Student Name: ______________________________

Student ID: ________________________________

CSC3170 Introduction to Database Systems (2023-24 Term 2) Assignment 3

Submission deadline: before 12 April 2024 11:59 pm

General Guidelines:

- Please submit your solutions via Blackboard.
- Do not close your browser of the app before you have successfully uploaded your files. It is your own responsibility of keeping your file integrity.
- If you have any questions about this assignment, contact TA at ytyang@cse.cuhk.edu.hk.

---

1. Introduction

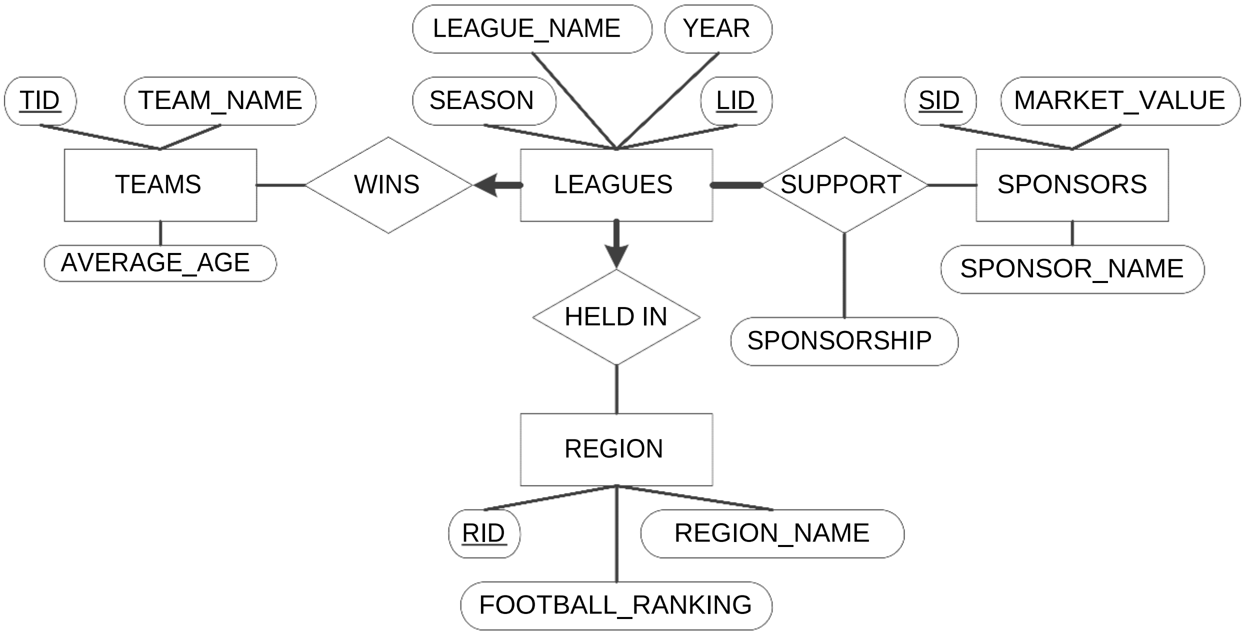
The International Football Organization keeps a record of all the football leagues, sponsors and football teams in different regions all over the world. In any region, there can be many leagues happening within the year in different seasons. Each league can be uniquely determined by its league ID (LID). Support of the leagues, which can be uniquely determined by their sponsor ID (SID) together with their amount of sponsorship, are kept in the database for future reference. In addition, the database also records the champion team of all the leagues, which can be uniquely determined by their team ID (TID).

2. Schema

The relational database schema is shown as follows:

```
TEAMS(TID, TEAM_NAME, AVERAGE_AGE)
LEAGUES(LID, LEAGUE_NAME, CHAMPION_TID, YEAR, SEASON, RID)
SPONSORS(SID, SPONSOR_NAME, MARKET_VALUE)
REGIONS(RID, REGION_NAME, FOOTBALL_RANKING)
SUPPORT(LID, SID, SPONSORSHIP)
```

3. ER-Diagram



4. Description

TEAMS - It stores information about the teams.

Item Name	Format	Description
TID	Integer	The ID of the team. It is unique.
TEAM_NAME	30 Char	The full name of the team. **It is also unique.**
AVERAGE_AGE	Float	The average age of players in the team.

LEAGUES - It stores information about the leagues.

Item Name	Format	Description
LID	Integer	The ID of the league. It is unique.
LEAGUE_NAME	30 Char	The full name of the league.
CHAMPION_TID	Integer	The ID of the champion team of this league.
YEAR	Integer	The year when the league was held.
SEASON	10 Char	The season when the league was held, includes "Spring", "Summer", "Autumn" and "Winter".
RID	Integer	The ID of the region where the league was held.

SPONSORS - It stores information about the sponsors.

Item Name	Format	Description
SID	Integer	The ID of the sponsor. It is unique.
SPONSOR_NAME	30 Char	The name of the sponsor.
MARKET_VALUE	Float	The market value of the sponsor. (in million dollar)

REGIONS - It stores region information.

Item Name	Format	Description
RID	Integer	The ID of the region. It is unique.
REGION_NAME	30 Char	The name of the region.
FOOTBALL_RANKING	Integer	The ranking of the region team in the world.

SUPPORT - It shows which sponsor supports which league.

Item Name	Format	Description
LID	Integer	The ID of the supported league.
SID	Integer	The ID of the sponsor.
SPONSORSHIP	Float	The total amount of money the sponsor supports. (in million dollar)

5. Queries (1 mark for each query)

You are required to write the queries below in SQL. Your queries will be tested under the db18 Oracle server in CSE department. You can execute **create_table.sql** to create all tables, and execute **add.sql** to load the test data(If you are using SQLWorkbench, try to execute **add_for_sql_workbench.sql** to load the test data). Please refer to Tutorial 7 for the information about connection to the Oracle Server.

1. Find the **REGION_NAME** of the regions and the **LID**, **LEAGUE_NAME** and **YEAR** of the leagues of all the leagues held in 'Spring' or 'Summer' **SEASON**. The result should be sorted by **LID** in ascending order.

   The ordering of the columns:

LID	LEAGUE_NAME	REGION_NAME	YEAR

2. Find the **TID**, **TEAM_NAME** and **AVERAGE_AGE** of the team that won leagues in 'Autumn' **SEASON** since **YEAR** 2015(inclusively) more than once. The result should be sorted by **TID** in ascending order.

   The ordering of the columns:

TID	TEAM_NAME	AVERAGE_AGE

3. Find the **TID**, **TEAM_NAME**, **AVERAGE_AGE** which won most of the leagues in each **SEASON**, and show the number of leagues the team won (**W_NUM**) in that season. The result should be ordered by the **TID, SEASON** in ascending order.

   The ordering of the columns:

TID	TEAM_NAME	AVERAGE_AGE	SEASON	W_NUM

4. Find the **SID**, **SPONSOR_NAME** and the corresponding number of leagues (**L_NUM**) supported by each sponsor. The result should be ordered by **SID** in ascending order, and you only need to show the top-5 records in the result.

   The ordering of the columns:

SID	SPONSOR_NAME	L_NUM

   (Note: you may need to use a pseudocolumn called ROWNUM. For detail, please refer to http://docs.oracle.com/cd/B19306_01/server.102/b14200/pseudocolumns009.htm)

5. Find the **LID**, **LEAGUE_NAME** of the league(s) held in 'Autumn' or 'Winter' **SEASON**, supported by at least one sponsor with **MARKET_VALUE** > 50 and won by team with **AVERAGE_AGE** < 30. The result should be ordered by **LID** in descending order.

   The ordering of the columns:

LID	LEAGUE_NAME

6. We define the **HOT** to a region of a sponsor as the value calculated by the following equation

$$\frac{\text{total } \mathbf{SPONSORSHIP} \text{ to the region by the sponsor}}{\sqrt{\text{the sponsor's } \mathbf{MARKET\_VALUE}} * \log_2(\sqrt{\mathbf{FOOTBALL\_RANKING}} + \mathbf{1})}$$

   List the **SID**, **HOT** of the sponsor who satisfies **MARKET_VALUE** > 40 and has the highest **HOT** among regions with **FOOTBALL_RANKING** < 10. The result should be ordered by **SID** in descending order.

   The ordering of the columns:

SID	HOT

   (Note: You may consider using SQRT, LOG in your SQL statement. For detail, please refer to https://docs.oracle.com/database/121/SQLRF/functions182.htm#SQLRF06110 https://docs.oracle.com/database/121/SQLRF/functions105.htm#SQLRF00661)

7. List the **HOT** to each region of sponsors with **SID** = 4,5,6,7, and also the highest hot (**HOT_HIGH**) among the four sponsors. The result should be sorted by **RID** in descending order.

   The ordering of the columns:

RID	HOT_4	HOT_5	HOT_6	HOT_7	HOT_HIGH

   Note:

- **HOT_i** is the **HOT** to the region of sponsor with **SID=i**.
- **HOT_i** is **NULL**(instead of 0) iff the sponsor with **SID=i** has never supported the region.
- When computing **HOT_HIGH**, the **NULL** values in **HOT_i** should be regarded as **0**.
- You may consider using GREATEST, NVL in your SQL statement. For detail, please refer to https://docs.oracle.com/database/121/SQLRF/functions078.htm#SQLRF00645 https://docs.oracle.com/database/121/SQLRF/functions131.htm#SQLRF00684)

8. We define the most competitive team(s) as the team which won the maximum number of leagues. Find the **SID**, **SPONSOR_NAME** of the sponsor(s) who have sponsored at least one league won by one of the most competitive team(s). The result should be sorted by **SID** in ascending order.

   The ordering of the columns:

SID	SPONSOR_NAME

(Please follow the Submission Procedure in next page.)

6. Submission Procedure

You should follow this procedure to submit all your SQL queries **STRICTLY** or you may receive mark deduction. Assume your name is "Jack" and your student ID is 1101234567. The submission procedures are shown as follows:

1. Write your queries to single file called **<your_student_ID>.sql** (e.g. 1101234567.sql) for all of the above queries and save the query results to the files result1.lst, result2.lst, …, result8.lst for queries 1, 2, …, and 8 respectively using the Spool command in Oracle (see the example shown below). You don't need to worry about that the SQLWorkBench cannot run Spool well, just make sure the correctness of your SQL statements.

You should use comment lines to include your name and student ID at the header of 1101234567.sql. You should also use the Oracle command Spool for each of the queries. Do NOT add any comment lines inside your SQL statements. There is always at least one space between your comment body and /* (or */). Your 1101234567.sql should be in the following format:

```
/*
    Student ID: 1101234567
    Name: jack
*/
/* Query 1 */
Spool result1.lst
Select … from … ;
Spool off
/* Query 2 */
Spool result2.lst
Select … from … ;
Spool off
……
```

```
If you need to create views, DO NOT write the create and drop sql statements inside
the body of Spool. The format should be:
/* Query 8 */
Create OR Replace view temp AS …
Spool result8.lst
Select … from … ;
Spool off
Drop view temp;
```

Please use an Unix text editor (e.g. vim) instead of a Windows editor, or you should ensure that your submitted file should not contain any special characters (e.g. ^M), which are resulted from transferring your files from Windows to Unix. You can solve this by using a Unix command dos2unix on *linux* machines.

2. You should test your final .sql file (e.g. 1101234567.sql) before submission by typing the command "@<your_student_ID>" (e.g. @1101234567) in your Oracle account. This should generate the result files result1.lst, result2.lst, …, result8.lst in your current directory in Unix. You have to ensure that the content of each result file is correct in order to get score for the query. **IMPORTANT!!!**

3. Submit your .sql file to the submission box on the blackboard platform.