝。

b. 第二個 query 沿用上一個 query 的想法，但是因為本題是討論”獲勝的機率”，如果有兩組英雄的資料，一組英雄用 Flash+Ignite 獲勝的機率是 20%、用 Flash+Teleport 獲勝的機率是 80%，另一組用 Flash+Ignite 獲勝的機率是 53%、用 Flash+Teleport 獲勝的機率是 50%，用第一個 query 的話會造成第二筆資料被第一筆資料吃掉。因此這次，若這個英雄用 Flash+Ignite 獲勝的機率比 Flash+Teleport 高時，Flash+Ignite 記為 1、Flash+Teleport 記為 0，反之亦然。最後一樣，加總所有英雄的 FlashIgnite 項和 FlashTeleport
   ➔ Flash+Ignite 勝。

結論：Flash+Ignite 勝率高於 Flash+Teleport 勝率。

12. (Bonus 10%) Feel free to think any valuable observation with explanation.

```
mysql> SELECT god.champion_name, COUNT(*) as finalcount
    -> FROM(SELECT sky.newversion, C.champion_name
    ->   FROM(SELECT substring_index(M.version,'.',2) as newversion, bird.champion_id,
    ->       SUM(bird.wintimes)/SUM(bird.cnt) as win_ratio
    ->     FROM(SELECT P.champion_id, P.match_id, SUM(S.win) wintimes, COUNT(*) cnt
    ->       FROM participant P, stat S
    ->       WHERE P.player_id=S.player_id
    ->       GROUP BY P.champion_id, P.match_id
    ->     ) as bird, match_info M
    ->     WHERE bird.match_id=M.match_id
    ->     GROUP BY newversion, bird.champion_id
    ->   ) as sky, champ C
    ->   WHERE NOT EXISTS (
    ->     SELECT *
    ->     FROM(SELECT substring_index(M.version,'.',2) as newversion, bird.champion_id,
    ->         SUM(bird.wintimes)/SUM(bird.cnt) as win_ratio
    ->       FROM(SELECT P.champion_id, P.match_id, SUM(S.win) wintimes, COUNT(*) cnt
    ->         FROM participant P, stat S
    ->         WHERE P.player_id=S.player_id
    ->         GROUP BY P.champion_id, P.match_id
    ->       ) as bird, match_info M
    ->       WHERE bird.match_id=M.match_id
    ->       GROUP BY newversion, bird.champion_id
    ->     ) walker
    ->     WHERE sky.newversion=walker.newversion
    ->     AND walker.win_ratio > sky.win_ratio)
    ->     AND sky.champion_id=C.champion_id
    -> ) as god
    -> GROUP BY god.champion_name
    -> ORDER BY finalcount DESC
    -> LIMIT 20;
+---------------+------------+
| champion_name | finalcount |
+---------------+------------+
| Yorick        |         19 |
| Galio         |         17 |
| Skarner       |         16 |
| Urgot         |         16 |
| Karthus       |         10 |
| KogMaw        |          9 |
| Aatrox        |          9 |
| Poppy         |          9 |
| Mordekaiser   |          8 |
| Rammus        |          8 |
| Zilean        |          8 |
| Sion          |          7 |
| Heimerdinger  |          7 |
| Shyvana       |          7 |
| Fiddlesticks  |          7 |
| Olaf          |          7 |
| Kayle         |          7 |
| Singed        |          7 |
| Kassadin      |          7 |
| Xerath        |          6 |
+---------------+------------+
20 rows in set (47.57 sec)
```

Goal: 找出最強的英雄！

就我的認知，每次的改版對英雄的強弱都可能會有變動，所以有必要以版本來區分不同時期的所有英雄，並找出在每個版本下平均下來勝率最高的英雄，稱之為最強！

因此，我先找出在每個版本底下勝率最高(Top1)的英雄，再計算各個英雄在所有版本中有幾次為 Top1，並將次數(finalcount)由高排到低，則次數最高者為最強的角色。

➔ 最強角色：Yorick