PRACTICAL DATABASE REPORT

ASSIGNMENT 3:

Course: **PRACTICAL DATABASE CONCEPTS – ISYS3414**

Name: **Nguyen Dang Huynh Chau - s3777214**

Date: **06/05/2020**

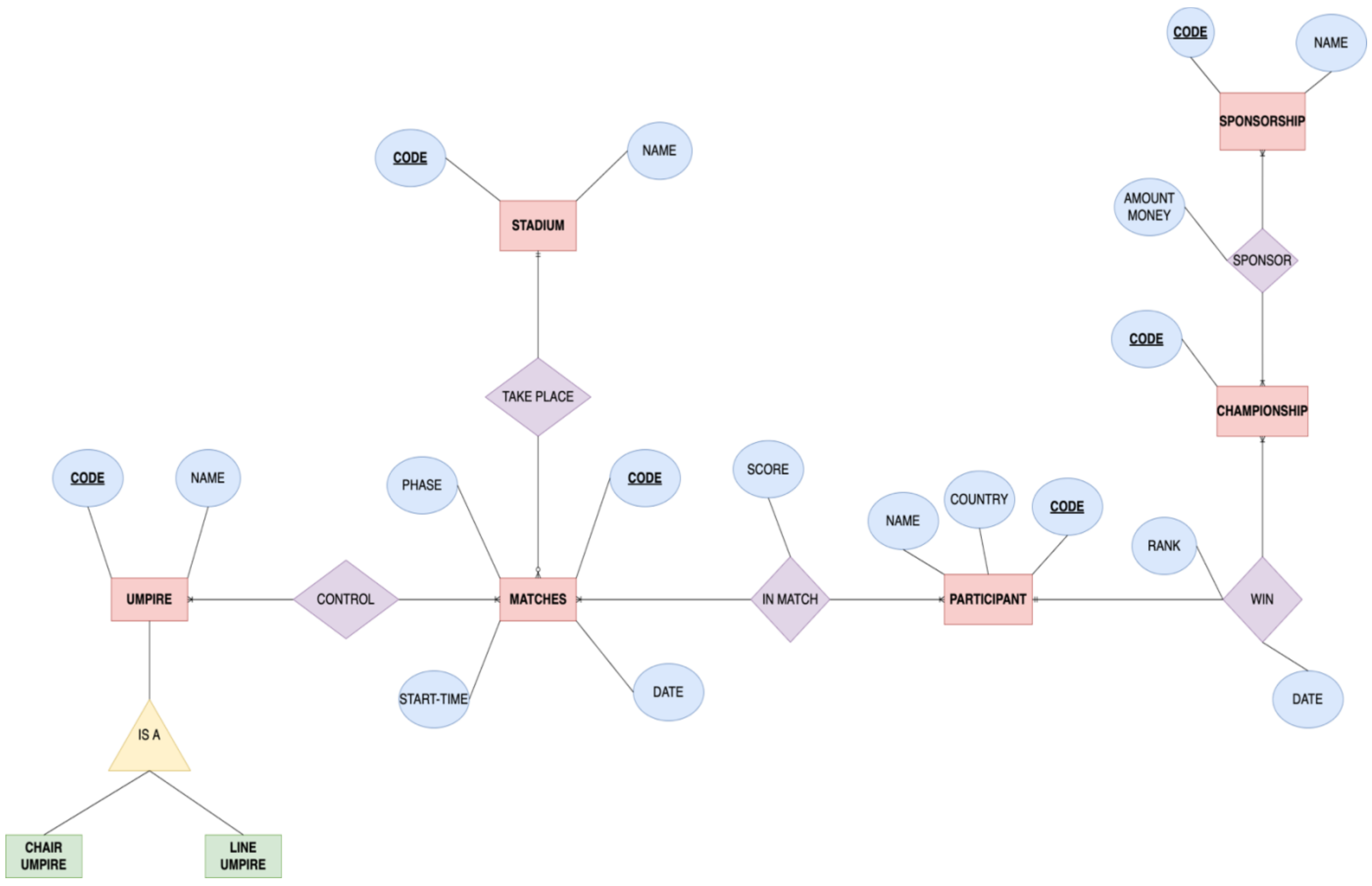
A description of sport tournament:

The ATP is the world’s tennis tournament for the male professionals. It takes place at a specific time of the year and at a different place worldwide every year and is sponsored. This Association has created a database to keep all the information of everything involved in all the events.

ATP tournament has 7 matches which are 4 quater final, 2 semi final and a final. Matches are held in different stadium. The officials are responsible for provide for spectators with a schedule for the whole tournament before its opening. Each match has 2 players participating and 2 umpire (chair and line umpire) who controls the match. Every data about the participants will be stored and as well the result of that match.

The championship has first, second, third, forth and fifth rankings for the players who got the highest, second, third, forth and fith highest scores. Each championship is sponsored by a variety of sponsorship

ERD:



Assumption of the assignment:

- Assume that no participant has the same rank

Relational Schema:

Participant (Code, Name, Country**)**

Stadium (Code, Name**)**

Matches (Code, Phase, Start_time, Date, **Stadium.code** Stadium**)**

In_match (Matches.code Matches, **Partcipant.code Participant**, Score**)**

Umpire (Code, Name**)**

Line_umpire (Umpire.code Code, Name**)**

Chair_umpire (Umpire.code Code, Name**)**

Control (Umpire.code Umpire, Matches.code Matches)

Championship (Code, **Participant.code** Participant ,Ranking, Date**)**

Sponsorship (Code, Name**)**

Sponsor(Sponsorship.code Sponsorship, **Championship.code Championship**,

Amount_of_money**)**

Create and Insert table query:

Create table query:

```
DROP DATABASE IF EXISTS ATP_Tour;
CREATE DATABASE ATP_Tour;
USE ATP_Tour;

SET FOREIGN_KEY_CHECKS=0;

-- Create table Participant (Code, Name, Country)
CREATE TABLE Participant (
    Code VARCHAR(5),
    Name VARCHAR(200),
    Country VARCHAR(200),
    CONSTRAINT pk_Participant
    PRIMARY KEY (code),
    CHECK (code REGEXP '^[A-Z]{2}[[:digit:]]{3}$'),
    CHECK (TRIM(name) > ''),
    CHECK (TRIM(country) > '')
                    );

-- Create table Stadium (Code, Name)
CREATE TABLE Stadium (
          Code VARCHAR(5),
    Name VARCHAR(200),
    CONSTRAINT pk_Stadium
    PRIMARY KEY (code),
    CHECK (code REGEXP '^[A-Z]{2}[[:digit:]]{3}$'),
    CHECK (TRIM(name) > '')
                 );

-- Create table Matches (Code, Phase, Start_time, Date, Stadium.code Stadium)
CREATE TABLE Matches (
    Code VARCHAR(5),
    Phase VARCHAR(200),
    Start_time TIME(0),
    Date DATE,
    Stadium VARCHAR(5),
    CONSTRAINT pk_Matches
    PRIMARY KEY (Code),
    CONSTRAINT fk_Matches_Stadium
    FOREIGN KEY (Stadium)
    REFERENCES Stadium (code) ON DELETE CASCADE ON UPDATE CASCADE,
    CHECK (code REGEXP '^[A-Z]{3}[[:digit:]]{2}$'),
    CHECK (TRIM(phase) > '')
                 );

-- Create table In_match (Matches.code Matches, Partcipant.code Participant, Score)
CREATE TABLE In_match (
    Matches VARCHAR(5),
    Participant VARCHAR(5),
    Score INT,
    CONSTRAINT fk_In_match_Matches
    FOREIGN KEY (Matches)
    REFERENCES Matches (code) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT pk_In_match
    PRIMARY KEY (Matches, Participant),
    CONSTRAINT fk_In_match_Participant
    FOREIGN KEY (Participant)
    REFERENCES Participant (code) ON DELETE CASCADE ON UPDATE CASCADE,
    CHECK (Score >= 0)
                  );

-- Create table Umpire (Code, Name) code: 2 first letter name, 3 number
CREATE TABLE Umpire (
    Code VARCHAR(5),
    Name VARCHAR(200),
    CONSTRAINT pk_Umpire
    PRIMARY KEY (code),
    CHECK (Code REGEXP '^(LI|CH)[[:digit:]]{3}$'),
    CHECK (TRIM(name) > '')
                );
```

```
-- Create table Line_umpire (Umpire.code Code, Name)
CREATE TABLE Line_umpire (
    Code VARCHAR(5),
    Name VARCHAR(200),
    CONSTRAINT pk_Line_umpire
    PRIMARY KEY (code),
    CONSTRAINT fk_Line_umpire_Umpire
    FOREIGN KEY (code)
    REFERENCES Umpire (code) ON DELETE CASCADE ON UPDATE CASCADE,
    CHECK (TRIM(name) > '')
                    );

-- Create table Chair_umpire (Umpire.code Code, Name)
CREATE TABLE Chair_umpire (
    Code VARCHAR(5),
    Name VARCHAR(200),
    CONSTRAINT pk_Chair_umpire
    PRIMARY KEY (code),
    CONSTRAINT fk_Chair_umpire_Umpire
    FOREIGN KEY (code)
    REFERENCES Umpire (code) ON DELETE CASCADE ON UPDATE CASCADE,
    CHECK (TRIM(name) > '')
               );

-- Create table Control (Umpire.code Umpire, Matches.code Matches)
CREATE TABLE Control (
    Umpire VARCHAR(5),
    Matches VARCHAR(5),
    CONSTRAINT pk_Control
    PRIMARY KEY (Umpire, Matches),
    CONSTRAINT fk_Control_Umpire
          FOREIGN KEY (Umpire)
    REFERENCES Umpire (code) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_Control_Matches
    FOREIGN KEY (Matches)
    REFERENCES Matches (code) ON DELETE CASCADE ON UPDATE CASCADE
                 );

-- Create table Championship (Code, Participant.code Participant, Ranking, Date) code: 2
first letter name, 3 number
CREATE TABLE Championship (
          Code VARCHAR(5),
    Participant VARCHAR(5),
    Ranking INT,
    DATE DATE,
    CONSTRAINT pk_Championship
    PRIMARY KEY (code),
    CONSTRAINT fk_Championship_Participant
    FOREIGN KEY (Participant)
    REFERENCES Participant (code) ON DELETE CASCADE ON UPDATE CASCADE,
    CHECK (0 < Ranking < 6),
    CHECK (code REGEXP '^[A-Z]{2}[[:digit:]]{3}$')
                     );
```

```
-- Create table Sponsorship (Code, Name)
CREATE TABLE Sponsorship (
    Code VARCHAR(5),
    Name VARCHAR(200),
    CONSTRAINT pk_Sponsorship
    PRIMARY KEY (Code),
    CHECK (code REGEXP '^[A-Z]{2}[[:digit:]]{3}$'),
    CHECK (TRIM(name) > '')
                    );

-- Create table Sponsor(Sponsorship.code Sponsorship, Championship.code Championship,
Amount_of_money)
CREATE TABLE Sponsor (
          Sponsorship VARCHAR(5),
    Championship VARCHAR(5),
    Amount_of_money FLOAT,
    CONSTRAINT pk_Sponsor
    PRIMARY KEY (Sponsorship, Championship),
    CONSTRAINT fk_Sponsor_Sponsorship
    FOREIGN KEY (Sponsorship)
    REFERENCES Sponsorship (code) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_Sponsor_Championship
    FOREIGN KEY (Championship)
    REFERENCES Championship (code) ON DELETE CASCADE ON UPDATE CASCADE,
    CHECK (Amount_of_money > 0)
                );

COMMIT;
SET FOREIGN_KEY_CHECKS=1;
```

Insert table query:

```
USE ATP_Tour;

-- Insert into Participant (Code, Name, Country)
INSERT INTO Participant VALUES ('RF001', 'Roger Federer', 'Switzerland'),
                            ('RN002', 'Rafael Nadal', 'Spain'),
                            ('ND003', 'Novak Djokovic', 'Serbia'),
                            ('AM004', 'Andy Murray', 'United Kingdom'),
                            ('DT005', 'Dominic Thiem', 'Austria'),
                            ('PS006', 'Pete Sampras', 'United States'),
                            ('SW007', 'Stan Wawrinka', 'Switzerland'),
                            ('AZ008', 'Alexander Zverev', 'Germany');

-- Insert into Stadium (Code, Name)
INSERT INTO Stadium VALUES ('CC001', 'Centre Court'),
                        ('AA002', 'Arthur Ashe Stadium'),
                        ('CP003', 'Court Philippe Chatrier'),
                        ('IW004', 'Indian Wells Tennis Garden'),
                        ('RL005', 'Rod Laver Arena'),
                        ('QS006', 'Qizhong Stadium'),
                        ('RE007', 'Roy Emerson Arena'),
                        ('FI008', 'Foro Italico'),
                        ('RA009', 'Royal Albert Hall'),
                        ('OA010', 'The O2 Arena');

-- Insert into Matches (Code, Phase, Start_time, Date, Stadium.code Stadium)
INSERT INTO Matches VALUES ('QFA01', 'Quater Final A', '09:00', '2020-01-20',
'CC001'),
                                                            ('QFB02', 'Quater Final B', '09:00', '2020-
01-25', 'CC001'),
                        ('QFC03', 'Quater Final C', '09:00', '2020-01-30', 'AA002'),
                        ('QFD04', 'Quater Final D', '09:00', '2020-02-05', 'AA002'),

                        ('SFA01', 'Semi Final A', '10:00', '2020-03-10', 'CP003'),
                        ('SFB02', 'Semi Final B', '10:00', '2020-03-20', 'CP003'),

                        ('FIN01', 'Final', '09:30', '2020-03-20', 'IW004');

-- Insert into In_match (Matches.code Matches, Partcipant.code Participant, Score)
INSERT INTO In_match VALUES ('QFA01', 'RF001', 10),
                                                                   ('QFA01', 'RN002', 20),

                                                                   ('QFB02', 'ND003', 30),
                         ('QFB02', 'AM004', 15),

                         ('QFC03', 'DT005', 40),
                         ('QFC03', 'PS006', 25),

                         ('QFD04', 'SW007', 25),
                         ('QFD04', 'AZ008', 10),

                         ('SFA01', 'RN002', 25),
                         ('SFA01', 'ND003', 30),

                         ('SFB02', 'DT005', 15),
                         ('SFB02', 'SW007', 40),

                         ('FIN01', 'SW007', 12),
                         ('FIN01', 'ND003', 14);

-- Insert into Umpire (Code, Name)
INSERT INTO Umpire VALUES ('CH001', 'Ali Nili'),
                        ('CH002', 'Carlos Bernardes'),
                        ('CH003', 'Carlos Ramos'),
                        ('CH004', 'Damien Dumusois'),
                        ('CH005', 'Emmanuel Joseph'),
                        ('CH006', 'Fergus Murphy'),

                        ('LI001', 'Gianluca Moscarella'),
                        ('LI002', 'James Keothavong'),
                        ('LI003', 'John Blom'),
                        ('LI004', 'Kader Nouni'),
                        ('LI005', 'Manuel Messina'),
                        ('LI006', 'Mohamed El Jennati');

-- Insert into Line_umpire (Umpire.code Code, Name)
INSERT INTO Line_umpire VALUES ('LI001', 'Gianluca Moscarella'),
                            ('LI002', 'James Keothavong'),
                            ('LI003', 'John Blom'),
                            ('LI004', 'Kader Nouni'),
                            ('LI005', 'Manuel Messina'),
                            ('LI006', 'Mohamed El Jennati');

-- Insert into Chair_umpire (Umpire.code Code, Name)
INSERT INTO Chair_umpire VALUES ('CH001', 'Ali Nili'),
                             ('CH002', 'Carlos Bernardes'),
                             ('CH003', 'Carlos Ramos'),
                             ('CH004', 'Damien Dumusois'),
                             ('CH005', 'Emmanuel Joseph'),
                             ('CH006', 'Fergus Murphy');

-- Insert into Control (Umpire.code Umpire, Matches.code Matches)
INSERT INTO Control VALUES  ('CH001', 'QFA01'),
                                                                     ('LI001', 'QFA01'),

                                                                     ('LI001', 'QFB02'),
                         ('CH001', 'QFB02'),

                         ('CH002', 'QFC03'),
                         ('LI002', 'QFC03'),

                         ('CH002', 'QFD04'),
                         ('LI002', 'QFD04'),

                         ('CH003', 'SFA01'),
                         ('LI003', 'SFA01'),

                         ('CH003', 'SFB02'),
                         ('LI003', 'SFB02'),

                         ('CH004', 'FIN01'),
                         ('LI004', 'FIN01');

-- Insert into Championship (Code, Participant.code Participant ,Ranking, Date)
INSERT INTO Championship VALUES ('FR001', 'ND003', 1, '2020-03-20'),
                                                                               ('SR002', 'SW007', 2, '2020-
03-20'),
                             ('TR003', 'RN002', 3, '2020-03-20'),
                             ('FR004', 'DT005', 4, '2020-03-20'),
                             ('FR005', 'PS006', 5, '2020-03-20');

-- Insert into Sponsorship (Code, Name)
```

```
INSERT INTO Sponsorship VALUES ('AS001', 'Asics'),
                               ('FI002', 'Fila'),
                               ('NB003', 'New Balance'),
                               ('KS004', 'K-Swiss'),
                               ('BP005', 'BNP Paribas'),
                               ('HE006', 'Head');

-- Insert into Sponsor(Sponsorship.code Sponsorship, Championship.code
Championship, Amount_of_money)
INSERT INTO Sponsor VALUES ('AS001', 'FR001', 10000),
                           ('FI002', 'FR001', 10000),
                           ('NB003', 'FR001', 10000),

                           ('KS004', 'SR002', 10000),
                           ('BP005', 'SR002', 10000),

                           ('HE006', 'TR003', 1000),
                           ('AS001', 'TR003', 1000),

                           ('FI002', 'FR004', 1000),
                           ('NB003', 'FR004', 1000),

                           ('KS004', 'FR005', 1000),
                           ('BP005', 'FR005', 1000);
```

Description of 10 queries and the SQL commands of those queries.

```
USE ATP_Tour;

-- Select a list of name of participant and their score in the increasing order of their
score ( use ORDER BY)
SELECT P.name, I.score
FROM Participant P, In_match I
WHERE P.code = I.participant
ORDER BY Score ASC;

-- Select a list of name of players in match (use INNER JOINS)
SELECT I.matches, P.Name AS "Participant"
FROM In_match I
INNER JOIN Participant P ON I.participant = P.code;

--  Number of matches in this Tournament (A query that uses aggregate functions)
SELECT COUNT(*)
FROM Matches M;

-- Select list of stadium which have at least once held a tournament and the number
of match
-- (A query that uses the GROUP BY and HAVING clauses)
```

```sql
SELECT S.Name AS 'Stadium', COUNT(M.code) AS 'Number of match'
FROM (Matches M
INNER JOIN Stadium S ON M.Stadium = S.code)
GROUP BY M.Stadium
HAVING COUNT(M.code) >= 1;

-- List of particioant have score which is higher than the average (A query that uses a
sub-query as a relation)
SELECT DISTINCT P.code, P.name, I.score
FROM (SELECT AVG(I.score) AS averageScore
    FROM In_match I) AS Score, Participant P, In_match I
WHERE I.score > Score.averageScore
AND I.participant = P.code;

-- Select information of any Umpire who has least controlled a match (A query that
uses a sub-query in the WHERE clause)
SELECT U1.code, U1.name
FROM Umpire U1
WHERE U1.code IN (SELECT C.Umpire
              FROM Umpire U2, Control C
              WHERE C.Umpire = U2.code
             );

--  Create a view to illustrate the name and code of 5 champion with their rank (A query
stored as a VIEW)
CREATE VIEW Champion AS
SELECT P.code, P.name AS 'Participant', C.ranking
FROM Participant P, Championship C
WHERE C.participant = P.code;

-- Select the information of the participants whose names start with A (A query that
uses LIKE in the WHERE clause)
SELECT Participant.*
FROM Participant
WHERE Participant.Name LIKE 'A%';

-- Select a list of match and its stadium where held the match (A query that uses a
self-JOIN)
SELECT M1.code AS 'Match', M1.stadium AS 'Stadium'
FROM Matches M1
INNER JOIN Matches M2
ON M1.code = M2.code;

-- The highest score with its participant name (A query that uses ALL or ANY)
SELECT Participant.code, Participant.Name, In_match.Score, In_match.matches
FROM Participant, In_match
WHERE Participant.code = In_match.participant
AND In_match.score >= ANY (SELECT MAX(In_match.score)
                      FROM In_match
                                             )
```

The link to web database application

https://apex.oracle.com/pls/apex/s3777214_assignment/r/assignment-s37772141/home?session=14175143077666

Id: PROFESSOR
Password: professor123