COMP421: Database Systems
Project: Database Design and Data Modelling

Part 2: Database Creation
Due: Feb 27, 12:00 pm

Justin Sun	260707684
Dominic Wener	260871052

Relational Schema

Teams(country, status, goal_differential, web_URL, gname)

- web_URL foreign key referencing NationalAssociation
- web_URL NOT NULL
- gname foreign key referencing Groups
- gname NOT NULL

Grouped(country, gname, points)

- country foreign key referencing Teams
- gname foreign key referencing Groups

Groups(gname)

National Associations(web_URL, aname, country)

- country foreign key referencing Teams
- country NOT NULL

Players(pid, pname, number, position, DOB, country)

- country foreign key referencing Teams
- country NOT NULL

Coaches(cid, cname, DOB, role, country)

- country foreign key referencing Teams
- country NOT NULL

Referees(rid, rname, country, experience)

Stadiums(sname, capacity, location)

Matches(mid, match_length, score, start_time, round, date, sname)

- sname foreign key referencing Stadiums
- sname NOT NULL

Scheduled(country, mid)

- country foreign key referencing Teams

mid foreign key referencing Matches

Refereed(rid, mid, role)

rid foreign key referencing Referees

mid foreign key referencing Matches

Played(pid, mid, y_cards, r_card, specific_position, time_in, time_out)

pid foreign key referencing Players

mid foreign key referencing Matches

Goals(gid, pid, mid, penalty, occurrence_order, scorer_name, time)

pid foreign key referencing Players

pid NOT NULL

mid foreign key referencing Matches

mid NOT NULL

Seats(seatNumber, sname, sectionNumber)

sname foreign key referencing Stadiums

sname NOT NULL

Clients(email, name, password)

Tickets(date, seatNumber, sname, mid, price, purchase_status)

seatNumber foreign key referencing Seat

sname foreign key referencing Stadiums

mid foreign key referencing Matches

Buys(email, date, seatNumber, sname, mid)

email foreign key referencing Client

date foreign key referencing Tickets

seatNumber foreign key referencing Seat

sname foreign key referencing Stadiums

mid foreign key referencing Matches

***Differences between relational schema defined above from P1 and the relations created in the DB2 database for P2:*

1) 'Scheduled' relation expressing a participation constraint between 'Teams' and 'Matches' was not implemented as its own relation table in the database, and was instead implemented as 2 attributes in the 'Matches' entities. They must be NOT NULL to ensure a Match doesn't exist without having 2 teams that will play in it (participation constraint). See below the possible 'Scheduled' relation implementation that was removed:

```
/*CREATE TABLE Scheduled
(
   country NOT NULL,
   mid NOT NULL,
   PRIMARY KEY (country, mid),
   FOREIGN KEY (country) REFERENCES Teams(country),
   FOREIGN KEY (mid) REFERENCES Matches(mid)
);*/
```

2) Score attribute in 'Matches' was implemented as separate t1_score and t2_score, as we now have team1 and team2 attributes as defined above. The previously defined score attribute in the Part1 relational schema would have needed to be a string because of our design, which would not be useful in any types of queries. On the other hand, the new separate score attributes serve the same purpose and can be queried appropriately, i.e. the score attribute didn't store useful data compared to the separate team1 score and team2 score attributes. However, most queries can still use the 'Goals' relation to calculate scores of all matches, since the 'Goals' relation links all goals to their scorer and respective match.
3) The National Association relation is represented within the Teams relation as name and URL attributes, possible by the one-to-one constraint between them (i.e. no relation table created for National Associations).

Pending constraints

- The total number of the following entities will generally be lower bounded and upper bounded by the specific rules of the tournaments: Players, Teams, Referees, Coaches, Matches and teams per group. These constraints could be expressed as checks when creating tables, but aren't implemented in our database as of the submission for the project Part 2.
- Matches in the same round cannot be between the same teams (i.e. tuples (round, team1, team2) must be unique across the different match entities). Similarly, when teams are eliminated, their team name (country) cannot appear in the later round matches.
- Matches require a specific amount of assigned referees, which would translate to Refereed relation entities. Exactly one head referee, and multiple assistants.
- This implementation currently allows two clients to buy the same ticket. This should be controlled by using the 'purchased_status' attribute of tickets when the application to sell tickets is implemented.
- Goal entities have a time attribute that indicates the time a goal is scored during a match. These timestamps must be after (i.e. larger than) the start_time attribute of the match it was scored in. Similarly, goals scored in the same match must have occurence_order values according to their time attribute values. In other words, goals with the smallest time values will also have the smallest order values and so forth (i.e. chronological order).
- Matches will have team 1 and team 2 score attribute values, which should correspond to the amount of goals created in the database along with the correct goal information (i.e. goal scored from player in the correct team, with the correct ordering of goals, etc.). The existence of the goal entities isn't yet enforced to match the actual match data in 'Matches' and would require these constraints to have relevant data in the database. The queries as of now use the goal entities and not the Match score attributes, so results for some queries will not match the t1_score and t2_score.
- All artificial ids created (i.e. to help create the entity primary key) should be non-negative integers, and increment starting from 1. Auto-incrementing ids could be implemented in the future.
- The total number of tickets available for a given match must be upper-bounded by the stadium capacity it takes place in.
- Match scheduling constraints are not enforced yet, that is more than one match cannot occur at the same time in the same stadium and a team cannot play more than one match at the same time.

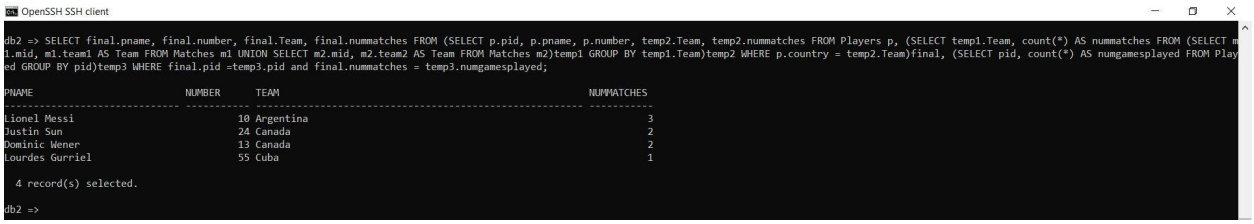
SQL Queries

a) Lists all the stadium names and their locations and the match date of matches in which player Lionel Messi has played and scored at least one goal:

```
SELECT s.sname, s.location, temp.date
FROM Stadiums s, (SELECT m.sname, m.date
                  FROM Matches m
                  WHERE mid IN (SELECT g.mid
                                FROM Goals g
                                WHERE g.scorer_name = 'Lionel Messi')
)temp
WHERE s.sname = temp.sname
;
```

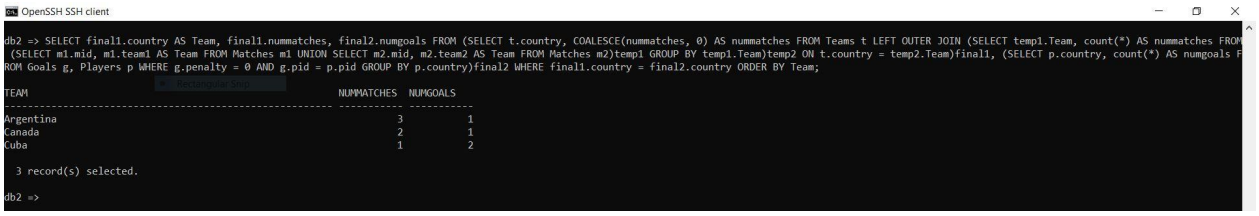
b) Lists the name, shirt number and country of all players that have played in all matches of their teams:

```
SELECT final.pname, final.number, final.Team, final.nummatches
FROM (SELECT p.pid, p.pname, p.number, temp2.Team, temp2.nummatches
      FROM Players p, (SELECT temp1.Team, count(*) AS nummatches
                       FROM (SELECT m1.mid, m1.team1 AS Team
                             FROM Matches m1
                             UNION
                             SELECT m2.mid, m2.team2 AS Team
                             FROM Matches m2
                       )temp1
                       GROUP BY temp1.Team
      )temp2
      WHERE p.country = temp2.Team
)final, (SELECT pid, count(*) AS numgamesplayed
         FROM Played
         GROUP BY pid
)temp3
WHERE final.pid =temp3.pid and final.nummatches = temp3.numgamesplayed
;
```



c) 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

```
SELECT final1.country AS Team, final1.nummatches, final2.numgoals
FROM (SELECT t.country, COALESCE(nummatches, 0) AS nummatches
        FROM Teams t LEFT OUTER JOIN (SELECT temp1.Team, count(*) AS nummatches
                FROM (SELECT m1.mid, m1.team1 AS Team
                        FROM Matches m1
                        UNION
                        SELECT m2.mid, m2.team2 AS Team
                        FROM Matches m2
                        )temp1
                        GROUP BY temp1.Team
                )temp2
                ON t.country = temp2.Team
)final1, (SELECT p.country, count(*) AS numgoals
        FROM Goals g, Players p
        WHERE g.penalty = 0 AND g.pid = p.pid
        GROUP BY p.country
)final2
WHERE final1.country = final2.country
ORDER BY Team
;
```

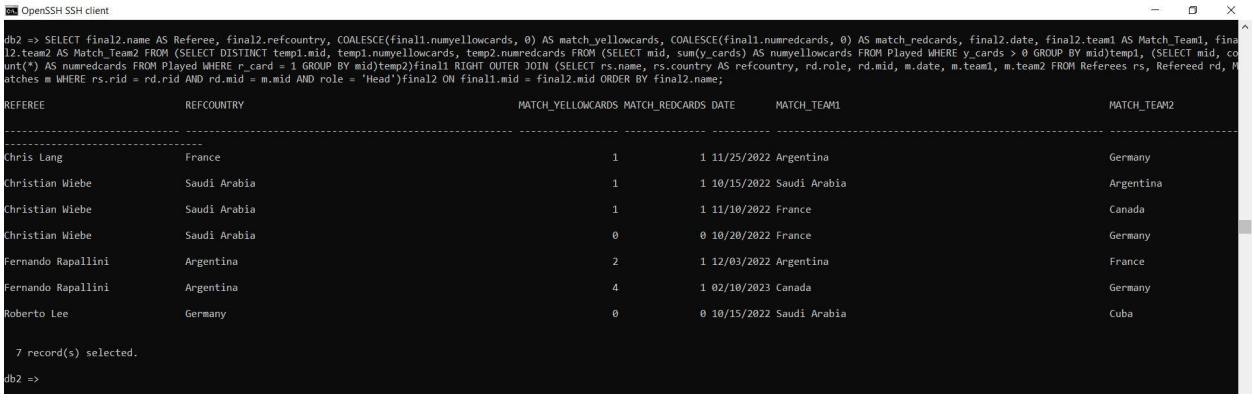


d) *Description Q5.d):* This query returns the email and the total amount of money spent by the clients that have spent the most money when buying tickets across all the games in the tournaments.

```
SELECT temp1.email, temp2.MostSpent
FROM
  (SELECT b.email,SUM(t.price) AS TotalSpent
   FROM Buys b,
       Tickets t
   WHERE b.date = t.date
     AND b.seatNumber = t.seatNumber
     AND b.sname = t.sname
     AND b.mid = t.mid
     AND t.purchase_status = 0
   GROUP BY b.email
  )temp1,
  (SELECT MAX(prices.TotalSpent) AS MostSpent
   FROM (SELECT b.email, SUM(t.price) AS TotalSpent
         FROM Buys b,
             Tickets t
         WHERE b.date = t.date
           AND b.seatNumber = t.seatNumber
           AND b.sname = t.sname
           AND b.mid = t.mid
           AND t.purchase_status = 0
         GROUP BY b.email
        )prices
  )temp2
WHERE temp1.TotalSpent = temp2.MostSpent
;
```

e) *Description Q5.e):* This query returns information about the head referee for every unique game played in the tournament, as well as sanctions given to the players in that match. In other words, for every match in the database, this query returns the head referee name, their country, the 2 teams (country names) that played in the match, the match date and the total amount of yellow cards and red cards given to the players during this match.

```
SELECT final2.name AS Referee, final2.refcountry,
        COALESCE(final1.numyellowcards, 0) AS match_yellowcards,
        COALESCE(final1.numredcards, 0) AS match_redcards, final2.date,
        final2.team1 AS Match_Team1, final2.team2 AS Match_Team2
FROM (SELECT DISTINCT temp1.mid, temp1.numyellowcards, temp2.numredcards
    FROM (SELECT mid, sum(y_cards) AS numyellowcards
        FROM Played
        WHERE y_cards > 0
        GROUP BY mid
    )temp1, (SELECT mid, count(*) AS numredcards
        FROM Played
        WHERE r_card = 1
        GROUP BY mid
    )temp2
)final1 RIGHT OUTER JOIN
        (SELECT rs.name, rs.country AS refcountry, rd.role, rd.mid, m.date, m.team1, m.team2
         FROM Referees rs, Refereed rd, Matches m
         WHERE rs.rid = rd.rid AND rd.mid = m.mid AND role = 'Head'
)final2
ON final1.mid = final2.mid
ORDER BY final2.name
;
```

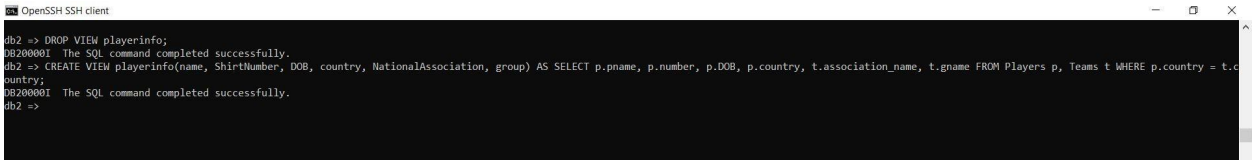


Player Information

a)

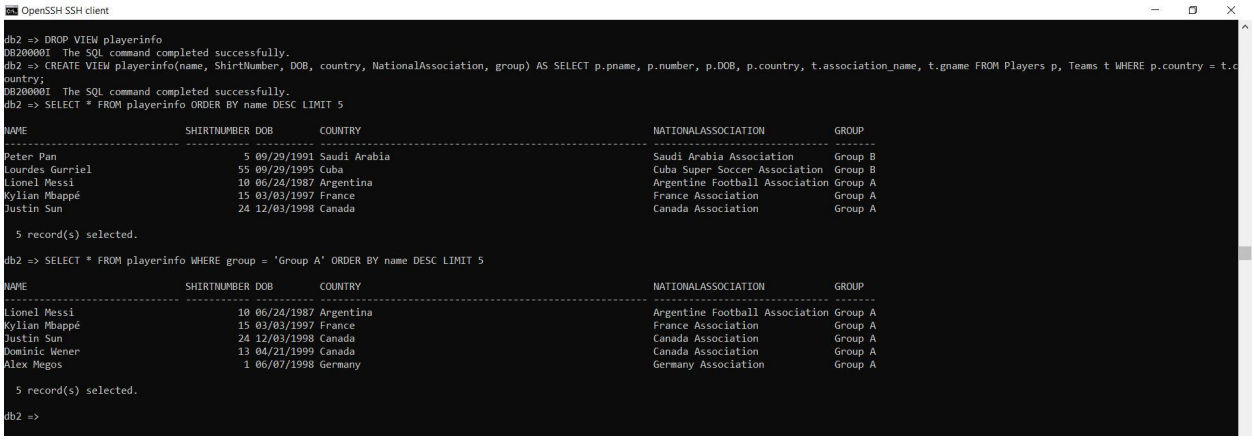
```
CREATE VIEW playerinfo(name, ShirtNumber, DOB, country, NationalAssociation, group)
AS SELECT p.pname, p.number, p.DOB, p.country, t.association_name, t.gname
FROM Players p, Teams t
WHERE p.country = t.country;
```

b)

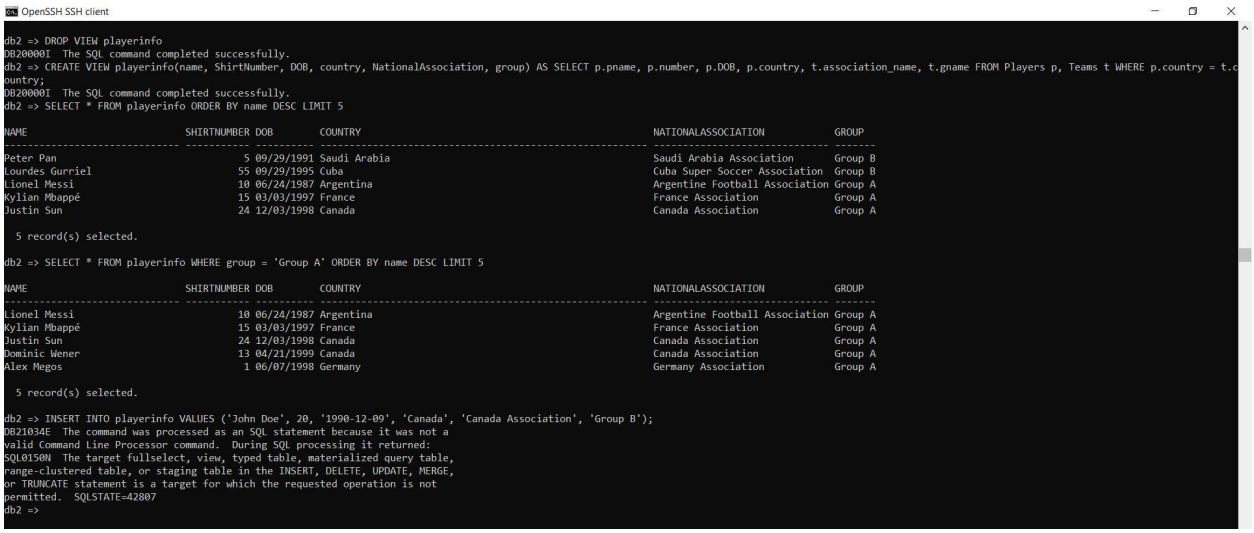


c)

d)

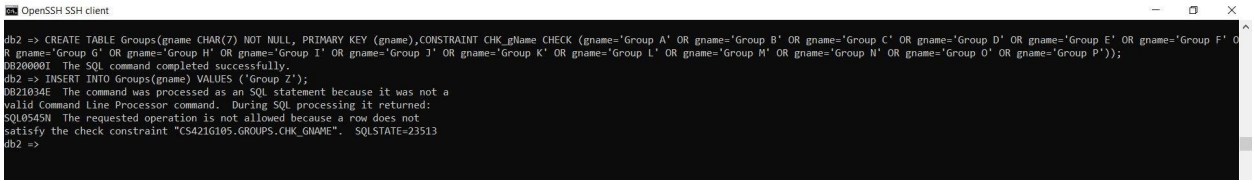


e)



This is because a view is an unmaterialized relation; a definition is stored rather than a set of tuples. We can not insert a tuple into a definition or unmaterialized relation. We also used a join to create our view, so inserting is problematic here since you can't insert some of the values into one table and the rest into another table.

Check Constraints



This CHECK constraint prevents the creation of more than 16 groups by checking that any group inserted into the Groups(gname) table is one of Groups A through P. When all 16 possible names have been inserted, if we try to insert 'Group Z' into our relation, the constraint properly returns the following error: 'The requested operation is not allowed because a row does not satisfy the check constraint "CS421G105.GROUPS.CHK_GNAME"'. Moreover, not only does the constraint limit the number of entities, it also limits the possible naming permutations for the groups and makes sure the names are all uniform and sensible.