THOMAS MORE

FIRST NAME NAME

SQL Project

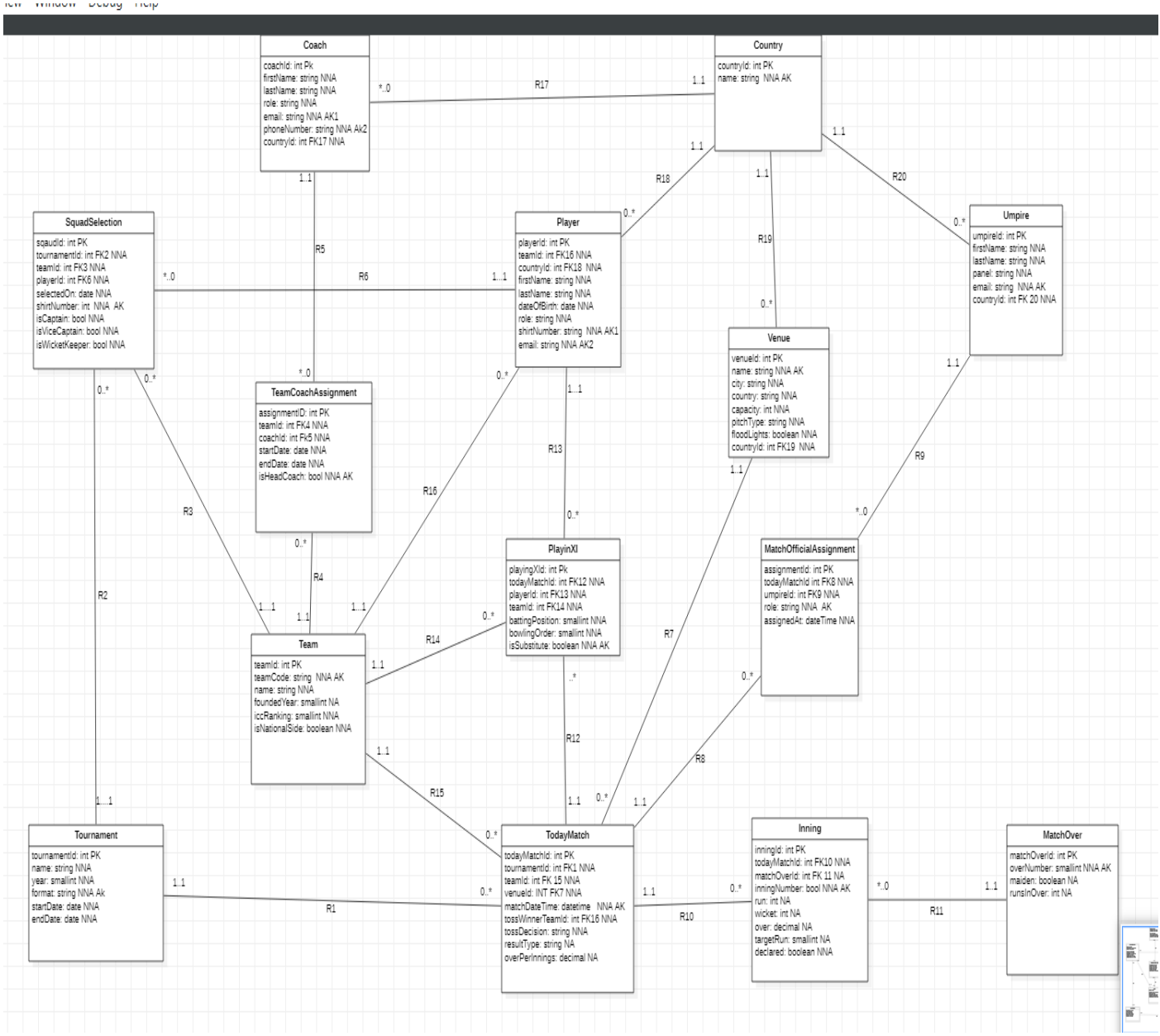
Class group

2025 - 2026

Part	Score
Part 1: ERD / Datamodel	… / 15
Part 2: Creating a database	… / 5
Part 3: Fill in the data	… / 5
Part 4: Where and scalar	… / 5
Part 5: Joins	… / 5
Part 6: Subqueries	… / 5
Part 7: Set functions	… / 2
Part 8: Correlated subqueries	… / 3
Part 9: Group by	… / 5
Total	**… / 50**

Inhoud

1. Assignment 1 – Data model



2. Assignment 2 – Creating a database

```
DROP TABLE IF EXISTS MatchOfficialAssignment;
DROP TABLE IF EXISTS PlayingXI;
DROP TABLE IF EXISTS SquadSelection;
DROP TABLE IF EXISTS Inning;
DROP TABLE IF EXISTS MatchOver;
DROP TABLE IF EXISTS TodayMatch;
DROP TABLE IF EXISTS TeamCoachAssignment;
DROP TABLE IF EXISTS Player;
DROP TABLE IF EXISTS Coach;
DROP TABLE IF EXISTS Umpire;
DROP TABLE IF EXISTS Team;
DROP TABLE IF EXISTS Venue;
DROP TABLE IF EXISTS Tournament;
DROP TABLE IF EXISTS Country;


 drop schema if exists sport;
 create schema if not exists sport;
 use sport;

/* 1st table*/
CREATE TABLE IF NOT EXISTS Country (

countryId int PRIMARY KEY ,
name varchar(50)  NOT NULL UNIQUE
);


/* 2nd table*/

CREATE TABLE IF NOT EXISTS Umpire (
    umpireId int NOT NULL    PRIMARY KEY,
    firstName varchar(50)   NOT NULL,
    lastName varchar(50)    NOT NULL,
    panel  varchar(40)      NOT NULL,
    eMail  varchar(200)     NOT NULL UNIQUE,
    countryId  int          NOT NULL,
    CONSTRAINT FK_Umpire_Country
    FOREIGN KEY (countryId) REFERENCES Country(countryId)

);
/* 3rd table*/

CREATE TABLE IF NOT EXISTS Coach(
 coachId  int NOT NULL      PRIMARY KEY,
 firstName  varchar(50)     NOT NULL,
 lastName  varchar(50)      NOT NULL,
 eMail  varchar(200)        NOT NULL UNIQUE,
 phoneNumber   varchar(300)  NOT NULL UNIQUE,
 countryId     int           NOT NULL,
  CONSTRAINT FK_Coach_Country
```

```
  FOREIGN KEY (countryId) REFERENCES Country(countryId)

);
/* 4th table*/

CREATE TABLE IF NOT EXISTS Venue(
  venueId int NOT NULL PRIMARY KEY,
  name   varchar(50) NOT NULL UNIQUE,
  city varchar(60) NOT NULL,
  country  varchar(50) NOT NULL,
  capacity  int  NOT NULL,
  pitchType varchar(50) NOT NULL,
  floodLights bool NOT NULL,
  countryId int NOT NULL,
  CONSTRAINT FK_Venue_Country
  FOREIGN KEY (countryId) REFERENCES country(countryId)

);

/* 5th table*/

 CREATE TABLE IF NOT EXISTS Team(
    teamId        int NOT NULL PRIMARY KEY,
    teamCode       int NOT NULL UNIQUE,
    name           varchar(60) NOT NULL,
    foundedYear     smallint NOT NULL,
    iccRanking     smallint NOT NULL,
    isNationalSide  bool NOT NULL,
    countryId       int NOT NULL,
    CONSTRAINT FK_Team_Country
    FOREIGN KEY (countryId) REFERENCES Country(countryId)
 );

 /* 6th table*/

 create table if not exists TeamCoachAssignment(
        assignmentId int NOT NULL PRIMARY KEY,
        startDate  date NOT NULL,
        endDate date NOT NULL,
        isHeadCoach bool NOT NULL UNIQUE,
        coachId int NOT NULL,
        teamId  int NOT NULL,
         CONSTRAINT FK_TeamCoachAssignment_Coach
    FOREIGN KEY (coachId) REFERENCES Coach(coachId),
    CONSTRAINT FK_TeamCoachAssignment_Team
    FOREIGN KEY (teamId) REFERENCES Team(teamId)

 );

 /* 7th table*/

create table if not exists Player(
    playerId     int NOT NULL     PRIMARY KEY,
    firstName     varchar(60)      NOT NULL,
    lastName      VARCHAR(50)      NOT NULL,
    dateOfBirth   date             NOT NULL,
```

```
    shirtNumber   varchar(50)      NOT NULL UNIQUE,
    role          varchar (50)     NOT NULL,
    eMail         varchar(200)     NOT NULL UNIQUE,
    teamId         int             NOT NULL,
    countryId      int             NOT NULL,
    CONSTRAINT FK_Player_Country
    FOREIGN KEY (countryId) REFERENCES Country(countryId),
    CONSTRAINT FK_Player_Team
    FOREIGN KEY (teamId) REFERENCES Team(teamId)
);


/* 8th table*/

create table if not exists Tournament(
    tournamentId int NOT NULL PRIMARY KEY,
    name  varchar (60) NOT NULL,
    year  year  NOT NULL,
    format varchar(50) NOT NULL UNIQUE,
    startDate date  NOT NULL,
    endDate  date NOT NULL

);

/* 12th table*/

create table if not exists TodayMatch(
     todayMatchId   int NOT NULL PRIMARY KEY,
       matchDateTime   datetime  NOT NULL UNIQUE ,
     resultType   varchar (50)  NOT NULL,
     tossDecision  varchar(50)  Not null,
     overPerInning   decimal (4,1)   NOT NULL,
     tournamentId    int NOT NULL,
     venueId        int NOT NULL,
     teamId         int NOT NULL,

      CONSTRAINT FK_TodayMatch_Tournament
      FOREIGN KEY (tournamentId) REFERENCES Tournament(tournamentId),

        CONSTRAINT FK_TodayMatch_Venue
      FOREIGN KEY (venueId) REFERENCES Venue(venueId),

        CONSTRAINT FK_TodayMatch_Team
      FOREIGN KEY (teamId) REFERENCES Team(teamId)

);

/* 9th table*/

create table if not exists SquadSelection(
      squadId     int NOT NULL PRIMARY KEY,
      selectedOn  date NOT NULL,
      shirtNumber  int NOT NULL UNIQUE,
      isCaptain  bool NOT NULL,
      isViceCaptain  bool NOT NULL,
      isWicketKeeper bool NOT NULL,
```

```
    tournamentId  int NOT NULL,
    playerId    int NOT NULL,
    teamId    int NOT NULL,

    CONSTRAINT FK_SquadSelection_Tournament
    FOREIGN KEY (tournamentId) REFERENCES Tournament(tournamentId),
    CONSTRAINT FK_SquadSelection_PlayerId
    FOREIGN KEY (PlayerId) REFERENCES Player(PlayerId),
     CONSTRAINT FK_SquadSelection_Team
    FOREIGN KEY (teamId) REFERENCES Team(teamId)
);



/* 11th table*/

create table if not exists PlayingXI(
    playingXId      int NOT NULL PRIMARY KEY,
    battingPossition smallint   NOT NULL,
    bowlingOrder   smallint NOT NULL,
    isSubstitute   bool NOT NULL UNIQUE,
    TodaymatchId      int NOT NULL,
    playerId     int NOT NULL,
    teamId       int NOT NULL,
    CONSTRAINT FK_PlayingXI_TodayMatch
    FOREIGN KEY (TodaymatchId) REFERENCES TodayMatch(TodayMatchId),

     CONSTRAINT FK_PlayingXI_Player
    FOREIGN KEY (PlayerId) REFERENCES Player(PlayerId),

     CONSTRAINT FK_PlayingXI_Team
    FOREIGN KEY (teamId) REFERENCES Team(teamId)


);


/* 10th table*/

 create table if not exists MatchOfficialAssignment(
        assignmentId       int  NOT NULL PRIMARY KEY,
        role               varchar(50)  NOT NULL UNIQUE,
        assignedAt         datetime NOT NULL,
        umpireId          int NOT NULL,
        todayMatchId     int NOT NULL,

          CONSTRAINT FK_MatchOfficialAssignment_Umpire
        FOREIGN KEY (umpirId) REFERENCES Umpire(umpirId),

          CONSTRAINT FK_MatchOfficialAssignment_TodayMatch
        FOREIGN KEY (TodaymatchId) REFERENCES TodayMatch(TodaymatchId)
 );

/* 13th table*/

create table if not exists  MatchOver(
```

```
    matchOverId int NOT NULL PRIMARY KEY,
    overNumber   int NOT NULL UNIQUE,
    maiden       bool  NULL,
    runsInOver   int NULL

);

/* 14th table*/

create table if not exists Inning(
    inningId      int NOT NULL PRIMARY KEY,
    inningNumber   int  NOT NULL UNIQUE,
    run            int NULL,
    wicket         int NULL,
    overs          decimal(4,1) NOT NULL,
    targetRun      smallint NOT NULL,
    declared       bool NULL,
    todayMatchId        int NOT NULL,
    matchOverId       int NOT NULL,

    CONSTRAINT FK_Inning_TodayMatch
    FOREIGN KEY (TodaymatchId) REFERENCES TodayMatch(TodaymatchId),

      CONSTRAINT FK_Inning_MatchOver
    FOREIGN KEY (matchOverId) REFERENCES MatchOver(matchOverId)
);
```

3. Assignment 3 – Fill in data

/* country table*/

```
INSERT INTO Country (countryId, name) VALUES (1, 'Belgium');
INSERT INTO Country (countryId, name) VALUES (2, 'Netherlands');
INSERT INTO Country (countryId, name) VALUES (3, 'France');
INSERT INTO Country (countryId, name) VALUES (4, 'Germany');
INSERT INTO Country (countryId, name) VALUES (5, 'Spain');
INSERT INTO Country (countryId, name) VALUES (6, 'Portugal');
INSERT INTO Country (countryId, name) VALUES (7, 'Italy');
INSERT INTO Country (countryId, name) VALUES (8, 'England');
INSERT INTO Country (countryId, name) VALUES (9, 'USA');
INSERT INTO Country (countryId, name) VALUES (10, 'Brazil');
INSERT INTO Country (countryId, name) VALUES (11, 'Argentina');
INSERT INTO Country (countryId, name) VALUES (12, 'Canada');
INSERT INTO Country (countryId, name) VALUES (13, 'Australia');
INSERT INTO Country (countryId, name) VALUES (14, 'Japan');
INSERT INTO Country (countryId, name) VALUES (15, 'Sweden');
```

/*team*/

```
INSERT INTO Umpire (umpirId, firstName, lastName, panel, eMail, countryId) VALUES
(1,  'Peter',   'Johnson',  'Elite',         'peter.johnson@umpires.com',      7),
(2,  'Mark',    'Vermeer',  'International', 'mark.vermeer@umpires.com',       11),
(3,  'Jean',    'Dupont',   'International', 'jean.dupont@umpires.com',        4),
(4,  'Hans',    'Müller',   'Elite',         'hans.mueller@umpires.com',       10),
(5,  'Carlos',  'Garcia',   'International', 'carlos.garcia@umpires.com',      14),
(6,  'Rui',     'Silva',    'Domestic',      'rui.silva@umpires.com',          6),
(7,  'Marco',   'Rossi',    'Elite',         'marco.rossi@umpires.com',        3),
(8,  'David',   'Brown',    'International', 'david.brown@umpires.com',        8),
(9,  'John',    'Smith',    'Domestic',      'john.smith@umpires.com',         9),
(10, 'Paulo',   'Souza',    'Elite',         'paulo.souza@umpires.com',        3),
(11, 'Diego',   'Lopez',    'International', 'diego.lopez@umpires.com',        2),
(12, 'Alex',    'Martin',   'Domestic',      'alex.martin@umpires.com',        12),
(13, 'Liam',    'Wilson',   'International', 'liam.wilson@umpires.com',        13),
(14, 'Kenji',   'Sato',     'Elite',         'kenji.sato@umpires.com',         5),
(15, 'Rahul',   'Sharma',   'International', 'rahul.sharma@umpires.com',       15);
```

/*coach table*/

```
INSERT INTO Coach (coachId, firstName, lastName, eMail, phoneNumber, countryId) VALUES
(1, 'Thomas',   'De Smet',    'thomas.desmet@coaches.com',    '+32-470-100001', 6),
(2, 'Jeroen',   'Bakker',     'jeroen.bakker@coaches.com',    '+31-6-20000002', 2),
(3, 'Claude',   'Martin',     'claude.martin@coaches.com',    '+33-6-30000003', 1),
(4, 'Stefan',   'Keller',     'stefan.keller@coaches.com',    '+49-151-4000004', 6),
(5, 'Miguel',   'Lopez',      'miguel.lopez@coaches.com',     '+34-600000005', 12),
(6, 'Joao',     'Ferreira',   'joao.ferreira@coaches.com',    '+351-910000006', 10),
```

```
(7,  'Luca',     'Bianchi',    'luca.bianchi@coaches.com',     '+39-3400000007', 7),
(8,  'Richard',  'Evans',      'richard.evans@coaches.com',    '+44-7700000008', 8),
(9,  'Michael',  'Taylor',     'michael.taylor@coaches.com',   '+1-555-0000009', 9),
(10, 'Roberto',  'Silva',      'roberto.silva@coaches.com',    '+55-21-90000010', 4),
(11, 'Javier',   'Gomez',      'javier.gomez@coaches.com',     '+54-9-110000011', 11),
(12, 'Daniel',   'Clark',      'daniel.clark@coaches.com',     '+1-604-70000012', 5),
(13, 'Ethan',    'Hughes',     'ethan.hughes@coaches.com',     '+61-410000013', 13),
(14, 'Taro',     'Kobayashi',  'taro.kobayashi@coaches.com',   '+81-80-70000014', 14),
(15, 'Vikram',   'Patel',      'vikram.patel@coaches.com',     '+91-980000015', 15);

/* venue  table*/
INSERT INTO Venue (venueId, name, city, country, capacity, pitchType, floodLights, countryId) VALUES
(1,  'King Baudouin Stadium',     'Brussels',     'Belgium',     50000, 'Grass',        TRUE,  5),
(2,  'Johan Cruyff Arena',        'Amsterdam',    'Netherlands', 55000, 'Hybrid',       TRUE,  8),
(3,  'Stade de France',           'Paris',        'France',      80000, 'Grass',        TRUE,  3),
(4,  'Olympiastadion Berlin',     'Berlin',       'Germany',     74000, 'Grass',        TRUE,  12),
(5,  'Santiago Bernabeu',         'Madrid',       'Spain',       81000, 'Hybrid',       TRUE,  1),
(6,  'Estadio da Luz',            'Lisbon',       'Portugal',    65000, 'Grass',        TRUE,  15),
(7,  'San Siro',                  'Milan',        'Italy',       80000, 'Grass',        TRUE,  7),
(8,  'Wembley Stadium',           'London',       'England',     90000, 'Hybrid',       TRUE,  2),
(9,  'MetLife Stadium',           'New York',     'USA',         82500, 'Artificial',   TRUE,  9),
(10, 'Maracanã',                  'Rio de Janeiro','Brazil',     78000, 'Grass',        TRUE,  4),
(11, 'Estadio Monumental',        'Buenos Aires', 'Argentina',   70000, 'Grass',        TRUE,  11),
(12, 'BC Place',                  'Vancouver',    'Canada',      54000, 'Artificial',   TRUE,  10),
(13, 'ANZ Stadium',               'Sydney',       'Australia',   83500, 'Grass',        TRUE,  13),
(14, 'Tokyo National Stadium',    'Tokyo',        'Japan',       68000, 'Hybrid',       TRUE,  14),
(15, 'Narendra Modi Stadium',     'Ahmedabad',    'India',       132000,'Grass',        TRUE,  6);

/*Team table*/
INSERT INTO Team (teamId, teamCode, name, foundedYear, iccRanking, isNationalSide, countryId)
VALUES
(1, 100, 'Belgium National Team',      1904, 20, TRUE, 8),
(2, 101, 'Netherlands National Team',  1889, 14, TRUE, 5),
(3, 102, 'France National Team',       1900, 3,  TRUE, 14),
(4, 103, 'Germany National Team',      1900, 4,  TRUE, 4),
(5, 104, 'Spain National Team',        1913, 2,  TRUE, 2),
(6, 105, 'Portugal National Team',     1914, 9,  TRUE, 11),
(7, 106, 'Italy National Team',        1898, 8,  TRUE, 7),
(8, 107, 'England National Team',      1863, 5,  TRUE, 1),
(9, 108, 'USA National Team',          1913, 11, TRUE, 9),
(10,109, 'Brazil National Team',       1914, 1,  TRUE, 10),
(11,110, 'Argentina National Team',    1893, 6,  TRUE, 6),
(12,111, 'Canada National Team',       1912, 23, TRUE, 12),
(13,112, 'Australia National Team',    1922, 25, TRUE, 13),
(14,113, 'Japan National Team',        1921, 18, TRUE, 3),
(15,114, 'India National Team',        1930, 22, TRUE, 15);

/*teamcoachassignment*/

INSERT INTO TeamCoachAssignment (assignmentId, startDate, endDate, isHeadCoach, coachId, teamId)
VALUES
(1, '2020-01-01', '2021-12-31', TRUE, 1, 2),
(2, '2019-07-01', '2021-06-30', TRUE, 2, 3),
(3, '2021-03-15', '2022-12-31', TRUE, 3, 3),
(4, '2020-02-10', '2022-11-01', TRUE, 4, 12),
(5, '2020-04-01', '2023-03-31', TRUE, 5, 13),
```

```
(6, '2021-01-01', '2023-12-31', TRUE, 6, 6),
(7, '2018-05-01', '2021-05-01', TRUE, 7, 7),
(8, '2019-01-01', '2020-12-31', TRUE, 8, 8),
(9, '2020-09-10', '2022-09-10', TRUE, 9, 9),
(10,'2022-01-01', '2023-01-01', TRUE, 11, 4),
(11,'2020-06-01', '2022-06-01', TRUE, 10, 11),
(12,'2019-04-01', '2022-04-01', TRUE, 14, 12),
(13,'2021-05-15', '2023-05-15', TRUE, 13, 5),
(14,'2020-08-01', '2023-08-01', TRUE, 12, 14),
(15,'2021-07-01', '2023-07-01', TRUE, 15, 15);

/*player Table*/

INSERT INTO Player
(playerId, firstName, lastName, dateOfBirth, shirtNumber, role, eMail, teamId, countryId)
VALUES
(1,  'Liam',    'De Smet',   '1995-03-12', '7',  'Batsman',      'liam.desmet@players.com',      1, 1),
(2,  'Noah',    'Janssen',   '1993-07-25', '10', 'All-rounder',  'noah.janssen@players.com',     2, 1),
(3,  'Daan',    'Bakker',    '1997-01-05', '18', 'Bowler',       'daan.bakker@players.com',      2, 2),
(4,  'Sam',     'Vermeer',   '1994-11-20', '22', 'Wicketkeeper', 'sam.vermeer@players.com',      2, 2),
(5,  'Lucas',   'Martin',    '1992-02-15', '3',  'Batsman',      'lucas.martin@players.com',     3, 3),
(6,  'Hugo',    'Dupont',    '1996-09-09', '5',  'Bowler',       'hugo.dupont@players.com',      3, 3),
(7,  'Jonas',   'Keller',    '1991-06-21', '8',  'All-rounder',  'jonas.keller@players.com',     5, 6),
(8,  'Finn',    'Müller',    '1998-12-30', '23', 'Bowler',       'finn.mueller@players.com',     4, 4),
(9,  'Carlos',  'Lopez',     '1995-04-18', '9',  'Batsman',      'carlos.lopez@players.com',     5, 5),
(10, 'Miguel',  'Garcia',    '1993-10-03', '11', 'All-rounder',  'miguel.garcia@players.com',    5, 5),
(11, 'Joao',    'Silva',     '1999-01-27', '4',  'Bowler',       'joao.silva@players.com',       1, 6),
(12, 'Marco',   'Bianchi',   '1994-08-14', '12', 'Wicketkeeper', 'marco.bianchi@players.com',    3, 8),
(13, 'Harry',   'Evans',     '1992-05-09', '14', 'Batsman',      'harry.evans@players.com',      3, 8),
(14, 'Ethan',   'Taylor',    '1996-03-03', '19', 'Bowler',       'ethan.taylor@players.com',     4, 9),
(15, 'Ravi',    'Sharma',    '1990-07-30', '21', 'All-rounder',  'ravi.sharma@players.com',      5, 15);

/* tournament table*/

INSERT INTO Tournament (tournamentId, name, year, format, startDate, endDate) VALUES
(14,  'ICC Cricket World Cup',       2019, 'ODI', '2019-05-30', '2019-07-14'),
(6,  'ICC T20 World Cup',           2021, 'T20', '2021-10-17', '2021-11-14'),
(2,  'ICC Test Championship Final', 2021, 'Test', '2021-06-18', '2021-06-23'),
(4,  'Asia Cup',                    2022, 'T20', '2022-08-27', '2022-09-11'),
(5,  'Champions Trophy',            2017, 'ODI', '2017-06-01', '2017-06-18'),
(3,  'IPL Season 14',               2021, 'T20 League', '2021-04-09', '2021-10-15'),
(7,  'Big Bash League',             2022, 'T20 League', '2022-12-13', '2023-02-04'),
(8,  'Pakistan Super League',       2023, 'T20 League', '2023-02-13', '2023-03-19'),
(10,  'CPL Caribbean Premier League',2022, 'T20 League', '2022-08-31', '2022-10-01'),
(13, 'County Championship',         2021, 'Test League', '2021-04-08', '2021-09-29'),
(11, 'The Hundred',                 2022, 'T20', '2022-08-03', '2022-09-03'),
(12, 'Womens T20 World Cup',        2023, 'T20', '2023-02-10', '2023-02-26'),
(9, 'Under-19 World Cup',           2020, 'ODI', '2020-01-17', '2020-02-09'),
(1, 'Afro-Asia Cup',                2007, 'ODI', '2007-06-06', '2007-06-10'),
(15, 'World Test Championship',     2023, 'Test', '2023-06-07', '2023-06-11');

/*todymatch*/

INSERT INTO TodayMatch (TodaymatchId, matchDateTime, resultType, tossDecision, overPerInning,
tournamentId, venueId, teamId)
VALUES
```

```
(1,  '2023-01-10 14:00:00', 'Completed',  'Bat', 20, 2, 2, 2),
(2,  '2023-01-12 15:30:00', 'Completed',  'Bowl', 20, 3, 2, 3),
(3,  '2023-01-14 13:00:00', 'Completed',  'Bat', 50, 1, 1, 1),
(4,  '2023-01-16 16:00:00', 'Scheduled',  'Bowl', 20, 4, 4, 4),
(5,  '2023-01-18 18:00:00', 'Completed',  'Bat', 50, 5, 5, 5),
(6,  '2023-01-20 11:00:00', 'Abandoned',  'Bowl', 20, 6, 6, 6),
(7,  '2023-01-22 13:30:00', 'Completed',  'Bat', 50, 8, 8, 8),
(8,  '2023-01-24 14:00:00', 'Completed',  'Bowl', 20, 8, 8, 8),
(9,  '2023-01-26 17:00:00', 'Completed',  'Bat', 20, 9, 9, 9),
(10, '2023-01-28 19:30:00', 'Scheduled',  'Bowl', 50, 10, 10, 10),
(11, '2023-01-30 14:30:00', 'Completed',  'Bat', 20, 12, 11, 1),
(12, '2023-02-01 15:00:00', 'Completed',  'Bowl', 20, 13, 12, 2),
(13, '2023-02-03 11:30:00', 'Tie',        'Bat', 50, 13, 13, 3),
(14, '2023-02-05 12:00:00', 'Completed',  'Bat', 20, 14, 14, 4),
(15, '2023-02-07 16:30:00', 'Completed',  'Bowl', 20, 15, 15, 5);

/* squad selection*/

INSERT INTO SquadSelection(selectedOn, shirtNumber, isCaptain, isViceCaptain, isWicketKeeper,
tournamentId, playerId, teamId)
VALUES

('2019-04-01', 7,  TRUE,  FALSE, FALSE, 1, 2, 1),
('2019-04-01', 10, FALSE, TRUE,  FALSE, 1, 2, 1),
('2019-04-01', 4,  FALSE, FALSE, TRUE,  1, 11,1),
('2019-04-02', 18, TRUE,  FALSE, FALSE, 1, 3, 2),
('2019-04-02', 22, FALSE, TRUE,  FALSE, 1, 4, 2),
('2019-04-03', 3,  TRUE,  FALSE, FALSE, 1, 5, 3),
('2019-04-03', 5,  FALSE, FALSE, TRUE,  1, 6, 3),
('2021-09-15', 7,  TRUE,  FALSE, FALSE, 2, 1, 1),
('2021-09-15', 10, FALSE, TRUE,  FALSE, 2, 2, 1),
('2021-09-15', 4,  FALSE, FALSE, TRUE,  2, 11,1),
('2021-09-16', 18, TRUE,  FALSE, FALSE, 2, 3, 2),
('2021-09-16', 22, FALSE, TRUE,  FALSE, 2, 4, 2),
('2021-09-17', 3,  TRUE,  FALSE, FALSE, 2, 5, 3),
('2021-09-17', 5,  FALSE, FALSE, TRUE,  2, 6, 3),
('2021-09-17', 14, FALSE, TRUE,  FALSE, 2, 13,3);

/*playingXI*/

INSERT INTO PlayingXI
(battingPossition, bowlingOrder, isSubstitute, TodaymatchId, playerId, teamId)
VALUES
-- Match 1 (Team 1)
(1,  1, FALSE, 2, 3, 1),
(2,  3, FALSE, 1, 2, 1),
(3,  5, FALSE, 1, 11,1),

-- Match 2 (Team 2)
(1,  2, FALSE, 2, 3, 2),
(2,  4, FALSE, 2, 4, 2),
(3,  6, TRUE,  2, 12,2),  -- substitute

-- Match 3 (Team 3)
(1,  1, FALSE, 3, 5, 3),
```

```
(2,  3, FALSE, 3, 6, 3),
(3,  6, TRUE,  3, 13,3),

-- Match 4 (Team 1 again)
(1,  1, FALSE, 4, 1, 1),
(2,  2, FALSE, 4, 2, 1),
(3,  5, TRUE,  4, 11,1),

-- Match 5 (Team 2 again)
(1,  2, FALSE, 5, 3, 2),
(2,  3, FALSE, 5, 4, 2),
(3,  4, TRUE,  5, 12,2);

/*matchofficialassignment*/

INSERT INTO MatchOfficialAssignment
(assignmentId, role, assignedAt, umpirId, TodaymatchId)
VALUES
(1,  'On-field Umpire',     '2023-01-09 10:00:00', 2, 3),
(2,  'On-field Umpire',     '2023-01-09 10:00:00', 2, 1),
(3,  'Third Umpire',        '2023-01-09 10:00:00', 3, 1),

(4,  'On-field Umpire',     '2023-01-11 11:00:00', 4, 2),
(5,  'Match Referee',       '2023-01-11 11:00:00', 5, 2),

(6,  'On-field Umpire',     '2023-01-13 12:00:00', 6, 3),
(7,  'Third Umpire',        '2023-01-13 12:00:00', 7, 3),
(8,  'Match Referee',       '2023-01-13 12:00:00', 8, 3),

(9,  'On-field Umpire',     '2023-01-15 09:30:00', 9, 4),
(10, 'On-field Umpire',     '2023-01-15 09:30:00', 10, 4),

(11, 'Match Referee',       '2023-01-17 14:00:00', 1, 5),
(12, 'Third Umpire',        '2023-01-17 14:00:00', 2, 5),

(13, 'On-field Umpire',     '2023-01-19 15:00:00', 3, 6),
(14, 'Match Referee',       '2023-01-19 15:00:00', 4, 6),
(15, 'Third Umpire',        '2023-01-19 15:00:00', 5, 6);

/*matchOver*/

INSERT INTO MatchOver (matchOverId, overNumber, maiden, runsInOver) VALUES
(4, 1,  FALSE, 6),
(2, 2,  FALSE, 8),
(3, 3,  TRUE,  0),
(1, 4,  FALSE, 10),
(5, 5,  FALSE, 3),
(6, 6,  FALSE, 7),
(7, 7,  TRUE,  0),
(8, 8,  FALSE, 5),
(9, 9,  FALSE, 12),
(10, 10, FALSE, 4),
(11, 11, TRUE,  0),
(12, 12, FALSE, 9),
(13, 13, FALSE, 6),
(14, 14, FALSE, 2),
```

```
(15, 15, TRUE,  0);

/*innings*/

INSERT INTO Inning
(inningNumber, run, wicket, overs, targetRun, declared, TodaymatchId, matchOverId)
VALUES
-- Match 1 (T20, TodaymatchId = 1)
(1, 160, 6, 20.0, 0,    FALSE, 2, 2),
(2, 158, 8, 20.0, 161,  FALSE, 1, 2),

-- Match 2 (T20, TodaymatchId = 2)
(1, 180, 5, 20.0, 0,    FALSE, 3, 2),
(2, 181, 7, 19.4, 181,  FALSE, 2, 4),

-- Match 3 (ODI, TodaymatchId = 3, 50 overs)
(1, 250, 9, 50.0, 0,    FALSE, 3, 5),
(2, 200,10, 40.3, 251,  FALSE, 3, 6),

-- Match 4 (T20, TodaymatchId = 4)
(1, 145, 7, 20.0, 0,    FALSE, 4, 7),
(2, 120, 9, 20.0, 146,  FALSE, 4, 8),

-- Match 5 (ODI, TodaymatchId = 5)
(1, 280, 6, 50.0, 0,    FALSE, 5, 6),
(2, 281, 4, 48.2, 281, FALSE, 5, 10),

-- Match 6 (T20, TodaymatchId = 6) – one declared innings
(1, 350, 4, 20.0, 0,    TRUE,  6, 11),
(2, 150, 5, 18.0, 351,  FALSE, 6, 12),

-- Match 7 (ODI, TodaymatchId = 7)
(1, 230, 5, 50.0, 0,    FALSE, 7, 13),
(2, 200, 8, 45.1, 231,  FALSE, 7, 14),

-- Match 8 (T20, TodaymatchId = 8) – declared is NULL
(1, 150,10, 18.5, 0,    NULL,  8, 15);
```

4. Assignment 4 - WHERE + scalar functions

Guidelines:

You have to make **5 different** Queries. Try to make the queries **as relevant as possible**. There are no specific guidelines regarding the output of the queries such as column names, order, … . Please apply this adequately.

In each query one of the following topics should be used:
- Operators (=, <, >, <>, …)
- In
- [not] Between
- [not] like (with % or _ and escape)
- Is [not] null
- And, or, not

In each query one of the following topics should be used:
- String functions
- Numeric functions
- Date and time functions
- Ifnull or coalesce
- Distinct
- Order by

So, in each query you write, you have to use a topic from the first list and a topic from the second list.

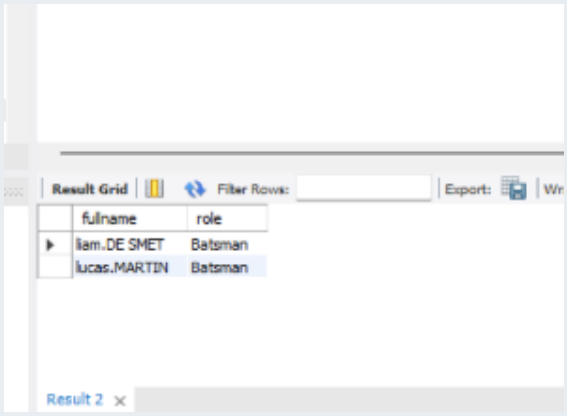
If you use a topic in a query, you will get a score for this topic. If you use it again in an other query, no score will be retrieved anymore for this topic.
This means a topic can be used several times, but is only rewarded once.
So make sure each query you write, contains different topics as the previous one.

4.1. Query 1

Question: Retrieve all players whose first name starts with the letter "L", and display their full name in the format *lowercase firstname + '.' + uppercase lastname*, along with their playing role.

Select statement:

```
select concat(lower(firstName),'.',upper(lastName)) as fullname ,
role
from sport. Player
where firstName like 'L%';
```



Query example:

For each player living in Geel, show the initials, last name and bondnumber. The initials must appear in lower case, the last name in capitals. Initials and last name are shown in one column, separated by a dot. Furthermore, it should not make any difference whether Geel is listed in the database in lower case or in upper case, or in combination of both.

```
SELECT  concat(lower(initials), '. ', upper(name)) as name, bondNumber
FROM    tennis.player
WHERE  lower(city) = 'geel'
```

	name	bondNumber
▶	r. ENGELEN	2411
	r. PEETERS	8467
	g. WIJERS	NULL
	d. BELLENS	NULL
	m. GORP, VAN	6409
	p. HENDERICKX	1608
	p. PEETERS	6524

Content check:

☐ **Operators (=, <, >, <>, …)**	☐ **String functions**
☐ In	☐ Numeric functions
☐ [not] Between	☐ Date and time functions
☐ [not] like (with % or _ and escape)	☐ Ifnull or coalesce
☐ Is [not] null	☐ Distinct
☐ And, or, not	☐ Order by

(So if you use operators or string functions in one of the following queries, you won't get rewarded.)

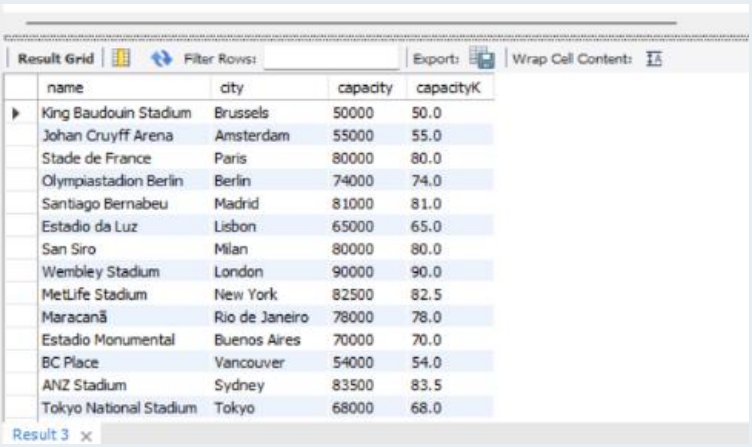
4.2. Query 2

Question: List all venues whose capacity is not equal to 5,000, 6,000, or 7,000, and display their name, city, capacity, and the capacity converted into thousands (rounded to one decimal place).

Select statement:

```
select name, city, capacity,
round(capacity/1000,1) as capacityK
from sport.Venue
where capacity not in(5000,6000,7000);
```

Screenshot of the result:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	city	capacity	capacityK
King Baudouin Stadium	Brussels	50000	50.0
Johan Cruyff Arena	Amsterdam	55000	55.0
Stade de France	Paris	80000	80.0
Olympiastadion Berlin	Berlin	74000	74.0
Santiago Bernabeu	Madrid	81000	81.0
Estadio da Luz	Lisbon	65000	65.0
San Siro	Milan	80000	80.0
Wembley Stadium	London	90000	90.0
MetLife Stadium	New York	82500	82.5
Maracanã	Rio de Janeiro	78000	78.0
Estadio Monumental	Buenos Aires	70000	70.0
BC Place	Vancouver	54000	54.0
ANZ Stadium	Sydney	83500	83.5
Tokyo National Stadium	Tokyo	68000	68.0

Result 3

Content check:

☐ Operators (=, <, >, <>, …)	☐ String functions
☐ In	☐ Numeric functions
☐ [not] Between	☐ Date and time functions
☐ [not] like (with % or _ and escape)	☐ Ifnull or coalesce
☐ Is [not] null	☐ Distinct
☐ And, or, not	☐ Order by

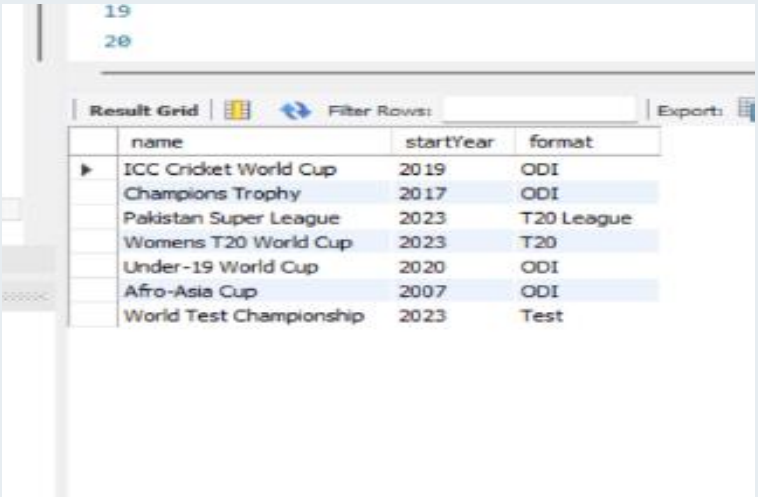
4.3. Query 3

Question: Retrieve all tournaments whose start date does not fall between January 1, 2021 and December 31, 2022, and display their name, format, and the year in which each tournament started.

Select statement:

```
SELECT
    name,
    YEAR(startDate) AS startYear,
    format
FROM sport.Tournament
WHERE startDate NOT BETWEEN '2021-01-01' AND '2022-12-31';
```

Screenshot of the result:



name	startYear	format
ICC Cricket World Cup	2019	ODI
Champions Trophy	2017	ODI
Pakistan Super League	2023	T20 League
Womens T20 World Cup	2023	T20
Under-19 World Cup	2020	ODI
Afro-Asia Cup	2007	ODI
World Test Championship	2023	Test

Content check:

☐ Operators (=, <, >, <>, …)	☐ String functions
☐ In	☐ Numeric functions
☐ [not] Between	☐ Date and time functions
☐ [not] like (with % or _ and escape)	☐ Ifnull or coalesce
☐ Is [not] null	☐ Distinct
☐ And, or, not	☐ Order by

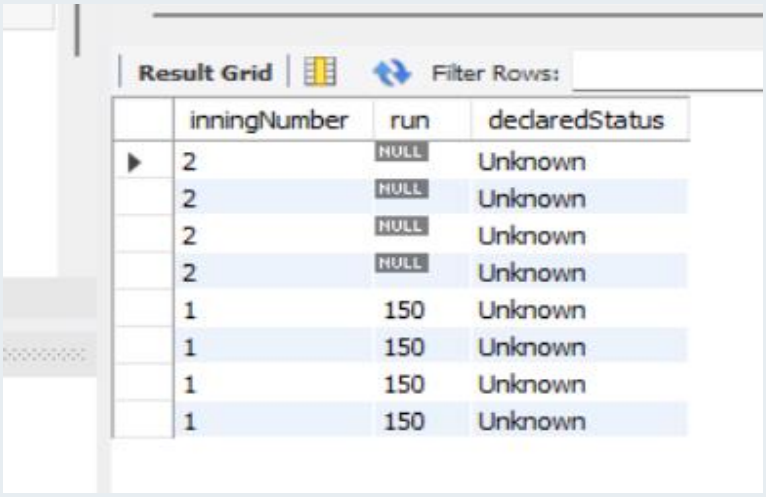
4.4. Query 4

Question: Show all innings where the declared status is unknown (NULL), and display the inning number, runs scored, and a text label indicating the declared status (use "Unknown" when the value is NULL).

Select statement:

```
SELECT
   inningNumber,
   run,
   COALESCE(declared, 'Unknown') AS declaredStatus
FROM sport.Inning
WHERE declared IS NULL;
```

Screenshot of the result:



inningNumber	run	declaredStatus
2	NULL	Unknown
2	NULL	Unknown
2	NULL	Unknown
2	NULL	Unknown
1	150	Unknown
1	150	Unknown
1	150	Unknown
1	150	Unknown

Content check:

☐ Operators (=, <, >, <>, …)	☐ String functions
☐ In	☐ Numeric functions
☐ [not] Between	☐ Date and time functions
☐ [not] like (with % or _ and escape)	☐ Ifnull or coalesce
☐ Is [not] null	☐ Distinct
☐ And, or, not	☐ Order by

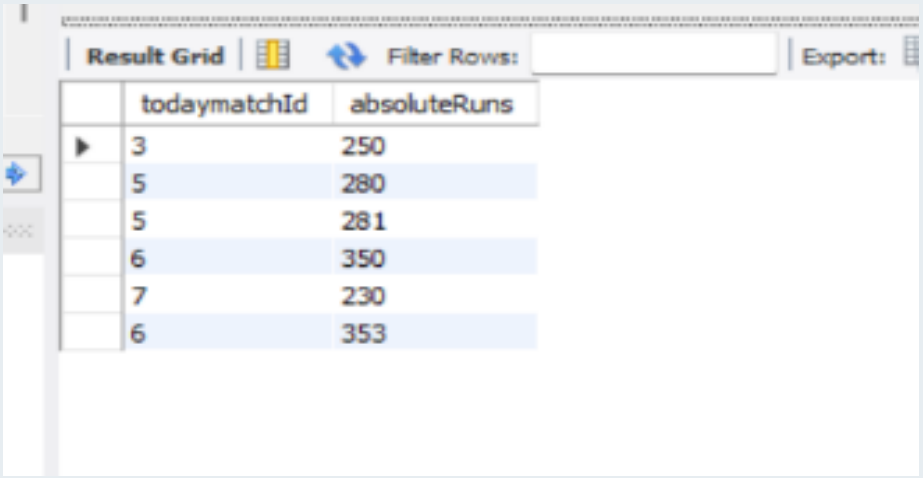
4.5. Query 5

Question:
Retrieve all distinct match IDs from innings where the runs scored were greater than 200, and display the runs using the absolute value function.

Select statement:

```
SELECT DISTINCT
    todaymatchId,
    ABS(run) AS absoluteRuns
FROM sport.Inning
WHERE run > 200;
```

Screenshot of the result



Content check:

☐ Operators (=, <, >, <>, …)	☐ String functions
☐ In	☐ Numeric functions
☐ [not] Between	☐ Date and time functions
☐ [not] like (with % or _ and escape)	☐ Ifnull or coalesce
☐ Is [not] null	☐ Distinct
☐ And, or, not	☐ Order by

5. Assignment 5 - JOINS

Guidelines:

You have to make **5 different** Queries. Try to make the queries **as relevant as possible**. There are no specific guidelines regarding the output of the queries such as column names, order, … . Please apply this adequately.

- An inner join between two tables.
- An inner join between more than two tables.
- A left outer join.
- A right outer join.
- You can choose the fifth join.

5.1. Inner join between two tables

Question: **Retrieve all players along with the name of the team they belong to by performing an inner join between the Player and Team tables.**

Select statement:

```
SELECT
    p.playerId,
    p.firstName,
    p.lastName,
    t.name AS teamName
FROM sport.Player p
INNER JOIN team t ON p.teamId = t.teamId;
```

Screenshot of the result

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

playerId	firstName	lastName	teamName
1	Liam	De Smet	Belgium National Team
2	Noah	Janssen	Belgium National Team
11	Joao	Silva	Belgium National Team
3	Daan	Bakker	Netherlands National Team
4	Sam	Vermeer	Netherlands National Team
12	Marco	Bianchi	Netherlands National Team
5	Lucas	Martin	France National Team
6	Hugo	Dupont	France National Team
13	Harry	Evans	France National Team
7	Jonas	Keller	Germany National Team
8	Finn	Müller	Germany National Team
14	Ethan	Taylor	Germany National Team

Result 3

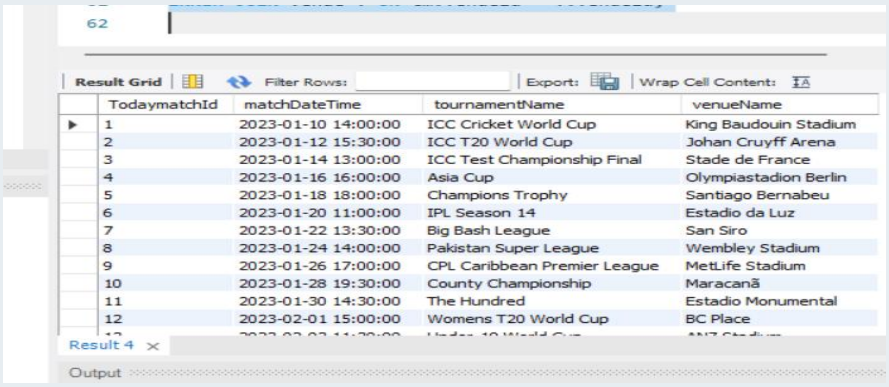
5.2. Inner join between more than two tables

Question: Display each match along with its date and time, the tournament it belongs to, and the venue where it is played by performing inner joins between the TodayMatch, Tournament, and Venue tables.

Select statement:

```
SELECT
    tm.TodaymatchId,
    tm.matchDateTime,
    t.name AS tournamentName,
    v.name AS venueName
FROM sport.TodayMatch tm
INNER JOIN Tournament t ON tm.tournamentId = t.tournamentId
INNER JOIN Venue v ON tm.venueId = v.venueId;
```

Screenshot of the result



TodaymatchId	matchDateTime	tournamentName	venueName
1	2023-01-10 14:00:00	ICC Cricket World Cup	King Baudouin Stadium
2	2023-01-12 15:30:00	ICC T20 World Cup	Johan Cruyff Arena
3	2023-01-14 13:00:00	ICC Test Championship Final	Stade de France
4	2023-01-16 16:00:00	Asia Cup	Olympiastadion Berlin
5	2023-01-18 18:00:00	Champions Trophy	Santiago Bernabeu
6	2023-01-20 11:00:00	IPL Season 14	Estadio da Luz
7	2023-01-22 13:30:00	Big Bash League	San Siro
8	2023-01-24 14:00:00	Pakistan Super League	Wembley Stadium
9	2023-01-26 17:00:00	CPL Caribbean Premier League	MetLife Stadium
10	2023-01-28 19:30:00	County Championship	Maracanã
11	2023-01-30 14:30:00	The Hundred	Estadio Monumental
12	2023-02-01 15:00:00	Womens T20 World Cup	BC Place

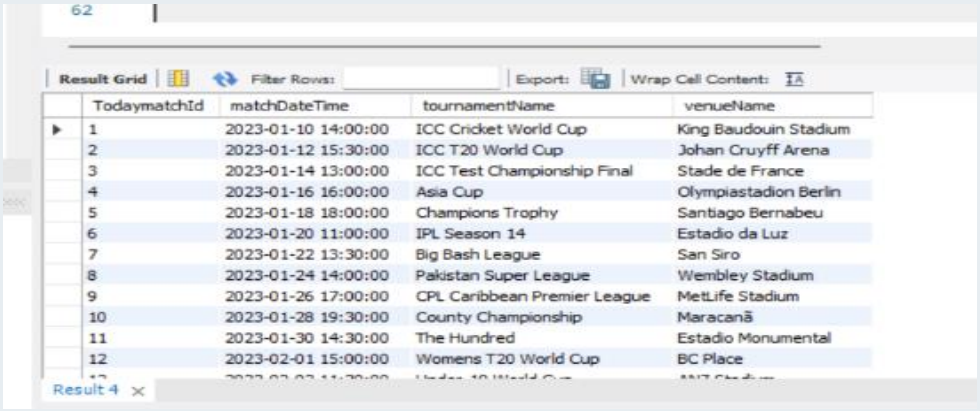
5.3. A left outer join

Question: List all teams and show their corresponding coach assignment details, if available. Use a left join so that teams without a coaching assignment still appear in the results.

Select statement:

```
SELECT
    t.teamId,
    t.name AS teamName,
    a.assignmentId,
    a.startDate
FROM Team t
LEFT JOIN TeamCoachAssignment a
      ON t.teamId = a.teamId;
```

Screenshot of the result



TodaymatchId	matchDateTime	tournamentName	venueName
1	2023-01-10 14:00:00	ICC Cricket World Cup	King Baudouin Stadium
2	2023-01-12 15:30:00	ICC T20 World Cup	Johan Cruyff Arena
3	2023-01-14 13:00:00	ICC Test Championship Final	Stade de France
4	2023-01-16 16:00:00	Asia Cup	Olympiastadion Berlin
5	2023-01-18 18:00:00	Champions Trophy	Santiago Bernabeu
6	2023-01-20 11:00:00	IPL Season 14	Estadio da Luz
7	2023-01-22 13:30:00	Big Bash League	San Siro
8	2023-01-24 14:00:00	Pakistan Super League	Wembley Stadium
9	2023-01-26 17:00:00	CPL Caribbean Premier League	MetLife Stadium
10	2023-01-28 19:30:00	County Championship	Maracanã
11	2023-01-30 14:30:00	The Hundred	Estadio Monumental
12	2023-02-01 15:00:00	Womens T20 World Cup	BC Place

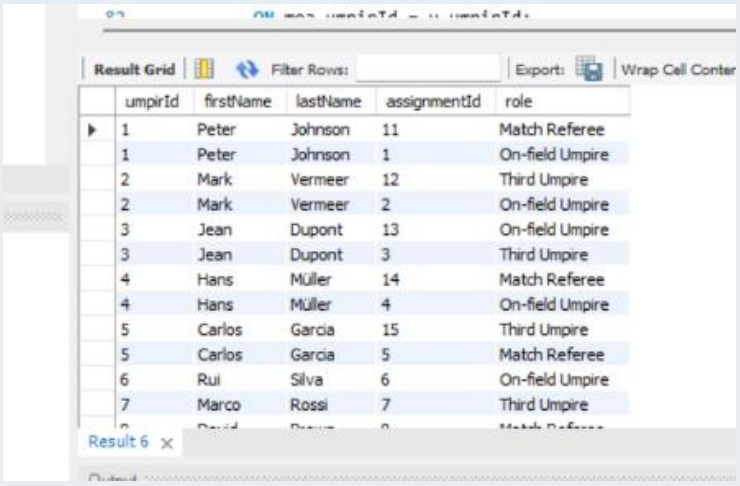
5.4. A right outer join

Question: **Retrieve a list of all umpires along with any match official assignments they have. Use a right join to ensure that every umpire appears in the results, even if they have no assignment.**

Select statement:

```
SELECT
    u.umpirId,
    u.firstName,
    u.lastName,
    moa.assignmentId,
    moa.role
FROM MatchOfficialAssignment moa
RIGHT JOIN Umpire u
      ON moa.umpirId = u.umpirId;
```

Screenshot of the result



umpirId	firstName	lastName	assignmentId	role
1	Peter	Johnson	11	Match Referee
1	Peter	Johnson	1	On-field Umpire
2	Mark	Vermeer	12	Third Umpire
2	Mark	Vermeer	2	On-field Umpire
3	Jean	Dupont	13	On-field Umpire
3	Jean	Dupont	3	Third Umpire
4	Hans	Müller	14	Match Referee
4	Hans	Müller	4	On-field Umpire
5	Carlos	Garcia	15	Third Umpire
5	Carlos	Garcia	5	Match Referee
6	Rui	Silva	6	On-field Umpire
7	Marco	Rossi	7	Third Umpire

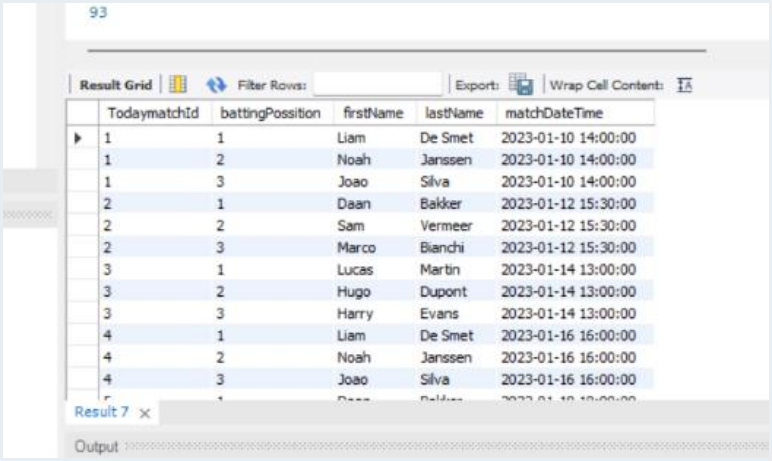
5.5. You can choose the fifth join.

Question: Retrieve the playing XI details for each match, including the batting position, player name, and the date and time of the match, by joining the PlayingXI table with the Player and TodayMatch tables.

Select statement:

```
SELECT
    px.TodaymatchId,
    px.battingPossition,
    p.firstName,
    p.lastName,
    tm.matchDateTime
FROM PlayingXI px
INNER JOIN Player p ON px.playerId = p.playerId
INNER JOIN TodayMatch tm ON px.TodaymatchId = tm.TodaymatchId;
```

Screenshot of the result



TodaymatchId	battingPossition	firstName	lastName	matchDateTime
1	1	Liam	De Smet	2023-01-10 14:00:00
1	2	Noah	Janssen	2023-01-10 14:00:00
1	3	Joao	Silva	2023-01-10 14:00:00
2	1	Daan	Bakker	2023-01-12 15:30:00
2	2	Sam	Vermeer	2023-01-12 15:30:00
2	3	Marco	Bianchi	2023-01-12 15:30:00
3	1	Lucas	Martin	2023-01-14 13:00:00
3	2	Hugo	Dupont	2023-01-14 13:00:00
3	3	Harry	Evans	2023-01-14 13:00:00
4	1	Liam	De Smet	2023-01-16 16:00:00
4	2	Noah	Janssen	2023-01-16 16:00:00
4	3	Joao	Silva	2023-01-16 16:00:00

6. Assignment 6 - Subqueries

You have to make **5 different** Queries. Try to make the queries **as relevant as possible**. There are no specific guidelines regarding the output of the queries such as column names, order, … . Please apply this adequately.

- Single-row subquery
- Multiple-row subquery
- Double key subquery
- Nested subquery
- Subquery in the select

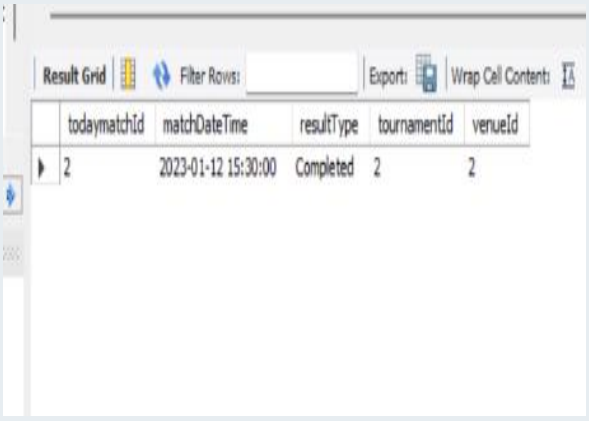
6.1. Single-row subquery

Question: List all matches that were held in the stadium identified by venueId = 2.

Select statement:

```
SELECT tm.todaymatchId,
       tm.matchDateTime,
       tm.resultType,
       tm.tournamentId,
       tm.venueId
FROM sport.TodayMatch AS tm
WHERE tm.venueId = 2;
```

Screenshot of the result



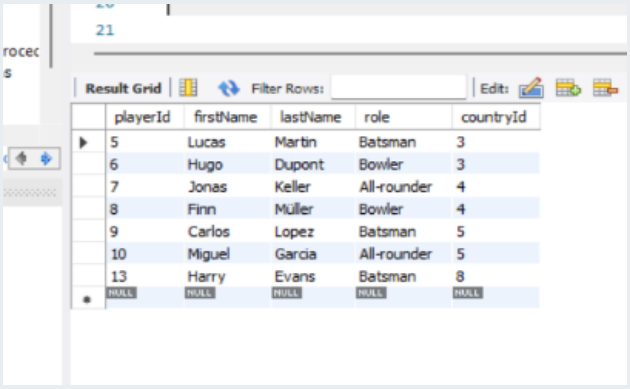
6.2. Multiple-row subquery

Question: Get all players whose country has a team ranked in the top 5 in ICC ranking.

Select statement:

```
SELECT p.playerId,
       p.firstName,
       p.lastName,
       p.role,
       p.countryId
FROM Player AS p
WHERE p.countryId IN (
    SELECT DISTINCT t.countryId
    FROM Team AS t
    WHERE t.iccRanking <= 5
);
```

Screenshot of the result:



21

Result Grid | Filter Rows: | Edit:

playerId	firstName	lastName	role	countryId
5	Lucas	Martin	Batsman	3
6	Hugo	Dupont	Bowler	3
7	Jonas	Keller	All-rounder	4
8	Finn	Müller	Bowler	4
9	Carlos	Lopez	Batsman	5
10	Miguel	Garcia	All-rounder	5
13	Harry	Evans	Batsman	8
NULL	NULL	NULL	NULL	NULL

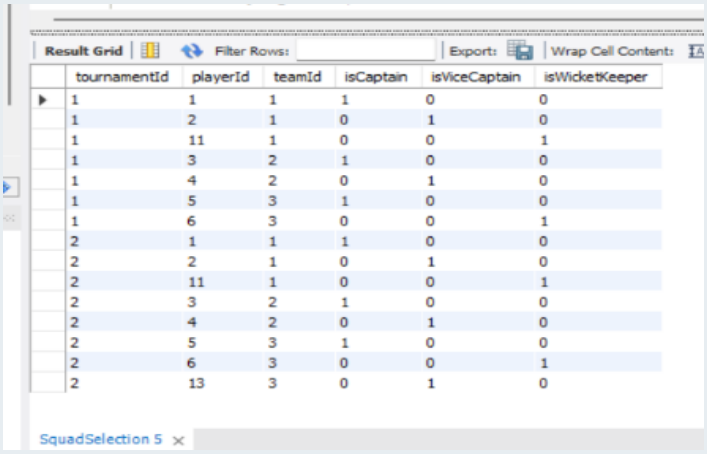
6.3. Double key subquery

Question: Find all squad selections where the player actually appeared in a Playing XI (at least once) for the same team.

Select statement:

```
SELECT s.tournamentId,
       s.playerId,
       s.teamId,
       s.isCaptain,
       s.isViceCaptain,
       s.isWicketKeeper
FROM SquadSelection AS s
WHERE (s.playerId, s.teamId) IN (
   SELECT px.playerId, px.teamId
   FROM PlayingXI AS px
);
```

Screenshot of the result:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

tournamentId	playerId	teamId	isCaptain	isViceCaptain	isWicketKeeper
1	1	1	1	0	0
1	2	1	0	1	0
1	11	1	0	0	1
1	3	2	1	0	0
1	4	2	0	1	0
1	5	3	1	0	0
1	6	3	0	0	1
2	1	1	1	0	0
2	2	1	0	1	0
2	11	1	0	0	1
2	3	2	1	0	0
2	4	2	0	1	0
2	5	3	1	0	0
2	6	3	0	0	1
2	13	3	0	1	0

SquadSelection 5

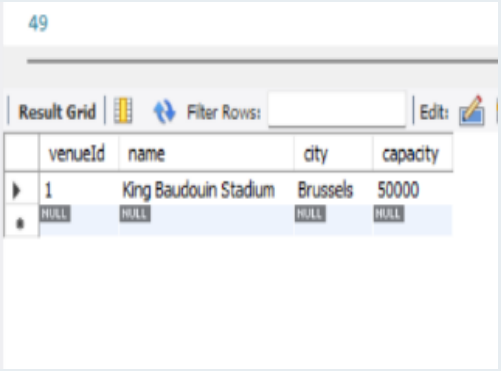
6.4. Nested subquery

Question:
Find the venue(s) where the very first match in the database was played, and show the venue details.

Select statement

```
SELECT v.venueId,
       v.name,
       v.city,
       v.capacity
FROM sport.Venue AS v
WHERE v.venueId IN (
    SELECT DISTINCT tm.venueId
    FROM sport.TodayMatch AS tm
    WHERE tm.matchDateTime = (
        SELECT MIN(matchDateTime)
        FROM sport.TodayMatch
    )
);
```

Screenshot of the result



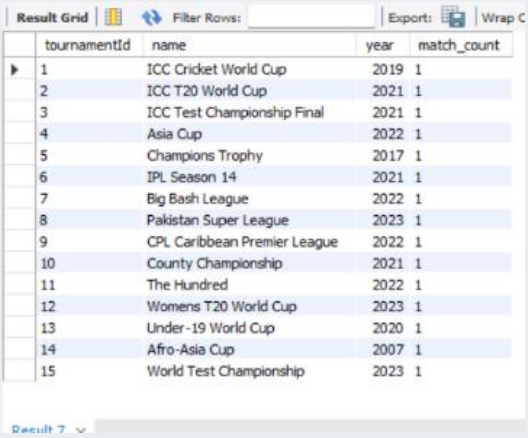
6.5. Subquery in the select

Question:
List all tournaments and show how many matches belong to each tournament.

Select statement

```
SELECT t.tournamentId,
       t.name,
       t.year,
       (
          SELECT COUNT(*)
          FROM TodayMatch AS tm
          WHERE tm.tournamentId = t.tournamentId
       ) AS match_count
FROM Tournament AS t;
```

Screenshot of the result


Result Grid | Filter Rows: | Export: | Wrap C

tournamentId	name	year	match_count
1	ICC Cricket World Cup	2019	1
2	ICC T20 World Cup	2021	1
3	ICC Test Championship Final	2021	1
4	Asia Cup	2022	1
5	Champions Trophy	2017	1
6	IPL Season 14	2021	1
7	Big Bash League	2022	1
8	Pakistan Super League	2023	1
9	CPL Caribbean Premier League	2022	1
10	County Championship	2021	1
11	The Hundred	2022	1
12	Womens T20 World Cup	2023	1
13	Under-19 World Cup	2020	1
14	Afro-Asia Cup	2007	1
15	World Test Championship	2023	1

Result 7

7. Assignment 7 - Set functions

You have to make **2 different** Queries. Try to make the queries **as relevant as possible**. There are no specific guidelines regarding the output of the queries such as column names, order, … . Please apply this adequately.

- Count()
- MIN() or MAX() or AVG() or SUM()

7.1. Count()

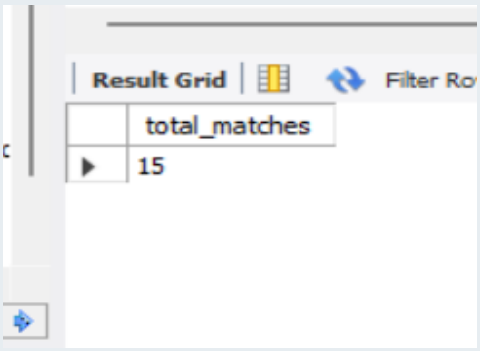
Question:

Count how many matches are in the TodayMatch table.

Select statement:

```
SELECT COUNT(*) AS total_matches
FROM sport.TodayMatch;
```

Screenshot of the result



7.2. MIN() or MAX() or AVG() or SUM()

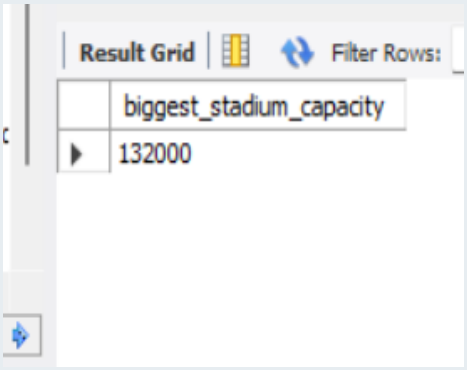
Question:

Find the highest stadium capacity.

Select statement

```
SELECT MAX(capacity) AS biggest_stadium_capacity
FROM sport.Venue;
```

Screenshot of the result



8. Assignment 8 - Correlated subqueries

You have to make **3 different** Queries. Try to make the queries **as relevant as possible**. There are no specific guidelines regarding the output of the queries such as column names, order, … . Please apply this adequately.

You are not allowed to use

- Limit
- Where [not] exists

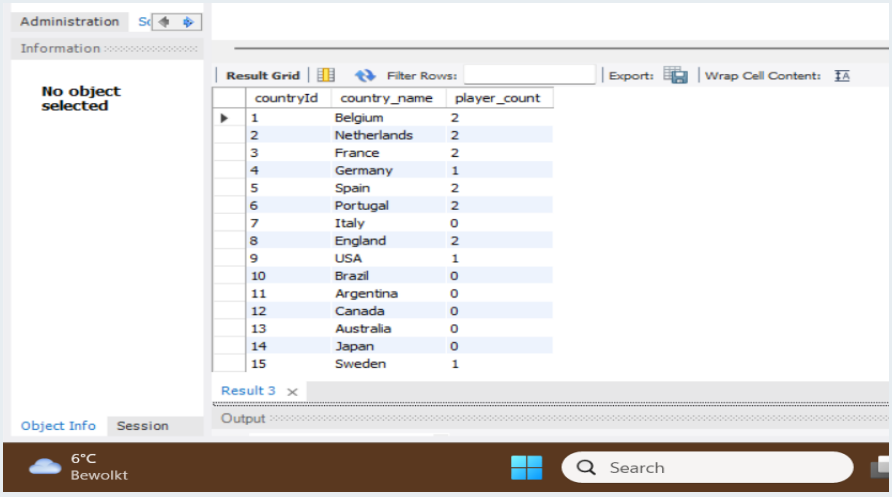
8.1. Correlated subquery 1

Question:
For every country, count how many players come from that country.

Select statement

```
SELECT c.countryId,
       c.name AS country_name,
       (
          SELECT COUNT(*)
          FROM sport.Player AS p
          WHERE p.countryId = c.countryId
       ) AS player_count
FROM sport.Country AS c;
```

Screenshot of the result



8.2. Correlated subquery 2

Question:

For every team, count how many entries they have in TodayMatch

Select statement:

```
SELECT t.teamId,
       t.name AS team_name,
       (
           SELECT COUNT(*)
           FROM sport.TodayMatch AS tm
           WHERE tm.teamId = t.teamId
       ) AS match_count
FROM sport.Team AS t;
```

Screenshot of the result

19

Result Grid | Filter Rows: | Export: | Wrap

teamId	team_name	match_count
1	Belgium National Team	2
2	Netherlands National Team	2
3	France National Team	2
4	Germany National Team	2
5	Spain National Team	2
6	Portugal National Team	1
7	Italy National Team	0
8	England National Team	2
9	USA National Team	1
10	Brazil National Team	1
11	Argentina National Team	0
12	Canada National Team	0
13	Australia National Team	0
14	Japan National Team	0
15	India National Team	0

Result 4 | Result 5 ×

Output

8.3. Correlated subquery 3

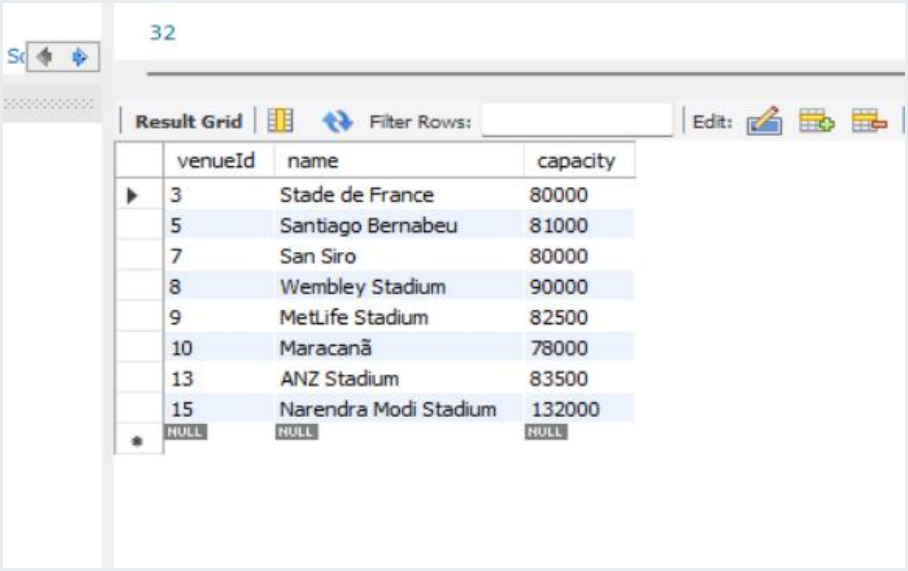
Question:

Venues with capacity above the overall average capacity

Select statement

```
SELECT v.venueId,
       v.name,
       v.capacity
FROM sport.Venue AS v
WHERE v.capacity >
      (
         SELECT AVG(v2.capacity)
         FROM sport.Venue AS v2
      );
```

Screenshot of the result



venueId	name	capacity
3	Stade de France	80000
5	Santiago Bernabeu	81000
7	San Siro	80000
8	Wembley Stadium	90000
9	MetLife Stadium	82500
10	Maracanã	78000
13	ANZ Stadium	83500
15	Narendra Modi Stadium	132000
NULL	NULL	NULL

9. Assignment 9 – Group by

You have to make **5 different** Queries. Try to make the queries **as relevant as possible**. There are no specific guidelines regarding the output of the queries such as column names, order, … . Please apply this adequately.

- Group by with one or more set functions
- Group by on multiple columns
- Group by with an expression
- Group by with having
- Group by with having

9.1. Group by with one ore more set functions

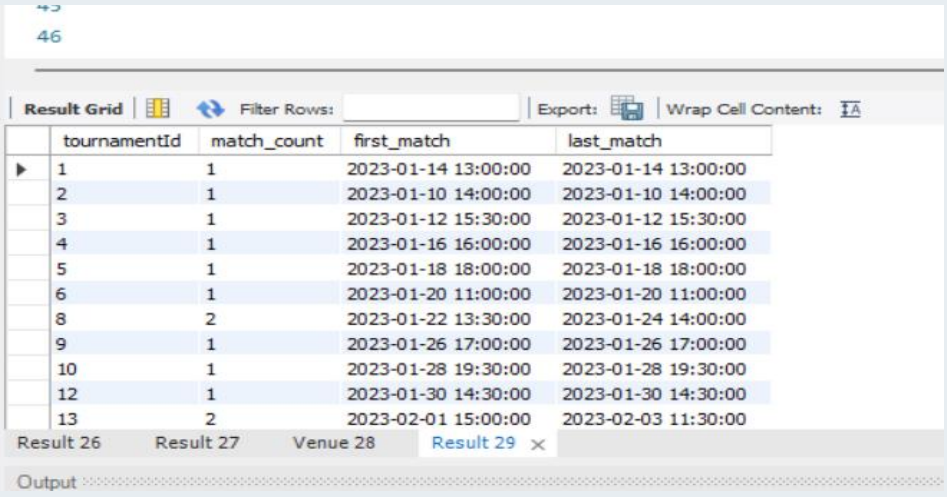
Question

How many matches does each tournament have, and when was the first and last match

Select statement

```
SELECT tm.tournamentId,
       COUNT(*)            AS match_count,
       MIN(tm.matchDateTime) AS first_match,
       MAX(tm.matchDateTime) AS last_match
FROM sport.TodayMatch AS tm
GROUP BY tm.tournamentId;
```

Screenshot of the result



9.2. Group by on multiple columns

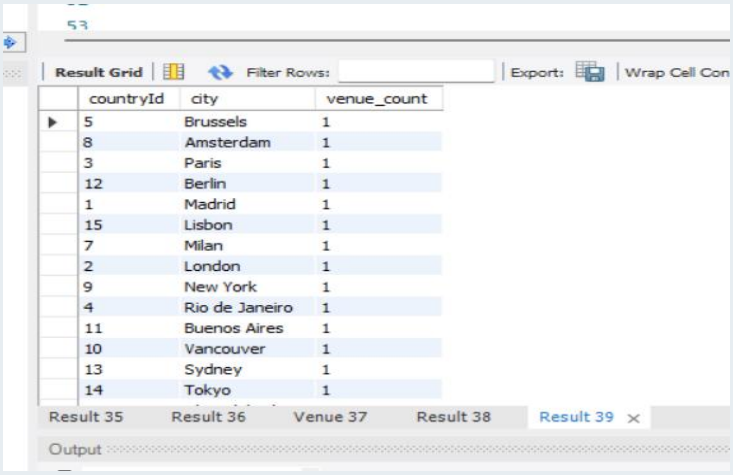
Question

How many venues are there in the country and city as combination

Select statement

```
SELECT v.countryId,
       v.city,
       COUNT(*) AS venue_count
FROM sport.Venue AS v
GROUP BY v.countryId, v.city;
```

Screenshot of the result



9.3. Group by with an expression

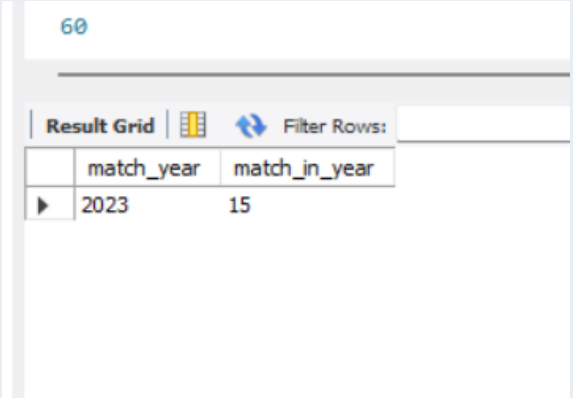
Question

How many matches are played per year based on matchDateTime

Select statement

```
SELECT YEAR(tm.matchDateTime) AS match_year,
       COUNT(*)               AS matches_in_year
FROM sport.TodayMatch AS tm
GROUP BY YEAR(tm.matchDateTime);
```

Screenshot of the result



9.4. Group by with having

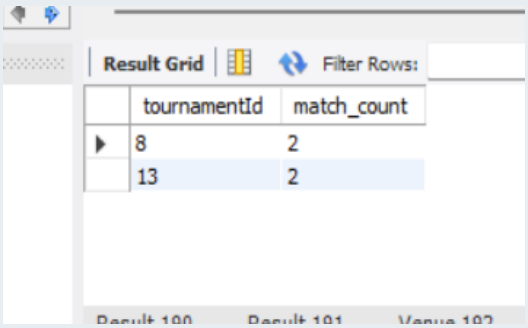
Question

Show tournaments that have at least 2 matches.

Select statement

```
SELECT tm.tournamentId,
       COUNT(*) AS match_count
FROM sport.TodayMatch AS tm
GROUP BY tm.tournamentId
HAVING COUNT(*) >= 2;
```

Screenshot of the result



9.5. Group by with having

Question
Teams with at least 2 players selected in SquadSelection

Select statement

```
SELECT s.teamId,
       COUNT(s.playerId) AS squad_members
FROM sport.SquadSelection AS s
GROUP BY s.teamId
HAVING COUNT(s.playerId) >= 2;
```

Screenshot of the result