Project : Bridge Competition Management System

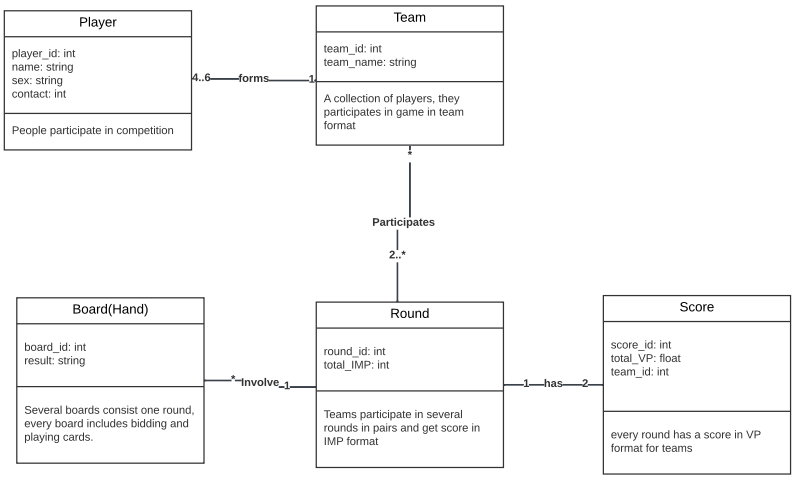
The system aims to develop a monitor that could help anticipants and spectators of a Bridge competition find relative information quickly especially regarding live scores, and detailed performance statistics.

Nouns & Actions

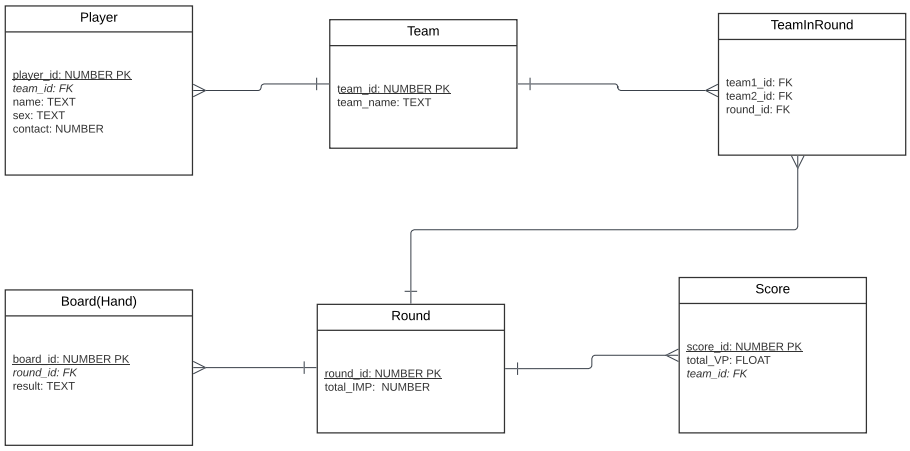
Rules:

1. Players must register for the competition, providing their personal details and contact information.
2. Players form teams often with regular partners (in pairs). A team contains 4 or 6 players so that there are 2 or 3 pairs.
3. The competition consists of multiple rounds which teams all participate in, each involving a set number of boards (hands) to be played.
4. At the beginning of each round, cards are randomly distributed to players according to the rules of the game. In the big tournaments, the machine deals the cards.
5. After every round, system shows real-time scores for teams, enabling participants and spectators to follow the progress of the competition.
6. Teams engage in bidding to determine the contract and play the hands according to the contract's specifications.
7. International Match Point (IMP) scores are calculated based on the results of each round, taking into account the difference in performance between competing teams.
8. Victory Points (VP) are computed to establish rankings among teams in the competition.

UML:



ERD:



Link:
https://lucid.app/lucidchart/42b89d46-2afb-40b9-9d99-0652f82ad0fe/edit?beaconFlowId=92FD39F2F578FC8C&invitationId=inv_77f57017-d4a1-4bc2-b4d9-2e6ad412696c&page=0_0#

Relational Schema:
Player(player_id: PK, team_id: FK, name, sex, contact)
Team(team_id: PK, team_name)
Round(round_id: PK, VP)
Board(board_id: PK, round_id: FK, result)
Score(score_id: PK, team_id; FK, total_VP)

Functional dependencies:
player_id → name, sex, contact
team_id → team_name
(round_id, team_id) → VP
(board_id, round_id) → result
(score_id, team_id) → total_VP

In X → Y, all X are superkey, it is at least BCNF.

Query backup:

```
--5
--Complex Search Criterion with CASE/WHEN:
--Find teams with the highest VP scores and classify them as 'Top Player.
-- SELECT T.team_name,
--       (CASE
--         WHEN S.total_VP = (SELECT MAX(total_VP) FROM Scores) THEN 'Top Player'
--         ELSE ''
--       END) AS Classification
-- FROM Scores AS S
-- JOIN Teams AS T ON S.team_id = T.team_id;

--4
--Group By with Having Clause:
--List teams that have scored a total VP of at least 40 except team 1
-- SELECT T.team_name, SUM(S.total_VP) AS TotalVP
-- FROM Scores AS S
-- JOIN Teams AS T ON S.team_id = T.team_id
-- WHERE S.team_id > 1
-- GROUP BY T.team_name
-- HAVING SUM(S.total_VP) >= 40;

--3
--Subquery
```

```
--Find the total number of players in the system who have participated in more than one
session
-- SELECT COUNT(*)
-- FROM (
--     SELECT team_id
--     FROM Scores
--     GROUP BY team_id, score_id
--     HAVING COUNT(DISTINCT score_id) < 1
-- ) AS Subquery;

--2
-- Join of Three Tables:

-- SELECT R.round_id, B.board_id, T.team_name
-- FROM Rounds AS R
-- JOIN Boards AS B ON B.round_id = R.round_id
-- JOIN Scores AS S ON T.team_id = S.team_id
-- JOIN Teams AS T ON 1=1

--1
-- SELECT name
-- FROM Players
-- WHERE sex = "F"
```