

1 Task A

```
mysql> DESC champ;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| champion_name | varchar(15) | NO | | NULL | |
| champion_id | int | NO | PRI | NULL | |
+-----+
2 rows in set (0.03 sec)

mysql> DESC match_info;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| match_id | int | NO | PRI | NULL | |
| duration | int | YES | | NULL | |
| version | varchar(15) | YES | | NULL | |
+-----+
3 rows in set (0.01 sec)

mysql> DESC participant;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| player_id | int | NO | PRI | NULL | |
| match_id | int | NO | MUL | NULL | |
| player | tinyint | YES | | NULL | |
| champion_id | int | NO | | NULL | |
| ss1 | varchar(15) | YES | | NULL | |
| ss2 | varchar(15) | YES | | NULL | |
| position | varchar(13) | NO | | NULL | |
+-----+
7 rows in set (0.01 sec)

mysql> DESC teanban;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| match_id | int | NO | PRI | NULL | |
| team | char(1) | NO | | NULL | |
| champion_id | int | NO | | NULL | |
| banturn | tinyint | NO | PRI | NULL | |
+-----+
4 rows in set (0.02 sec)

mysql> DESC stat;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| player_id | int | NO | PRI | NULL | |
| win | tinyint(1) | YES | | NULL | |
| item1 | smallint | YES | | NULL | |
| item2 | smallint | YES | | NULL | |
| item3 | smallint | YES | | NULL | |
| item4 | smallint | YES | | NULL | |
| item5 | smallint | YES | | NULL | |
| item6 | smallint | YES | | NULL | |
| kills | tinyint | YES | | NULL | |
| deaths | tinyint | YES | | NULL | |
| assists | tinyint | YES | | NULL | |
| longesttimespentliving | smallint | YES | | NULL | |
| doublekills | tinyint | YES | | NULL | |
| triplekills | tinyint | YES | | NULL | |
| quadrakills | tinyint | YES | | NULL | |
| pentakills | tinyint | YES | | NULL | |
| legendarykills | tinyint | YES | | NULL | |
| goldearned | mediumint | YES | | NULL | |
| firstblood | tinyint(1) | YES | | NULL | |
+-----+
19 rows in set (0.08 sec)
```

1. What the difference between type **char** and type **varchar**?
char: Stores only fixed-length string
varchar: Stores variable-length string with upper limit specified.
2. Type **boolean** would be stored as which type in MySQL?
tinyint(1)
3. How many bytes it should take for **tinyint**, **smallint**, **mediumint**, **int**? And what's the range they can express?
tinyint: 1 bytes, from -128 to 127
smallint: 2 bytes, from -32768 to 32767
mediumint: 3 bytes, from -8388608 to 8388607
int: 4 bytes, from -2147483648 to 2147483647
4. What do you think about this table schema? If you can change this table architecture, how would you modify it and why?
This is a complete table schema that is able to operate easily. I may add some attributes such as champion_atk, champion_def, champion_HP, etc. in table **stat** to log the state.

2 Task C

1.

```
mysql> SELECT COUNT(DISTINCT champion_id) AS cnt FROM champ;
+-----+
| cnt |
+-----+
| 138 |
+-----+
1 row in set (0.00 sec)
```
2.

```
mysql> SELECT COUNT(DISTINCT SUBSTRING_INDEX(version, ".", 2)) AS cnt FROM match_info;
+-----+
| cnt |
+-----+
| 74 |
+-----+
1 row in set (0.14 sec)
```
3.

```
mysql> SELECT C.champion_name, COUNT(P.champion_id) AS cnt
-> FROM champ C, participant P
-> WHERE C.champion_id = P.champion_id AND P.position = "JUNGLE"
-> GROUP BY C.champion_name
-> ORDER BY COUNT(P.champion_id) DESC
-> LIMIT 3;
+-----+-----+
| champion_name | cnt |
+-----+-----+
| Lee Sin      | 56598 |
| Master Yi    | 23385 |
| Graves       | 19767 |
+-----+-----+
3 rows in set (1.96 sec)
```
4.

```
mysql> SELECT match_id, SEC_TO_TIME(duration) AS time
-> FROM match_info
-> ORDER BY duration 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 (0.14 sec)
```
5.

```
mysql> (SELECT "lose" AS win_lose, COUNT(AA.match_id) AS cnt
-> FROM (SELECT AVG(longesttimespentliving) AS avg_time, P.match_id
-> FROM participant P INNER JOIN stat S ON S.player_id = P.player_id
-> WHERE S.win = 0
-> GROUP BY P.match_id) AS AA
-> WHERE avg_time >= 1200)
-> UNION
-> (SELECT "win" AS win_lose, COUNT(AA.match_id) AS cnt
-> FROM (SELECT AVG(longesttimespentliving) AS avg_time, P.match_id
-> FROM participant P INNER JOIN stat S ON S.player_id = P.player_id
-> WHERE S.win = 1
-> GROUP BY P.match_id) AS AA
-> WHERE avg_time >= 1200);
+-----+-----+
| win_lose | cnt |
+-----+-----+
| lose    | 338 |
| win     | 807 |
+-----+-----+
2 rows in set (17.04 sec)
```

6.

```
mysql> (SELECT "DUO_CARRY" AS position, C.champion_name
-> FROM champ C, participant P
-> WHERE P.position = "DUO_CARRY"
-> AND P.match_id IN (SELECT match_id FROM match_info WHERE duration BETWEEN 2400 AND 3000)
-> AND P.champion_id = C.champion_id
-> GROUP BY C.champion_name
-> ORDER BY COUNT(P.champion_id) DESC
-> LIMIT 1)
-> UNION
-> (SELECT "DUO_SUPPORT" AS position, C.champion_name
-> FROM champ C, participant P
-> WHERE P.position = "DUO_SUPPORT"
-> AND P.match_id IN (SELECT match_id FROM match_info WHERE duration BETWEEN 2400 AND 3000)
-> AND P.champion_id = C.champion_id
-> GROUP BY C.champion_name
-> ORDER BY COUNT(P.champion_id) DESC
-> LIMIT 1)
-> UNION
-> (SELECT "JUNGLE" AS position, C.champion_name
-> FROM champ C, participant P
-> WHERE P.position = "JUNGLE"
-> AND P.match_id IN (SELECT match_id FROM match_info WHERE duration BETWEEN 2400 AND 3000)
-> AND P.champion_id = C.champion_id
-> GROUP BY C.champion_name
-> ORDER BY COUNT(P.champion_id) DESC
-> LIMIT 1)
-> UNION
-> (SELECT "MID" AS position, C.champion_name
-> FROM champ C, participant P
-> WHERE P.position = "MID"
-> AND P.match_id IN (SELECT match_id FROM match_info WHERE duration BETWEEN 2400 AND 3000)
-> AND P.champion_id = C.champion_id
-> GROUP BY C.champion_name
-> ORDER BY COUNT(P.champion_id) DESC
-> LIMIT 1)
-> UNION
-> (SELECT "TOP" AS position, C.champion_name
-> FROM champ C, participant P
-> WHERE P.position = "TOP"
-> AND P.match_id IN (SELECT match_id FROM match_info WHERE duration BETWEEN 2400 AND 3000)
-> AND P.champion_id = C.champion_id
-> GROUP BY C.champion_name
-> ORDER BY COUNT(P.champion_id) DESC
-> LIMIT 1);
+-----+-----+
| position | champion_name |
+-----+-----+
| DUO_CARRY | Caitlyn       |
| DUO_SUPPORT | Thresh       |
| JUNGLE    | Lee Sin      |
| MID       | Ahri         |
| TOP       | Riven        |
+-----+-----+
5 rows in set (7.64 sec)
```

7.

```
mysql> (SELECT "DUO_CARRY" AS position, C.champion_name, (IFNULL((SUM(S.kills) + SUM(S.assists)) / NULLIF(SUM(S.deaths), 0), 0)) AS kda
-> FROM champ C, participant P, stat S
-> WHERE P.position = "DUO_CARRY"
-> AND P.champion_id = C.champion_id
-> AND S.player_id = P.player_id
-> GROUP BY C.champion_name
-> ORDER BY (IFNULL((SUM(S.kills) + SUM(S.assists)) / NULLIF(SUM(S.deaths), 0), 0)) DESC
-> LIMIT 1)
-> UNION
-> (SELECT "DUO_SUPPORT" AS position, C.champion_name, (IFNULL((SUM(S.kills) + SUM(S.assists)) / NULLIF(SUM(S.deaths), 0), 0)) AS kda
-> FROM champ C, participant P, stat S
-> WHERE P.position = "DUO_SUPPORT"
-> AND P.champion_id = C.champion_id
-> AND S.player_id = P.player_id
-> GROUP BY C.champion_name
-> ORDER BY (IFNULL((SUM(S.kills) + SUM(S.assists)) / NULLIF(SUM(S.deaths), 0), 0)) DESC
-> LIMIT 1)
-> UNION
-> (SELECT "JUNGLE" AS position, C.champion_name, (IFNULL((SUM(S.kills) + SUM(S.assists)) / NULLIF(SUM(S.deaths), 0), 0)) AS kda
-> FROM champ C, participant P, stat S
-> WHERE P.position = "JUNGLE"
-> AND P.champion_id = C.champion_id
-> AND S.player_id = P.player_id
-> GROUP BY C.champion_name
-> ORDER BY (IFNULL((SUM(S.kills) + SUM(S.assists)) / NULLIF(SUM(S.deaths), 0), 0)) DESC
-> LIMIT 1)
-> UNION
-> (SELECT "MID" AS position, C.champion_name, (IFNULL((SUM(S.kills) + SUM(S.assists)) / NULLIF(SUM(S.deaths), 0), 0)) AS kda
-> FROM champ C, participant P, stat S
-> WHERE P.position = "MID"
-> AND P.champion_id = C.champion_id
-> AND S.player_id = P.player_id
-> GROUP BY C.champion_name
-> ORDER BY (IFNULL((SUM(S.kills) + SUM(S.assists)) / NULLIF(SUM(S.deaths), 0), 0)) DESC
-> LIMIT 1)
-> UNION
-> (SELECT "TOP" AS position, C.champion_name, (IFNULL((SUM(S.kills) + SUM(S.assists)) / NULLIF(SUM(S.deaths), 0), 0)) AS kda
-> FROM champ C, participant P, stat S
-> WHERE P.position = "TOP"
-> AND P.champion_id = C.champion_id
-> AND S.player_id = P.player_id
-> GROUP BY C.champion_name
-> ORDER BY (IFNULL((SUM(S.kills) + SUM(S.assists)) / NULLIF(SUM(S.deaths), 0), 0)) DESC
-> LIMIT 1);
+-----+-----+-----+
| position | champion_name | kda |
+-----+-----+-----+
| DUO_CARRY | Shaco         | 19.0000 |
| DUO_SUPPORT | Janna        | 3.8330 |
| JUNGLE    | Ivern        | 3.8764 |
| MID       | Ivern        | 3.7015 |
| TOP       | Sona         | 3.1538 |
+-----+-----+-----+
5 rows in set (24.73 sec)
```

8.

```
mysql> SELECT DISTINCT C.champion_name
-> FROM champ C, match_info M, teamban T
-> WHERE STRCMP(SUBSTRING_INDEX(M.version, ".", 2), "7.7") = 0
-> AND T.match_id = M.match_id
-> AND C.champion_id NOT IN (
-> SELECT DISTINCT T.champion_id
-> FROM teamban T, match_info M
-> WHERE STRCMP(SUBSTRING_INDEX(M.version, ".", 2), "7.7") = 0
-> AND T.match_id = M.match_id
-> )
-> ORDER BY C.champion_name;
+-----+
| champion_name |
+-----+
| Kayn          |
| Ornn          |
| Rakan         |
| RekSai        |
| Sion          |
| Xayah         |
+-----+
6 rows in set (0.32 sec)
```

9.

```
mysql> SELECT SUBSTRING_INDEX(A.version, ".", 2) AS version, SUM(A.win = 1) AS win_cnt,
-> SUM(A.win = 0) AS lose_cnt, SUM(A.win = 1) / (SUM(A.win = 1) + SUM(A.win = 0)) AS win_ratio
-> FROM (
-> (SELECT SUBSTRING_INDEX(M.version, ".", 2) AS version, M.match_id, S.win
-> FROM match_info M, participant P, champ C, stat S
-> WHERE P.match_id = M.match_id
-> AND C.champion_id = P.champion_id
-> AND C.champion_name = "Lee Sin"
-> AND S.player_id = P.player_id
-> ) AS A
-> INNER JOIN(
-> SELECT SUBSTRING_INDEX(M.version, ".", 2) AS version, M.match_id, S.win
-> FROM match_info M, participant P, champ C, stat S
-> WHERE P.match_id = M.match_id
-> AND C.champion_id = P.champion_id
-> AND C.champion_name = "Teemo"
-> AND S.player_id = P.player_id
-> ) AS B
-> ON A.match_id = B.match_id AND A.win = B.win)
-> GROUP BY A.version
-> ORDER BY A.version;
+-----+-----+-----+-----+
| 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 (5.55 sec)
```

10.

```
mysql> SELECT * FROM(
-> SELECT A.champion_name AS self_champ_name, SUM(A.win = 1) / (SUM(A.win = 1) + SUM(A.win = 0)) AS win_ratio,
-> IFNULL((SUM(A.kills) + SUM(A.assists)) / NULLIF(SUM(A.deaths), 0), 0) AS self_kda, AVG(A.goldearned) AS self_avg_gold,
-> "Gragas" AS enemy_champ_name, IFNULL((SUM(B.kills) + SUM(B.assists)) / NULLIF(SUM(B.deaths), 0), 0) AS enemy_kda,
-> AVG(B.goldearned) AS enemy_avg_gold, COUNT(A.champion_name) AS battle_record
-> FROM (
-> (SELECT C.champion_name, P.match_id, S.win, S.kills, S.assists, S.deaths, S.goldearned
-> FROM participant P, champ C, stat S
-> WHERE C.champion_id = P.champion_id
-> AND C.champion_name != "Gragas"
-> AND P.position = "TOP"
-> AND S.player_id = P.player_id
-> ) AS A
-> INNER JOIN
-> (SELECT C.champion_name, P.match_id, S.win, S.kills, S.assists, S.deaths, S.goldearned
-> FROM participant P, champ C, stat S
-> WHERE C.champion_id = P.champion_id
-> AND C.champion_name = "Gragas"
-> AND P.position = "TOP"
-> AND S.player_id = P.player_id
-> ) AS B
-> ON A.match_id = B.match_id AND A.win != B.win)
-> GROUP BY A.champion_name
-> ORDER BY (SUM(A.win = 1) / (SUM(A.win = 1) + SUM(A.win = 0))) DESC) AS AA
-> WHERE AA.battle_record > 100
-> LIMIT 5;
```

self_champ_name	win_ratio	self_kda	self_avg_gold	enemy_champ_name	enemy_kda	enemy_avg_gold	battle_record
Yasuo	0.6042	1.7255	12501.5833	Gragas	2.4916	10755.1042	288
Darius	0.5897	2.1526	12049.7920	Gragas	2.2283	10709.1567	351
Jax	0.5805	1.7122	12111.4634	Gragas	2.3960	10978.0780	205
Teemo	0.5798	1.8163	12361.1968	Gragas	2.4122	11212.8511	188
Pantheon	0.5515	2.2233	11616.3015	Gragas	2.1137	10785.1176	136

5 rows in set (6.04 sec)

11.

```
mysql> SELECT * FROM(
-> SELECT A.summoner_spell, A.win_cnt, A.lose_cnt, A.win_cnt / (A.win_cnt + A.lose_cnt) AS win_ratio FROM(
-> (SELECT P.ss1 AS summoner_spell, SUM(S.win = 1) AS win_cnt, SUM(S.win = 0) AS lose_cnt
-> FROM participant P, stat S
-> WHERE P.player_id = S.player_id
-> AND P.position = "TOP"
-> GROUP BY P.ss1) AS A
-> INNER JOIN
-> (SELECT P.ss2 AS summoner_spell, SUM(S.win = 1) AS win_cnt, SUM(S.win = 0) AS lose_cnt
-> FROM participant P, stat S
-> WHERE P.player_id = S.player_id
-> AND P.position = "TOP"
-> GROUP BY P.ss2) AS B
-> ON STRCMP(A.summoner_spell, B.summoner_spell) = 0)
-> GROUP BY A.summoner_spell) AS AA
-> WHERE (AA.win_cnt + AA.lose_cnt) > 100
-> ORDER BY win_ratio DESC;
```

summoner_spell	win_cnt	lose_cnt	win_ratio
Ignite	11325	10417	0.5209
Flash	97811	97174	0.5016
Teleport	67233	67127	0.5004
Haste	3250	3318	0.4954
Smite	273	317	0.4627
Exhaust	1551	1829	0.4589
Barrier	169	212	0.4436
Heal	379	480	0.4412

8 rows in set (5.04 sec)

想取得如何選擇召喚師技能以取得最大勝率的方法，首先就是要先分析選擇各種召喚師技能的勝率為何。首先要做的便是將 ss1、ss2 的數據合併，先將使用 ss1、ss2 的 summoner_spell、win_cnt、lose_cnt、win_ratio 輸出，其中取用了 participant(P)、stat(S) 這兩個表格，篩選條件則為 player_id 相同、position 為 TOP，並利用 summoner_spell 的種類來分群。接下來利用 INNER JOIN 合併，合併的條件為兩表格的 summoner_spell 相同，並將兩表格的數據合併。最後則是篩選資料大於 100 筆以扣除過小的樣本可能產生的誤差，並以勝率進行排序。從結果來看，可以發現使用 **Ignite**、**Flash** 這兩種召喚師技能的勝率最高，恰巧也是最多人使用的，因此應選用這兩種召喚師技能以獲得最大的勝率。

12. 在每個位置首殺比率前 3 名的英雄為？取得首殺的玩家會有較佳的數據嗎？

為了取得在每個位置首殺比率排名前 3 的英雄，需要先將各位置先排序後再合併。在表格中輸出的資料為 position、win_cnt、lose_cnt、first_blood_ratio，取用了 champ(C)、participant(P)、stat(S) 這三個表格，篩選條件為位置正確、champion_id 相等、player_id 相等，並以角色名稱分群，並篩選資料大於 100 筆以扣除過小的樣本可能產生的誤差，最後再將所有表格 UNION 得到各位置角色首殺比率前 3 名的英雄。從結果中我們也可以看到 **JUNGLE** 的角色首殺比率比其他位置高一些。

此外，想比較取得首殺的玩家數據是否較好，只要由 stat 這個表格輸出需要的資料 (win_ratio、avg_kda 等)，並以 firstblood 分群便可比較。我們可以明顯看出取得首殺的玩家在各項數據的平均值基本上皆超過沒有取得首殺的玩家，可見取得首殺的玩家實力相對優異。

```
mysql> (SELECT * FROM(
-> SELECT "DUO_CARRY" AS position, C.champion_name, SUM(S.win = 1) AS win_cnt, SUM(S.win = 0) AS lose_cnt,
-> SUM(S.firstblood = 1) / (SUM(S.firstblood = 0) + SUM(S.firstblood = 1)) AS first_blood_ratio
-> FROM champ C, participant P, stat S
-> WHERE P.position = "DUO_CARRY"
-> AND C.champion_id = P.champion_id
-> AND P.player_id = S.player_id
-> GROUP BY C.champion_name) AS AA
-> WHERE (win_cnt + lose_cnt) > 100
-> ORDER BY first_blood_ratio DESC
-> LIMIT 3)
-> UNION
-> (SELECT * FROM(
-> SELECT "DUO_SUPPORT" AS position, C.champion_name, SUM(S.win = 1) AS win_cnt, SUM(S.win = 0) AS lose_cnt,
-> SUM(S.firstblood = 1) / (SUM(S.firstblood = 0) + SUM(S.firstblood = 1)) AS first_blood_ratio
-> FROM champ C, participant P, stat S
-> WHERE P.position = "DUO_SUPPORT"
-> AND C.champion_id = P.champion_id
-> AND P.player_id = S.player_id
-> GROUP BY C.champion_name) AS AA
-> WHERE (win_cnt + lose_cnt) > 100
-> ORDER BY first_blood_ratio DESC
-> LIMIT 3)
-> UNION
-> (SELECT * FROM(
-> SELECT "JUNGLE" AS position, C.champion_name, SUM(S.win = 1) AS win_cnt, SUM(S.win = 0) AS lose_cnt,
-> SUM(S.firstblood = 1) / (SUM(S.firstblood = 0) + SUM(S.firstblood = 1)) AS first_blood_ratio
-> FROM champ C, participant P, stat S
-> WHERE P.position = "JUNGLE"
-> AND C.champion_id = P.champion_id
-> AND P.player_id = S.player_id
-> GROUP BY C.champion_name) AS AA
-> WHERE (win_cnt + lose_cnt) > 100
-> ORDER BY first_blood_ratio DESC
-> LIMIT 3)
-> UNION
-> (SELECT * FROM(
-> SELECT "MID" AS position, C.champion_name, SUM(S.win = 1) AS win_cnt, SUM(S.win = 0) AS lose_cnt,
-> SUM(S.firstblood = 1) / (SUM(S.firstblood = 0) + SUM(S.firstblood = 1)) AS first_blood_ratio
-> FROM champ C, participant P, stat S
-> WHERE P.position = "MID"
-> AND C.champion_id = P.champion_id
-> AND P.player_id = S.player_id
-> GROUP BY C.champion_name) AS AA
-> WHERE (win_cnt + lose_cnt) > 100
-> ORDER BY first_blood_ratio DESC
-> LIMIT 3)
-> UNION
-> (SELECT * FROM(
-> SELECT "TOP" AS position, C.champion_name, SUM(S.win = 1) AS win_cnt, SUM(S.win = 0) AS lose_cnt,
-> SUM(S.firstblood = 1) / (SUM(S.firstblood = 0) + SUM(S.firstblood = 1)) AS first_blood_ratio
-> FROM champ C, participant P, stat S
-> WHERE P.position = "TOP"
-> AND C.champion_id = P.champion_id
-> AND P.player_id = S.player_id
-> GROUP BY C.champion_name) AS AA
-> WHERE (win_cnt + lose_cnt) > 100
-> ORDER BY first_blood_ratio DESC
-> LIMIT 3);
```

position	champion_name	win_cnt	lose_cnt	first_blood_ratio
DUO_CARRY	Kalista	3054	3106	0.1593
DUO_CARRY	Vasuo	136	99	0.1447
DUO_CARRY	Draven	9664	8592	0.1265
DUO_SUPPORT	Ezreal	66	89	0.1226
DUO_SUPPORT	Pantheon	77	69	0.1164
DUO_SUPPORT	Nidalee	71	105	0.1080
JUNGLE	Twitch	2466	2363	0.3046
JUNGLE	Katarina	257	239	0.2177
JUNGLE	Talon	219	258	0.2034
MID	Pantheon	321	258	0.2159
MID	Talon	2732	2786	0.2151
MID	Wukong	157	221	0.2063
TOP	Rengar	545	564	0.2849
TOP	Twitch	122	83	0.2585
TOP	Draven	138	111	0.2450

15 rows in set (25.34 sec)

```
mysql> SELECT firstblood, SUM(win = 1) / (SUM(win = 0) + SUM(win = 1)) AS win_ratio,
-> (IFNULL((SUM(kills) + SUM(assists)) / NULLIF(SUM(deaths), 0), 0)) AS avg_kda,
-> AVG(goldearned) AS avg_gold, AVG(longesttimespentliving) AS avg_longest_alive,
-> AVG(doublekills) AS avg_double_kills, AVG(triplekills) AS avg_triple_kills, AVG(quadrakills) AS avg_quadra_kills,
-> AVG(pentakills) AS avg_penta_kills, AVG(legendarykills) AS avg_legendary_kills
-> FROM stat
-> GROUP BY firstblood;
```

firstblood	win_ratio	avg_kda	avg_gold	avg_longest_alive	avg_double_kills	avg_triple_kills	avg_quadra_kills	avg_penta_kills	avg_legendary_kills
0	0.4896	2.3890	11286.9000	630.9444	0.5112	0.0681	0.0096	0.0017	0.0000
1	0.5948	2.7927	12666.5192	633.7310	0.8164	0.1182	0.0181	0.0036	0.0000

2 rows in set (3.57 sec)