

COMP 421 - Project 2

Andrew Cook - 260781415

Lewis Kelly - 261123012

Relational Translations

Stadiums(stadiumName,capacity,location,directions,vipPrice,food)

Matches(identifier,priceMultiplier,length,dateTime,round,stadiumName)

- stadiumName foreign key referencing relation to Stadiums

Team(country,name,URL,group,stadiumName,identifier)

- stadiumName foreign key referencing Matches relation to Stadiums
- identifier foreign key referencing relation to Matches

Players(number,dateOfBirth,name,position,stadiumName,identifier,country)

- stadiumName foreign key referencing Matches relation to Stadiums
- identifier foreign key referencing relation to Matches
- country foreign key referencing relation to Team

PlayersInGame(number,numberOfYellowCards,redCard,minuteEntered,minuteLeft,detailedPosition,stadiumName,identifier,country)

- number key referencing ISA relation to Players
- stadiumName foreign key referencing Matches relation to stadiums
- identifier foreign key referencing relation to Matches
- country foreign key referencing relation of Players to Team

GoalsScored(occurrence,minute,penaltyKick,stadiumName,identifier, number)

- stadiumName foreign key referencing Matches relation to Stadiums
- identifier foreign key referencing relation to Matches
- Number foreign key referencing relation to PlayersInGame

Coaches(name,dateOfBirth,role,stadiumName,identifier,country)

- stadiumName foreign key referencing Matches relation to Stadiums
- identifier foreign key referencing Team relation to Matches
- country foreign key referencing relation to Team

Referees(refID,name,yearsOfExperience,country)

RefereesInGame(refID,name,role,stadiumName,identifier)

- refID key referencing ISA relation to Referees

- stadiumName foreign key referencing Matches relation to Stadiums
- identifier foreign key referencing relation to Matches

Seat(seatNumber,seatSection,basePrice,stadiumName,identifier)

- stadiumName foreign key referencing relation to Stadium
- identifier foreign key referencing relation to Matches

Tickets(verificationNumber,finalPrice,stadiumName, seatNumber, identifier)

- stadiumName foreign key referencing Seat relation to Stadium
- seatNumber,identifier foreign key referencing relation to Seat
- identifier foreign key referencing relation to Matches

```
User(password,name,paymentInfo,email, verificationNumber)
```

- verificationNumber foreign key referencing Seat relation to Tickets

Pending Constraints

- In the same match no two players can wear the same number regardless of team
- The same match identifier can only be applied to players from no more than two different teams
- Player shirt numbers must be between 1 and 99
- Players must be 18+ to compete
- Players can get at most two yellow cards per match
- Only one main ref is allowed per match all the rest must be assistants
- User may purchase multiple tickets however each ticket must come with a separate verification number

Some of these constraints could be enforced by changing the relation schema but as per assignment instructions only minimal changes were made from TA feedback and to ensure createtbl.sql was able to execute successfully.

SQL Queries

5.a)

[illegible]

;

```
cs421g138@winter2023-comp421:~/project2/p2$ db2 -t -f question5a.sql -v
SELECT s.stadiumName, s.location, m.DATETIME FROM Stadiums s, Matches m WHERE s.STADIUMNAME=m.STADIUMNAME AND m.identifier in (SELECT identifier FROM GoalsScored WHERE number in (SELECT number FROM PlayersInGame WHERE number in (SELECT number FROM Players WHERE name = 'Christine Sinclair'))))

STADIUMNAME                                LOCATION                                DATETIME
-----
MetLife Stadium                            7333 Sachs Crossing                    2023-03-05-00.00.00.000000
FNB Stadium                                0 Rowland Court                       2022-04-27-00.00.00.000000

2 record(s) selected.
```

5.b)

SELECT NAME, NUMBER, COUNTRY
FROM PLAYERS

WHERE NUMBER in (SELECT NUMBER

FROM (SELECT NUMBER, COUNTRY, COUNT(COUNTRY) as cone

FROM PLAYERSINGAME

GROUP BY NUMBER, COUNTRY) one

INNER JOIN (SELECT p.COUNTRY, COUNT(DISTINCT IDENTIFIER) ctwo

FROM PLAYERSINGAME p

GROUP BY p.COUNTRY) two

ON one.COUNTRY = two.COUNTRY AND one.cone = two.ctwo)

;

```
cs421g138@winter2023-comp421:~/project2/p2$ db2 -t -f question5b.sql -v
SELECT NAME, NUMBER, COUNTRY FROM PLAYERS WHERE NUMBER in (SELECT NUMBER FROM (SELECT NUMBER, COUNTRY, COUNT(COUNTRY) as cone FROM PLAYERSINGAME GROUP BY NUMBER, COUNTRY) one INNER JOIN (SELECT p.COUNTRY, COUNT(DISTINCT IDENTIFIER) ctwo FROM PLAYERSINGAME p GROUP BY p.COUNTRY) two ON one.COUNTRY = two.COUNTRY AND one.cone = two.ctwo)

NAME                                NUMBER    COUNTRY
-----
Baird Charke                        33        Ukraine
Alphonso Stoodale                   51        China
Christine Sinclair                  65        Canada

3 record(s) selected.
```

5.c)

SELECT two.COUNTRY, numMatches, numGoals

FROM (SELECT COUNTRY, COUNT(NUMBER) numGoals

FROM (SELECT COUNTRY, p.NUMBER

FROM PLAYERSINGAME p

INNER JOIN(SELECT NUMBER, COUNT(OCCURANCE)

FROM GOALSSCORED

WHERE PENALTYKICK = false

GROUP BY NUMBER) one

ON p.NUMBER = one.NUMBER)

GROUP BY COUNTRY) two

;

COUNTRY	NUMMATCHES	NUMGOALS
Canada	3	9
China	2	3
England	3	2
Ukraine	2	1

```
SELECT five.country, five.numTickets / six.numGames AvgTicketsPerGame
FROM (SELECT DISTINCT two.COUNTRY, SUM(three.four) numTickets
      FROM (SELECT DISTINCT t.IDENTIFIER, p.COUNTRY, COUNT(DISTINCT t.IDENTIFIER) one
            FROM TICKETS t,
                 PLAYERSINGAME p
            WHERE t.IDENTIFIER = p.IDENTIFIER
            GROUP BY t.IDENTIFIER, p.COUNTRY) two
            INNER JOIN (SELECT IDENTIFIER, COUNT(IDENTIFIER) four
                       FROM TICKETS
                       GROUP BY IDENTIFIER) three
            ON two.IDENTIFIER = three.IDENTIFIER
      GROUP BY two.COUNTRY) five
      INNER JOIN (SELECT COUNTRY, COUNT(DISTINCT IDENTIFIER) numGames
                  FROM PLAYERSINGAME
                  GROUP BY COUNTRY) six
      ON five.COUNTRY = six.COUNTRY
```

;

COUNTRY	AVGTICKETSPERGAME
Canada	9
China	7
England	9
Ukraine	14

4 record(s) selected.

5.e)

Number of cards (red plus yellow) given by each ref throughout the tournament assuming only the main ref can give cards.

```
SELECT DISTINCT r.NAME, SUM(numCards)
FROM REFEREESINGAME r
    INNER JOIN
    (SELECT one.IDENTIFIER, yellow + red AS numCards
    FROM (SELECT IDENTIFIER, COUNT(NUMBEROFYELLOWCARDS) yellow
    GROUP BY IDENTIFIER) one
    INNER JOIN
    (SELECT IDENTIFIER, COUNT(REDCARD) red
    FROM PLAYERSINGAME
    WHERE REDCARD = true
    GROUP BY IDENTIFIER) two
    ON one.IDENTIFIER = two.IDENTIFIER) three
    ON three.IDENTIFIER = r.IDENTIFIER
WHERE r.ROLE = 'main'
GROUP BY r.NAME
;
```

```
cs421g138@winter2023-comp421:~/project2/p2$ db2 -t -f question5e.sql -v
SELECT DISTINCT r.NAME, SUM(numCards) FROM REFEREESINGAME r INNER JOIN (SELECT one.IDENTIFIER, yellow + red AS numCards FROM (SELECT IDENTIFIER, COUNT(N
UMBEROFYELLOWCARDS) yellow FROM PLAYERSINGAME GROUP BY IDENTIFIER) one INNER JOIN (SELECT IDENTIFIER, COUNT(REDCARD) red FROM PLAYERSINGAME WHERE REDCAR
D = true GROUP BY IDENTIFIER) two ON one.IDENTIFIER = two.IDENTIFIER) three ON three.IDENTIFIER = r.IDENTIFIER WHERE r.ROLE = 'main' GROUP BY r.NAME

NAME                                2
-----
Cullie Corrin                        7
Kora Byford                          7
Palmer Rubury                        5
Rana Chapling                        4

  4 record(s) selected.
```

Player Information

6.a)

```
CREATE OR REPLACE VIEW playerinfo
    (NAME,NUMBER,DATEOFBIRTH,COUNTRY,ASSOCIATION,GROUP)
AS
SELECT p.NAME,p.NUMBER,p.DATEOFBIRTH,p.COUNTRY,t.NAME,t.GROUP
FROM PLAYERS p,TEAM t
WHERE p.COUNTRY=t.COUNTRY
```

;

6.b)

```
cs421g138@winter2023-comp421:~/project2/p2$ db2 -t -f question6b.sql -v
CREATE OR REPLACE VIEW playerinfo (NAME,NUMBER,DATEOFBIRTH,COUNTRY,ASSOCIATION,GROUP) AS SELECT p.NAME,p.NUMBER,p.DATEOFBIRTH,p.COUNTRY,t.NAME,t.GROUP FROM PLAYERS p,TEAM t WHERE p.COUNTRY=t.COUNTRY
DB20000I The SQL command completed successfully.
```

6.c)

```
cs421g138@winter2023-comp421:~/project2/p2$ db2 -t -f question6c.sql -v
SELECT * FROM playerinfo LIMIT 5

NAME                NUMBER    DATEOFBIRTH  COUNTRY          ASSOCIATION                GROUP
-----
Baird Charke        33 08/21/2001 Ukraine         Ukraine Football Club      B
Towney Schaffel     23 09/14/1995 Comoros         Comoros Football Club      D
Pierson Noland      8 01/28/1990 Canada          Canada Football Club       B
Alphonso Stoodale   51 04/03/2002 China           China Football Club        E
Christine Sinclair  65 07/11/2003 Canada          Canada Football Club       B

5 record(s) selected.
```

6.d)

```
cs421g138@winter2023-comp421:~/project2/p2$ db2 -t -f question6d.sql -v
SELECT * FROM playerinfo WHERE GROUP='A' LIMIT 5

NAME                NUMBER    DATEOFBIRTH  COUNTRY          ASSOCIATION                GROUP
-----
Lind Raynor         38 10/03/1994 Cuba           Cuba Football Club         A
Gothart Ygo         86 01/19/1997 Jordan         Jordan Football Club       A
Herculie Sines      96 09/09/1994 Mexico         Mexico Football Club       A

3 record(s) selected.
```

6.e)

```
cs421g138@winter2023-comp421:~/project2/p2$ db2 -t -f question6e.sql -v
INSERT INTO playerinfo(name, number, dateofbirth, country, association, group) VALUES ('Tom Smith',26,'1990-03-21','USA','USA Football Club','D')
DB21034E The command was processed as an SQL statement because it was not a
valid Command Line Processor command. During SQL processing it returned:
SQL0150N The target fullselect, view, typed table, materialized query table,
range-clustered table, or staging table in the INSERT, DELETE, UPDATE, MERGE,
or TRUNCATE statement is a target for which the requested operation is not
permitted. SQLSTATE=42807
```

It's not permitted because the view is a result of a join from two separate tables and since a view is just a virtual table that reflects the real tables. So this insert fails as it is trying to insert columns from two different tables into just one which of course is impossible.

Check Constraints

The constraint makes sure that all players shirt numbers are between 1 and 99.

```
db2 => cs421g138@winter2023-comp421:~/project2/p2$ db2 -t -f question7a.sql -v
ALTER TABLE PLAYERS ADD CONSTRAINT check_shirtnumber CHECK ( NUMBER BETWEEN 1 AND 99)
DB20000I  The SQL command completed successfully.
```

```
cs421g138@winter2023-comp421:~/project2/p2$ db2 -t -f question7b.sql -v
INSERT INTO PLAYERS(number, dateofbirth, name, position, stadiumname, identifier, country) VALUES (121,'2000-01-01','John','forward','Rose Bowl',5949,'Canada')
DB21034E  The command was processed as an SQL statement because it was not a
valid Command Line Processor command. During SQL processing it returned:
SQL0545N  The requested operation is not allowed because a row does not
satisfy the check constraint "CS421G138.PLAYERS.CHECK_SHIRTNUMBER".
SQLSTATE=23513
```