

## Part 2: Revised Relational Schema

### Relation 1

Teams (country, association, URL, group)

**Association** NOT NULL

**URL** NOT NULL

**group** NOT NULL

Players (country, shirt\_num, pname, position, DOB)

**Country** foreign key referencing **Teams**

**pname** NOT NULL

**DOB** NOT NULL

**position** NOT NULL

Coaches (country, cname, DOB, role)

**Country** foreign key referencing **Teams**

**DOB** NOT NULL

**role** NOT NULL

Referees (rname, year\_of\_experience, country)

**year\_of\_experience** NOT NULL

**country** NOT NULL

Stadiums(sname, location, capacity)

**location** NOT NULL

**capacity** NOT NULL

Games(id, date, time, length, round, stadium, playing\_country, opposing\_country)

**date** NOT NULL

**time** NOT NULL

**length** NOT NULL

**round** NOT NULL

**stadium** NOT NULL foreign key referencing **stadiums**(name)

**playing\_country** NOT NULL foreign key referencing **Teams**(country)

**opposing\_country** NOT NULL foreign key referencing **Teams**(country)

Goals(id, occurrence, minute, penalty, shirt\_num, country)

**Id** foreign key referencing **Games**

**minute** NOT NULL

**penalty** NOT NULL

**shirt\_num, country** NOT NULL foreign key referencing **Players**(shirt\_num, country)

Refs\_game (rname,id, role)

**rname** foreign key referencing **Referees**

**id** foreign key referencing **Games**

**role** NOT NULL

Performs(pname, id, etime, ltime, position, Rcard, Ycard)

**pname** foreign key referencing **Players**

**id** foreign key referencing **Games**

**etime** NOT NULL

**ltime** NOT NULL

**position** NOT NULL

**Rcard** NOT NULL

**Ycard** NOT NULL

## Relation 2

Customers(email,password)

**password** NOT NULL

Billing\_Addresses(aid,address, address2, city, province, country, postal\_code, phone, name)

**address** NOT NULL

**address2** NOT NULL

**City** NOT NULL

**Province** NOT NULL

**Country** NOT NULL

**postal\_code** NOT NULL

**phone** NOT NULL

**name** NOT NULL

**Email** NOT NULL

Stadium\_Addresses(aid, address,city, province, country)

**Address** NOT NULL

**City** NOT NULL

**Province** NOT NULL

**Country** NOT NULL

Games(gname, date, time, aid)

**date** NOT NULL

**time** NOT NULL

**aid** NOT NULL referencing **Stadium\_Addresses**

Credit\_cards(c\_num, c\_type, cvv, expiration\_dm, aid, cardholder\_name)

**C\_type** NOT NULL

**CVV** NOT NULL

**expiration\_dm** NOT NULL

**aid** NOT NULL foreign key referencing **Billing\_Addresses**

**Cardholder\_name** Not NULL

Tickets(tid, section, row, seat, gname, listed\_price)

**section** NOT NULL

**row** NOT NULL

**seat** NOT NULL

**gname** foreign key referencing **Games**

**listed\_price** NOT NULL

Sold\_tickets(tid, sold\_price)

**sold\_price** NOT NULL

Purchases(tid, c\_num, email, transaction\_id, transaction\_date, transaction\_time)

**tid** foreign key referencing **Sold\_tickets**

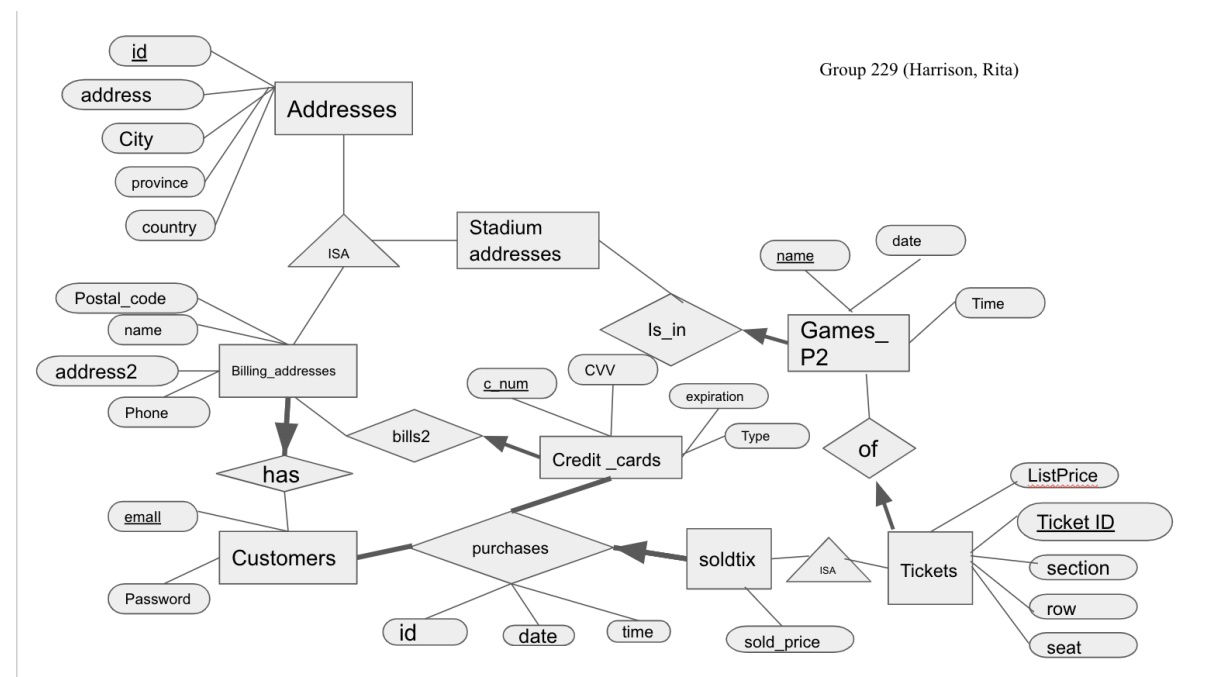
**c\_num** foreign key referencing **Credit\_cards**

**email** foreign key referencing **Customers**

**transaction\_id** NOT NULL

**transaction\_date** NOT NULL

**transaction\_time** NOT NULL



### Changes made to ER schema

- Renamed the "Games" relation in part 2 to "Games\_P2" to avoid having two tables with the same name.
- Adding "address2" attribute to "Billing\_addresses" entity set (for apartment number / PO box).
- Added both participation and key constraint to "Billing\_addresses" in relation to the relationship set "has."
- Added "postal\_code" attribute to "Billing\_addresses" entity set.
- Added participation constraint to "Credit\_cards" entity set on "Purchases" relationship set.
- Added participation constraint to "Customers" entity set on "Purchases" relationship set.
- Added credit card type (c\_type) attribute to the Credit\_cards entity set.

## Part 3: Pending Constraints

### PART 1

- No numbers in the schema (e.g. capacity, round, shirt number) should be negative in real life, but the database allows negative integers.
- The “Group” attribute in the Teams table must be a capital letter from A-H, but in the database it could be any character.
- The “Penalty” attribute in Goals should be “TRUE” (goal scored during penalty kick) or “FALSE” (goal scored outside penalty kick) but the database allows it to be any sequence of characters.
- In-game players are dismissed after one red card, but the database can record any integer of red cards per player.
- In real life, each team must have enough of each type of player to form a valid team (e.g. teams with only 2 players, or teams with no goalie cannot play), although this is not shown in the database.
- A game must only have one main referee in real life.

### PART 2

- No Numbers (eg. sold\_price, listed\_price, section, row, seat) should be negative.
- Ticket purchase date has to be before or on the game day. Customers cannot purchase tickets after the game’s already been played.
- Credit card expiration date and month should be valid. Currency is all in CAD, assuming this is a Canadian website.
- In the database implementation, the name on a certain billing address is allowed to be different from the cardholder name on the credit card associated with the billing address. In reality whether different names are allowed or not depends on the banking system.
- In one transaction, a customer is allowed to purchase tickets from multiple different games in the current database implementation, in reality each transaction should be tickets from the same game. Different games should require separate purchases.
- In every address, the city should belong to the state/province, and the state/province should belong to the country; In the case of the billing\_address, the postal code should be valid according to the region as well.
- The credit card numbers must be valid and match the credit card type.
- Passwords should have lowercase and uppercase letters and of a certain length.
- The ISA hierarchy is not object-oriented, so all Sold\_tickets must also be present in Tickets.

## Part 5. SQL Queries

(a) Write a SQL query that lists all the stadium names and their locations and the match date of matches in which player Christine Sinclair (12) has played and scored at least one goal. You can assume that there is only one player with this particular name (and you can change the actual player name if you have a particular favourite).

SELECT sname, location, date FROM Stadiums JOIN Games ON Stadiums.sname = Games.stadium WHERE id in (SELECT id FROM Goals WHERE country = (SELECT country FROM Players WHERE pname = 'Christine Sinclair') AND shirt\_num = (SELECT shirt\_num FROM Players WHERE pname = 'Christine Sinclair'));

```
db2 => SELECT sname, location, date FROM Stadiums JOIN Games ON Stadiums.sname = Games.stadium WHERE id in (SELECT id FROM Goals WHERE country = (SELECT country FROM Players WHERE pname = 'Christine Sinclair') AND shirt_num = (SELECT shirt_num FROM Players WHERE pname = 'Christine Sinclair'));
```

SNAME	LOCATION	DATE
McConnell Arena	Rawaglagah	08/27/2022
Rutherford Park	Erpeldange	04/28/2022

2 record(s) selected.

db2 =>

(b) Write a SQL query that lists the name, shirt number and country of all players that have played in all matches of their teams.

WITH countries\_and\_shirts (country, shirt\_num) AS (SELECT country, shirt\_num FROM (SELECT country, shirt\_num, COUNT(id) AS player\_game\_count FROM Performs GROUP BY country, shirt\_num) AS a JOIN (SELECT both\_teams.c, COUNT(both\_teams.c) AS game\_count FROM (SELECT playing\_country AS c FROM Games UNION ALL SELECT opposing\_country AS c FROM Games) both\_teams GROUP BY both\_teams.c) AS b ON b.c = a.country WHERE a.player\_game\_count=b.game\_count) SELECT Players.pname, Players.shirt\_num, Players.country FROM Players JOIN countries\_and\_shirts ON Players.country = countries\_and\_shirts.country AND Players.shirt\_num = countries\_and\_shirts.shirt\_num;

```
db2 => WITH countries_and_shirts (country, shirt_num) AS (SELECT country, shirt_num FROM (SELECT country, shirt_num, COUNT(id) AS player_game_count FROM Performs GROUP BY country, shirt_num) AS a JOIN (SELECT both_teams.c, COUNT(both_teams.c) AS game_count FROM (SELECT playing_country AS c FROM Games UNION ALL SELECT opposing_country AS c FROM Games) both_teams GROUP BY both_teams.c) AS b ON b.c = a.country WHERE a.player_game_count=b.game_count) SELECT Players.pname, Players.shirt_num, Players.country FROM Players JOIN countries_and_shirts ON Players.country = countries_and_shirts.country AND Players.shirt_num = countries_and_shirts.shirt_num;
```

PNAME	SHIRT_NUM	COUNTRY
Christine Sinclair	12	Canada
Loren Whyffen	953	Croatia
Allison Meadley	779	Ecuador
Rosalynd Spours	37160	France
Pen Philcott	5359	Iran
Baldwin Gianni	5688	Japan
Addy Domingues	7649	Mexico
Berny Simmank	31	Netherlands
Test Test	69555	Poland
Test Test	90	Tunisia

10 record(s) selected.

(c) Write a SQL query that lists for each team, the country, the number of matches they have played and the total number of goals they have scored during normal play (not counting the penalty kicks in case of a tie). Note that for this query, it might be very useful to have extra attributes attached to some

relations (which will depend on your schema) that somehow keep track of how many such goals each team scored in a match. Feel free to add such extra attributes to your schema if you find them useful. But you can also try to extract the information through the detailed information you store for each goal, but that might be quite complex.

```
SELECT Teams.country, COALESCE(teams_that_played.game_count, 0) AS game_count,
COALESCE(teams_that_played.num_kicks, 0) AS num_kicks FROM Teams LEFT OUTER JOIN
(SELECT num_of_matches.tname AS country, num_of_matches.game_count,
num_penalty_kicks.num_kicks FROM (SELECT both_teams.c AS tname, COUNT(both_teams.c) AS
game_count FROM (SELECT playing_country AS c FROM Games UNION ALL SELECT
opposing_country AS c FROM Games) both_teams GROUP BY both_teams.c) num_of_matches
LEFT OUTER JOIN (SELECT Country AS cname, COUNT(Country) AS num_kicks FROM Goals
WHERE penalty = 'FALSE' GROUP BY Country) num_penalty_kicks ON num_of_matches.tname =
num_penalty_kicks.cname) AS teams_that_played ON Teams.country = teams_that_played.country;
```

```
db2 => SELECT Teams.country, COALESCE(teams_that_played.game_count, 0) AS game_count, COALESCE(teams_t
hat_played.num_kicks, 0) AS num_kicks FROM Teams LEFT OUTER JOIN (SELECT num_of_matches.tname AS count
ry, num_of_matches.game_count, num_penalty_kicks.num_kicks FROM (SELECT both_teams.c AS tname, COUNT(b
oth_teams.c) AS game_count FROM (SELECT playing_country AS c FROM Games UNION ALL SELECT opposing_coun
try AS c FROM Games) both_teams GROUP BY both_teams.c) num_of_matches LEFT OUTER JOIN (SELECT Country
AS cname, COUNT(Country) AS num_kicks FROM Goals WHERE penalty = 'FALSE' GROUP BY Country) num_penalty
_kicks ON num_of_matches.tname = num_penalty_kicks.cname) AS teams_that_played ON Teams.country = team
s_that_played.country;
```

COUNTRY	GAME_COUNT	NUM_KICKS
Argentina	0	0
Australia	0	0
Belgium	1	0
Brazil	9	11
Cameroon	0	0
Canada	2	5
Costa Rica	2	0
Croatia	2	2
Denmark	0	0
Ecuador	1	0
England	2	0
France	4	1
Germany	0	0
Ghana	0	0
Iran	3	2
Japan	5	7
Mexico	4	4
Morocco	0	0
Netherlands	1	1
Poland	11	14
Portugal	8	10
Qatar	0	0
Saudi Arabia	0	0
Senegal	0	0
Serbia	0	0
South Korea	1	0
Spain	1	0
Switzerland	0	0
Tunisia	1	1
United States	5	3
Uruguay	0	0
Wales	1	0

32 record(s) selected.

(d) Create an interesting SQL query that extracts some information from tables that refers to purchasing tickets, e.g., some summary information about tickets sold for a particular match, information how many tickets were sold for a match / each match and whether the stadium was sold out, or anything else that might be interesting. The query should not only be a simple query on a single table with only basic selections and projections.

```
SELECT B.country, B.email FROM Billing_addresses B WHERE B.email IN (SELECT P.email
FROM Purchases P WHERE P.transaction_id IN (SELECT transaction_id FROM (SELECT
P.transaction_id, COUNT(DISTINCT P.tid) AS num_tickets FROM Purchases P WHERE
P.transaction_id IN (SELECT DISTINCT P.transaction_id FROM Tickets T,Purchases P WHERE
T.tid = P.tid AND T.gname LIKE '%France%' AND T.tid in (SELECT S.tid FROM Sold_tickets S
WHERE S.sold_price > 150)) GROUP BY P.transaction_id) WHERE num_tickets > 3));
```

Query returns the customer's country of their billing address and customer email for those customers who purchased more than 3 tickets where each ticket is sold for more than \$150 in a single purchase for a game that France played in.

```
db2 => SELECT B.country, B.email FROM Billing_addresses B WHERE B.email IN (SELECT P.email FROM Purchases P WHERE
P.transaction_id IN (SELECT transaction_id FROM (SELECT P.transaction_id, COUNT(DISTINCT P.tid) AS num_tickets F
ROM Purchases P WHERE P.transaction_id IN (SELECT DISTINCT P.transaction_id FROM Tickets T,Purchases P WHERE T.ti
d = P.tid AND T.gname LIKE '%France%' AND T.tid in (SELECT S.tid FROM Sold_tickets S WHERE S.sold_price > 150)) G
ROUP BY P.transaction_id) WHERE num_tickets > 3));
```

COUNTRY	EMAIL
China	drizzello1i@1und1.de
Panama	rkrale6c@i2i.jp
Russia	pedscer9g@sun.com
United States	tbroadleydh@rambler.ru

4 record(s) selected.

(e) Create a further SQL query that is of interest for this soccer world cup database. Maybe it uses some tables that are not used in any of the other queries, or performs some conditions on the date/time attributes of the schema or any other attributes that have not been used in one of the other queries.

```
SELECT country, PlayerAvgAge FROM (SELECT country, AVG(p_age) AS PlayerAvgAge FROM
(SELECT P.country,P.pname, YEAR(CURRENT_DATE)-YEAR(P.DOB) AS p_age FROM Players P
WHERE P.position LIKE '%Midfielder%') GROUP BY country) WHERE PlayerAvgAge < 35 AND
COUNTRY IN (SELECT country FROM Goals) ORDER BY 2;
```

Query returns the average age of midfielders as of today in ascending order of goal-scoring countries who have an average age of midfielders that's below 35.

```
db2 => SELECT country, PlayerAvgAge
FROM (SELECT country, AVG(p_age) AS PlayerAvgAge
      FROM (SELECT P.country, P.pname, YEAR(CURRENT_DATE) - YEAR(P.DOB) AS p
            _age
            FROM Players P
            WHERE P.position LIKE '%Midfielder%')
      GROUP BY country)
WHERE PlayerAvgAge < 35 AND COUNTRY IN (SELECT country
                                       FROM Goals)
ORDER BY 2; db2 (cont.) => db2 (cont.) => db2 (cont.) => db2 (cont.) => db2
(cont.) => db2 (cont.) => db2 (cont.) => db2 (cont.) =>

COUNTRY          PLAYERAVGAGE
-----
Poland            26
Japan             27
Brazil            28
Mexico            29
Iran              32
United States     33

6 record(s) selected.
```

## 6. Player Information

Create a view playerinfo that shows for each player the name, shirt number, DOB, their country, the national association their team belongs to, and the group name of the group the team is part of during group round.

(a)

```
CREATE VIEW playerinfo (pname, shirt_num, dob, country, association, groupname) AS SELECT
P.pname, P.shirt_num, P.DOB, P.country, T.association, T.group FROM Players P, Teams T WHERE
P.country = T.country;
```

(b)

```
db2 => CREATE VIEW playerinfo (pname, shirt_num, dob, country, association, groupname) AS SELECT P.pname, P.shir
t_num, P.DOB, P.country, T.association, T.group FROM Players P, Teams T WHERE P.country = T.country;
DB20000I The SQL command completed successfully.
```

(c)

```
db2 => SELECT * FROM playerinfo
```

PNAME	SHIRT_NUM	DOB	COUNTRY	ASSOCIATION	GROUPNAME
Rosalynd Spours	37160	04/14/1991	France	FFF	D
Friederike Bolan	54286	05/16/1999	Brazil	Brazilian Football Confederation	G
Tobin De Francesco	779	01/02/1994	Brazil	Brazilian Football Confederation	G
Test Test	30	01/19/1999	Portugal	FPF	H
Ingrim Bassick	837	07/01/1981	Germany	Deutscher Fußball-Bund	E

(d)

```
db2 => SELECT * FROM playerinfo WHERE groupname = 'A';
```

PNAME	SHIRT_NUM	DOB	COUNTRY	ASSOCIATION	GROUPNAME
Allison Meadley	779	03/15/1978	Ecuador	Ecuadorian Football Federation	A
Berny Simank	31	07/13/2000	Netherlands	KNVB	A

2 record(s) selected.

(e)



Query: INSERT INTO playerinfo VALUES ('Leo Messi', 30, '1987-06-24', 'Argentina', 'Argentine Football Association', 'C');

```
db2 => INSERT INTO playerinfo VALUES ('Leo Messi', 30, '1987-06-24', 'Argentina',
'Argentine Football Association', 'C');
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
db2 =>
```

This occurs because this query attempted to insert data into the view, instead of the original Players and Teams tables. Since this view draws on data from two different tables, it is read-only, so the insertion was not permitted (error 42807). As per IBM's documentation, inserts into a view are not allowed in the following situations: the view contains a join, GROUP BY, or HAVING clause, the SELECT contains various qualifiers and expressions, multiple columns are derived from the same column, and a base table of the view contains a column that isn't included in the view and does not contain a default value.

## Part 7. Check Constraints

Add ONE CHECK constraint to one of your tables that is important in the context of this database. Maybe you had indicated a constraint during your specification that you would like to ensure and that you can express with the options we discussed in class.

ALTER TABLE Games\_P2 Add CONSTRAINT gdatecheck CHECK( date > '2023-01-01');

```
db2 => ALTER TABLE Games_P2 Add CONSTRAINT gdatecheck CHECK( date > '2023-01-01');
DB20000I The SQL command completed successfully.
db2 => db2 =>
```

This constraint makes sure that the games that are on the ticket sales website should have not happened yet or should be after a certain date. The constraint here sets up a time where the tickets must be after a certain date, which can be changed manually, or set to "CURRENT\_DATE" based on the website's needs. Here we use the date '2023-01-01' as an example.

Inserting a record that violates this constraint:

INSERT INTO Games\_P2 VALUES ('France vs.Portugal 2022-07-20','2022-07-20','4:04 AM',4300111278);

```
db2 => INSERT INTO Games_P2 VALUES ('France vs.Portugal 2022-07-20','2022-07-20','4:04 AM',4300111278);
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 "CS421G229.GAMES_P2.GDATECHECK". SQLSTATE=23513
```