Peter Grantcharov pdg2116 *USED FOR SERVER March 15, 2019
Po-Chieh Liu pl2441

COMS W4111 PROJECT 1 PART 2

THREE QUERIES

1. This query will get the match outcomes (i.e. total scores) of all matches that Peter Grantcharov bet on. It will display the bet type (over/under, moneyline, or pointspread), the side that Peter Grantcharov bet on (over or under if for over/under bet, otherwise visiting team or home team), the bet amount, the odds (return = amount * odds for win), the cover (total points needed to exceed for over/under, otherwise amount of points needed for bet side to win by), the game date, and the final scores. It requires three inner joins.

```
SELECT
tmp3.name AS Type,
tmp3.odds_side AS Side,
tmp3.bet_size AS Amount,
tmp3.odds_payout AS Odds,
tmp3.odds_line AS Cover,
g.game_time AS Game_Date,
g.home_q1_score + g.home_q2_score +
g.home_q3_score + g.home_q4_score +
g.home_ot_score AS Home_Score,
g.away_q1_score + g.away_q2_score +
g.away_q3_score + g.away_q4_score +
g.away_ot_score AS Away_Scor
FROM (
  SELECT *
  FROM (
    SELECT *
    FROM (
      SELECT pb.o_id AS o_id, pb.bet_size AS bet_size
      FROM place_bet AS pb
      WHERE pb.u_id = (
        SELECT u.u_id
        FROM users AS u
        WHERE u.first_name = 'Peter'
        AND u.last_name = 'Grantcharov'
      )
    ) AS tmp1 INNER JOIN make_odds AS mo
    ON tmp1.o_id = mo.o_id
  ) AS tmp2 INNER JOIN bet_type AS bt
  ON tmp2.bt_id = bt.bt_id
) as tmp3 INNER JOIN game as g
ON tmp3.g_id = g.g_id;
```

RESULT:

```
   type    | side | amount | odds  | cover |      game_date      | home_score | away_score
-----------+------+--------+-------+-------+---------------------+------------+------------
 overunder | U    | 134.31 | 1.926 |   222 | 2018-10-31 20:00:00 |        128 |        125
 overunder | O    | 134.31 | 1.909 | 233.5 | 2018-10-26 20:00:00 |         95 |        125
(2 rows)
```

2. This query gets all instances of players who had a 40 point game. It returns the player names, the amount of points scored, and the game date.

```
SELECT p.first_name, p.last_name, tmp2.points, tmp2.game_time
FROM player AS p RIGHT OUTER JOIN (
  SELECT tmp1.p_id, tmp1.points, g.game_time
  FROM (
    SELECT pgs.p_id, pgs.points, pgs.g_id
    FROM player_game_stats AS pgs
    WHERE pgs.points > 39) AS tmp1
  INNER JOIN game AS g
  ON tmp1.g_id = g.g_id
) AS tmp2
ON p.p_id = tmp2.p_id
```

RESULT:

| first_name | last_name | points | game_time |
|---|---|---|---|
| Kemba | Walker | 41 | 2018-10-17 19:00:00 |
| Blake | Griffin | 50 | 2018-10-23 19:00:00 |
| Stephen | Curry | 51 | 2018-10-24 22:30:00 |
| Damian | Lillard | 41 | 2018-10-25 19:00:00 |
| Kevin | Durant | 41 | 2018-10-26 19:30:00 |
| Damian | Lillard | 42 | 2018-10-27 20:00:00 |
| Klay | Thompson | 52 | 2018-10-29 20:00:00 |
| Derrick | Rose | 50 | 2018-10-31 20:00:00 |

(8 rows)

3. This query gets the total amount of points scored by LeBron James in October of 2018.

```
SELECT SUM(pgs.points) AS Points
FROM player_game_stats AS pgs
WHERE pgs.p_id = (
  SELECT p.p_id
  FROM player AS p
  WHERE p.first_name = 'LeBron'
  AND p.last_name = 'James')
AND pgs.g_id IN (
  SELECT g.g_id
  FROM game AS g
  WHERE EXTRACT(MONTH FROM g.game_time) = 10
  AND EXTRACT(YEAR FROM g.game_time) = 2018
);
```

RESULT:

| points |
|---|
| 222 |

(1 row)

Peter Grantcharov pdg2116 *USED FOR SERVER March 15, 2019
Po-Chieh Liu pl2441

# APPLICATION CHANGES

Macro-level schema changes:

1) Removed REFEREE and ARENA entities (along with their relationships to the MATCH entity)
2) Created BET_TYPE entity
3) Aggregated the odds lines (attribute in MAKE_ODDS relationship between SPORTSBOOK and BET_TYPE)
4) Removed BET entity
5) Removed *at least one* participation constraint between PLAYER and PLAYED_IN, as well as for GAME and PLAYED_IN

Justification for changes:

1) We removed these two entities for the sole reason that they did not serve any purpose other than populating our database with more entities. As our application vision was developing, however, our schema became fairly involved on its own (currently contains 9 interconnected tables), and as such, we discarded them as they were no longer necessary.
2) By having a BET_TYPE entity, we significantly reduce the amount of attributes in our database, plus it allows for seamless integration of new betting types. Our previous schema had separate attributes for the different bet types (moneyline, spread, and total), and since not all sportsbooks post betting lines for all bet types for all matches, this would've resulted in a very high proportion of missing values. As a quick example, the previous MAKE_ODDS table would have been constructed of tuples in the format *(g_id, sb_id, odds_time, h_money_line, a_money_line, h_spread, v_spread, over_line, under_line, spread_value, over_under_value),* which we have now reduced to (*o_id, g_id, sb_id, bt_id, odds_time, odds_side, odds_payout, odds_line*), where bt_id is the bet type id.
3) Based on how we constructed our CREATE TABLE statements in our proposal, this was actually already done. However, our ER diagram did not properly reflect this, so we have noted this change here.
4) The BET entity showed in our ER diagram in the proposal effectively served no purpose based on how we constructed the CREATE TABLE statements, and this has been reflected in our updated version.
5) This constraint was removed because it would complicate matters when inserting new players and games into the database, as they would all have to be done by a transactional process. Further, we would like to be able to store players in the database before they have played a game, and the previous constraints would not allow this.

Micro-level schema changes:

1) The PLACE_BET table was also compressed significantly by the changes outlined in 2).
2) We added an integer *o_id* (odds_id) as a primary key to the MAKE_ODDS table as it would allow us to only have to use a single foreign key attribute when referring to a unique odds line in PLACE_BET, rather than a tuple of five attributes in MAKE_ODDS.
3) A few data types were also altered, notably *g_id* (used in four tables), was made to a VARCHAR(12).

The updated SQL schema can be found below:

```
-- stand alone tables
CREATE TABLE player (
      p_id int PRIMARY KEY,
      first_name text NOT NULL,
      last_name text NOT NULL
);
CREATE TABLE team (
      t_id int PRIMARY KEY,
      name text NOT NULL
);
CREATE TABLE users (
      u_id int PRIMARY KEY,
      name text NOT NULL,
      password text NOT NULL
);
CREATE TABLE sportsbook (
      sb_id int PRIMARY KEY,
      name text NOT NULL
);
CREATE TABLE bet_type (
      bt_id int PRIMARY KEY,
      name text NOT NULL
);

-- sportsbook table
CREATE TABLE game(
      g_id VARCHAR (12) PRIMARY KEY,
      game_time timestamp NOT NULL,
      t_id_home int REFERENCES team (t_id),
      t_id_away int REFERENCES team (t_id),
      CHECK (t_id_home != t_id_away),
      home_q1_score int NOT NULL,
      home_q2_score int NOT NULL,
      home_q3_score int NOT NULL,
      home_q4_score int NOT NULL,
      away_q1_score int NOT NULL,
      away_q2_score int NOT NULL,
      away_q3_score int NOT NULL,
      away_q4_score int NOT NULL,
      home_ot_score int,
      away_ot_score int,
      CHECK (
            home_q1_score >= 0 AND
            home_q2_score >= 0 AND
            home_q3_score >= 0 AND
            home_q4_score >= 0 AND
            away_q1_score >= 0 AND
            away_q2_score >= 0 AND
            away_q3_score >= 0 AND
            away_q4_score >= 0
      ),
```

```
	CHECK (
		home_ot_score >= 0 OR
		home_ot_score IS NULL
	),
	CHECK (
		away_ot_score >= 0 OR
		away_ot_score IS NULL
	)
);

CREATE TABLE player_game_stats (
	g_id VARCHAR (12) REFERENCES game (g_id) ON DELETE NO ACTION,
	p_id int REFERENCES player (p_id) ON DELETE NO ACTION,
	PRIMARY KEY (g_id, p_id),
	t_id int REFERENCES team (t_id),
	minutes_played float NOT NULL,
	field_goals_made int NOT NULL,
	field_goal_attempts int NOT NULL,
	three_pointers_made int NOT NULL,
	three_point_attempts int NOT NULL,
	free_throws_made int NOT NULL,
	free_throw_attempts int NOT NULL,
	offensive_rebounds int NOT NULL,
	defensive_rebounds int NOT NULL,
	assists int NOT NULL,
	steals int NOT NULL,
	blocks int NOT NULL,
	turnovers int NOT NULL,
	personal_fouls int NOT NULL,
	points int NOT NULL,
	plus_minus int NOT NULL,
	offensive_rebound_percentage float NOT NULL,
	defensive_rebound_percentage float NOT NULL,
	total_rebound_percentage float NOT NULL,
	assist_percentage float NOT NULL,
	steal_percentage float NOT NULL,
	block_percentage float NOT NULL,
	turnover_percentage float NOT NULL,
	usage_percentage float NOT NULL,
	offensive_rating int NOT NULL,
	defensive_rating int NOT NULL,
	CHECK (
		field_goals_made <= field_goal_attempts AND
		three_pointers_made <= three_point_attempts AND
		free_throws_made <= free_throw_attempts
	),
	CHECK (
		minutes_played >= 0 AND
		field_goals_made >= 0 AND
		field_goal_attempts >= 0 AND
		three_pointers_made >= 0 AND
		three_point_attempts >= 0 AND
		free_throws_made >= 0 AND
		free_throw_attempts >= 0 AND
```

```
            offensive_rebounds >= 0 AND
            defensive_rebounds >= 0 AND
            steals >= 0 AND
            blocks >= 0 AND
            turnovers >= 0 AND
            personal_fouls >= 0 AND
            points >= 0 AND
            offensive_rebound_percentage >= 0 AND
            offensive_rebound_percentage <= 100 AND
            defensive_rebound_percentage >= 0 AND
            defensive_rebound_percentage <= 100 AND
            total_rebound_percentage >= 0 AND
            total_rebound_percentage <= 100 AND
            assist_percentage >= 0 AND
            assist_percentage <= 100 AND
            steal_percentage >= 0 AND
            steal_percentage <= 100 AND
            block_percentage >= 0 AND
            block_percentage <= 100 AND
            turnover_percentage >= 0 AND
            turnover_percentage <= 100 AND
            usage_percentage >= 0 AND
            usage_percentage <= 100 AND
            offensive_rating >= 0 AND
            defensive_rating >= 0
      )
);

CREATE TABLE make_odds (
      o_id int PRIMARY KEY,
      g_id VARCHAR (12) REFERENCES game (g_id) ON DELETE NO ACTION,
      sb_id int REFERENCES sportsbook (sb_id) ON DELETE NO ACTION,
      bt_id int REFERENCES bet_type (bt_id) ON DELETE NO ACTION,
      odds_time timestamp NOT NULL,
      odds_side char NOT NULL,
      CHECK (
            odds_side = 'H' OR
            odds_side = 'V' OR
            odds_side = 'O' OR
            odds_side = 'U'
      ),
      UNIQUE (g_id, sb_id, bt_id, odds_time, odds_side),
      odds_payout float NOT NULL,
      CHECK (odds_payout >= 1), -- will use decimal odds
      odds_line float NOT NULL
);
CREATE TABLE place_bet(
      o_id int REFERENCES make_odds (o_id),
      bet_time timestamp NOT NULL,
      u_id int REFERENCES users (u_id) ON DELETE NO ACTION,
      bet_size decimal(12, 2) NOT NULL,
      CHECK (bet_size > 0),
      PRIMARY KEY (o_id, bet_time, u_id)
);
```