Introduction to Database Systems

Individual Homework 1 : SQL tasks in MySQL

Part A：

CREATE TABLE：

1. champ table：

```
mysql> describe 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)
```

2. match_info table：

```
mysql> DESCRIBE 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 (0.00 sec)
```

3. participant table：

```
mysql> DESCRIBE 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 (0.00 sec)
```

4. teamban table：

```
mysql> DESCRIBE teamban;
+-------------+---------+------+-----+---------+-------+
| 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 (0.00 sec)
```

5. stat table：

```
mysql> DESCRIBE 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 (0.00 sec)
```

QUESTIONS：

1. What the difference between type “char” and type “varchar”?

   ANS：

   The benefit of varchar is its storage efficiency. The type of char is used when we have fixed-length data requirements or when we are dealing with variable-length strings.

2. Type “boolean” would be stored as which type in MySQL?

   ANS：

   Using “tinyint “ to store booleans, it provides convenient handling of boolean values in SQL queries and applications.

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)

   ANS：

	tinyint	smallint	mediumint	int
byte	1 byte	2 bytes	3 bytes	4 bytes
range	-128 to 127 (signed), 0 to 255 (unsigned)	-32768 to 32767 (signed) 0 to 65535 (unsigned)	8388608 to 8388607 (signed) 0 to 16777215 (unsigned)	-2147483648 to 2147483647 (signed) 0 to 4294967295 (unsigned)

4. What do you think about this table schema? If you can change this table architecture, how would you modify it and why?

   ANS：

   It's clearly to understand the schema depends on the specific requirements and use cases of your application.

   In my perspective, adding an index on “match_id” in the “stat” table can speed up queries that filter or join based on player IDs.

Part C：

1. Please list the number of all different champions. You must have "**COUNT**" syntax in usage of SQL.

```
1   SELECT COUNT(champion_name) AS cnt
2   FROM champ;
```

```
mysql> select count(champion_name) as cnt
    -> from champ;
+-----+
| cnt |
+-----+
| 138 |
+-----+
1 row in set (0.01 sec)
```

2. Please list the number of different versions. They are same version if the first two numbers of version are same. For example, "7.9.185.1051" and "7.9.186.8155" belong to same version, but different with "7.92.184.113". You must have "**DISTINCT**" syntax in usage of SQL.

```
1   SELECT COUNT(distinct version) as cnt
2   FROM match_info;
```

```
mysql> select count(distinct version) as cnt
    -> from match_info;
+-----+
| cnt |
+-----+
| 152 |
+-----+
1 row in set (0.05 sec)
```

3. 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.

```sql
SELECT champ.champion_name, COUNT(*) AS cnt FROM champ
join participant on participant.champion_id=champ.champion_id
where participant.position="JUNGLE"
group by champion_name
order by cnt desc limit 3;
```

```
mysql> SELECT champ.champion_name, COUNT(*) AS cnt FROM champ
    -> join participant on participant.champion_id=champ.champion_id
    -> where participant.position="JUNGLE"
    -> group by champion_name
    -> order by cnt desc limit 3;
+---------------+-------+
| champion_name | cnt   |
+---------------+-------+
| Lee Sin       | 56598 |
| Master Yi     | 23385 |
| Graves        | 19767 |
+---------------+-------+
3 rows in set (1.25 sec)
```

4. Please list the top 5 longest match id and how long the game is taken.

   You should transfer time format to hh:mm:ss.

```
1   select match_id, sec_to_time(duration) as time
2   from match_info
3   order by duration desc limit 5;
```

```
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 (0.04 sec)
```

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.

```sql
SELECT IF(tmp.win=0, 'lose', 'win') AS win_lose, COUNT(*) AS cnt
FROM (
  SELECT s.win AS win
  FROM stat AS s
  JOIN participant AS p ON p.player_id = s.player_id
  GROUP BY s.win, p.match_id
  HAVING sum(s.longesttimespentliving) ≥ 6000
) AS tmp
GROUP BY tmp.win;
```

```
+----------+-----+
| win_lose | cnt |
+----------+-----+
| lose     | 338 |
| win      | 807 |
+----------+-----+
2 rows in set (7.92 sec)
```

6. 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.

```sql
SELECT position, champion_name
FROM (
    SELECT p.position AS position,COUNT(*) AS cnt,c.champion_name AS champion_name
    FROM participant AS p
    JOIN stat AS s
        ON s.player_id = p.player_id
    JOIN match_info AS m
        ON p.match_id = m.match_id
    JOIN champ AS c
        ON c.champion_id = p.champion_id
    WHERE m.duration BETWEEN 2400 AND 3000
    GROUP BY p.position, p.champion_id
    ORDER BY position, cnt DESC
) AS tmp
WHERE position IN ('DUO_CARRY', 'DUO_SUPPORT', 'JUNGLE', 'MID', 'TOP')
GROUP BY position
ORDER BY position;
```

```
+-------------+---------------+
| position    | champion_name |
+-------------+---------------+
| DUO_CARRY   | Caitlyn       |
| DUO_SUPPORT | Thresh        |
| JUNGLE      | Lee Sin       |
| MID         | Ahri          |
| TOP         | Riven         |
+-------------+---------------+
5 rows in set (4.17 sec)
```

7. 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**

```sql
SELECT *
FROM (
    SELECT
        p.position AS position,
        c.champion_name AS champion_name,
        ((SUM(s.kills) + SUM(s.assists)) / SUM(s.deaths)) AS kda
    FROM stat AS s
    JOIN participant AS p ON s.player_id = p.player_id
    JOIN champ AS c ON c.champion_id = p.champion_id
    GROUP BY c.champion_id, p.position
    HAVING SUM(s.deaths) > 0
    ORDER BY position, kda DESC
) AS tmp
WHERE position IN ('DUO_CARRY', 'DUO_SUPPORT', 'JUNGLE', 'MID', 'TOP')
GROUP BY position
ORDER BY position;
```

```
+-------------+---------------+---------+
| 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 (12.64 sec)
```

8. 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"**

```sql
SELECT champion_name FROM champ
WHERE champion_id NOT IN (
    SELECT DISTINCT c.champion_id
    FROM teamban AS t
    JOIN match_info AS m ON t.match_id = m.match_id
    JOIN champ AS c on t.champion_id = c.champion_id
    WHERE SUBSTRING_INDEX(m.version, '.', 2) = '7.7'
)
ORDER BY champion_name;
```

```
+---------------+
| champion_name |
+---------------+
| Kayn          |
| Ornn          |
| Rakan         |
| RekSai        |
| Sion          |
| Xayah         |
+---------------+
6 rows in set (0.13 sec)
```

9. syntax in usage of SQL. There is a slogan said by a famous streamer: If you pick Lee Sin, I will pick Teemo. 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. You need to sort version in alphabetical order as following example like 4.1, 4.10, 4.2, 4.3.

```sql
SELECT
    c.champion_name AS self_champ_name,
    SUM(target.team=0) / COUNT(*) AS win_ratio,
    (SUM(s.kills) + SUM(s.assists)) / SUM(s.deaths) AS self_kda,
    AVG(s.goldearned) AS seld_avg_gold,
    'Renekton' AS enemy_role,
    (SUM(target.kills) + SUM(target.assists)) / SUM(target.deaths) AS enemy_kills,
    AVG(target.goldearned) AS enemy_avg_gold,
    count(*) AS battle_record
FROM participant AS p
JOIN stat AS s ON p.player_id = s.player_id
JOIN champ AS c ON c.champion_id = p.champion_id
JOIN (
    SELECT
        p.match_id AS match_id,
        s.win AS team,
        s.kills AS kills,
        s.assists AS assists,
        s.deaths AS deaths,
        s.goldearned AS goldearned
    FROM stat AS s
    JOIN participant AS p ON s.player_id = p.player_id
    WHERE p.champion_id = 58 AND
        p.position = 'TOP' AND
        s.deaths > 0
) AS target
    ON p.match_id = target.match_id AND
        1 - s.win = target.team
GROUP BY p.champion_id
HAVING SUM(s.deaths) > 0 AND SUM(target.deaths) > 0 AND battle_record > 100
ORDER BY battle_record DESC
LIMIT 5;
```

version	win_cnt	lose_cnt	win_ratio
7.9	527	464	0.5318
7.7	32	29	0.5246
6.9	1	1	0.5000
6.1	0	1	0.0000
5.21	0	2	0.0000
5.15	0	1	0.0000
4.15	1	1	0.5000
4.10	2	1	0.6667
7.8	210	237	0.4698
7.6	2	5	0.2857
7.5	2	2	0.5000
7.10	282	304	0.4812
6.20	3	2	0.6000
6.18	1	1	0.5000
4.21	1	1	0.5000
4.19	0	1	0.0000
6.24	4	3	0.5714
6.2	1	1	0.5000
5.24	1	1	0.5000
4.12	0	1	0.0000
5.6	0	1	0.0000
6.23	3	2	0.6000
6.21	0	2	0.0000
5.1	1	2	0.3333
4.18	0	1	0.0000
7.4	1	1	0.5000
7.3	0	1	0.0000
7.2	2	1	0.6667
6.22	2	1	0.6667
5.13	0	1	0.0000
4.17	0	1	0.0000
6.6	0	1	0.0000
5.19	1	0	1.0000
5.12	1	0	1.0000
6.19	1	0	1.0000
6.8	1	0	1.0000
5.7	1	0	1.0000
5.20	2	0	1.0000
4.9	1	0	1.0000
6.14	1	0	1.0000
6.5	1	0	1.0000
5.5	1	0	1.0000
6.13	1	0	1.0000

43 rows in set (1.32 sec)

10. In LoL, every champion may be more easily defeated by specific champions, which is called “counter”. 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 Gragas. Note that you only need to consider the number of matches of each champion facing Gragas on TOP larger than 100.

```sql
SELECT
    c.champion_name AS self_champ_name,
    SUM(target.team=0) / COUNT(*) AS win_ratio,
    (SUM(s.kills) + SUM(s.assists)) / SUM(s.deaths) AS self_kda,
    AVG(s.goldearned) AS seld_avg_gold,
    'Renekton' AS enemy_role,
    (SUM(target.kills) + SUM(target.assists)) / SUM(target.deaths) AS enemy_kills,
    AVG(target.goldearned) AS enemy_avg_gold,
    count(*) AS battle_record
FROM participant AS p
JOIN stat AS s ON p.player_id = s.player_id
JOIN champ AS c ON c.champion_id = p.champion_id
JOIN (
    SELECT
        p.match_id AS match_id,
        s.win AS team,
        s.kills AS kills,
        s.assists AS assists,
        s.deaths AS deaths,
        s.goldearned AS goldearned
    FROM stat AS s
    JOIN participant AS p ON s.player_id = p.player_id
    WHERE p.champion_id = 58 AND
        p.position = 'TOP' AND
        s.deaths > 0
) AS target
    ON p.match_id = target.match_id AND
        1 - s.win = target.team
GROUP BY p.champion_id
HAVING SUM(s.deaths) > 0 AND SUM(target.deaths) > 0 AND battle_record > 100
ORDER BY battle_record DESC
LIMIT 5;
```

```
+----------------+-----------+----------+--------------+------------+-------------+----------------+--------------+
| self_champ_name | win_ratio | self_kda | seld_avg_gold | enemy_role | enemy_kills | enemy_avg_gold | battle_record |
+----------------+-----------+----------+--------------+------------+-------------+----------------+--------------+
| Lee Sin        |    0.4739 |   2.4494 |    11329.9710 | Renekton   |      2.0260 |     12157.8835 |          2859 |
| Caitlyn        |    0.5194 |   2.4856 |    12597.4094 | Renekton   |      2.0387 |     12079.9608 |          2809 |
| Lucian         |    0.5219 |   2.7194 |    12772.0047 | Renekton   |      2.0168 |     12060.3545 |          2767 |
| Thresh         |    0.5123 |   3.0729 |     9113.9899 | Renekton   |      2.0431 |     12035.0706 |          2280 |
| Ahri           |    0.5399 |   2.7933 |    12190.9901 | Renekton   |      1.9247 |     12098.6379 |          1715 |
+----------------+-----------+----------+--------------+------------+-------------+----------------+--------------+
5 rows in set (5.03 sec)
```

11. If you want to play the "TOP" position and get the maximum win rate, how will you choose your summoner spells (ss1 and ss2)? (Answer by your own view)

```sql
SELECT 'Flash/Ignite' AS skills, SUM(win=1) / SUM(win=0) AS ratio
FROM (
    SELECT s.win AS win
    FROM stat AS s
    JOIN participant AS p ON s.player_id = p.player_id
    WHERE ((p.ss1 = 'FLASH' AND p.ss2 = 'IGNITE') OR (p.ss2 = 'FLASH' AND p.ss1 = 'IGNITE'))
    AND p.position = 'TOP'
) AS temp
UNION
SELECT 'Flash/Teleport' as skills, SUM(win=1) / SUM(win=0) as ratio
FROM (
    SELECT s.win as win
    FROM stat AS s
    JOIN participant AS p ON s.player_id = p.player_id
    WHERE ((p.ss1 = 'FLASH' AND p.ss2 = 'TELEPORT') OR(p.ss2 = 'FLASH' AND p.ss1 = 'TELEPORT'))
    AND p.position = 'TOP'
) AS temp;
```

```
+----------------+--------+
| skills         | ratio  |
+----------------+--------+
| Flash/Ignite   | 1.0833 |
| Flash/Teleport | 0.9993 |
+----------------+--------+
2 rows in set (4.95 sec)
```

12. Feel free to think any valuable observation with explanation.

I found that when I use different code to run the query may have small differences in running time. In the future, I would like to try different ways to write other problem.