데이터베이스 과제

날짜: 2025년 5월 28일

이름: 문시원

학번: 32211522

◆ **주제 :** 지역 아마추어 농구 리그 관리 시스템

◆ **설명 :**

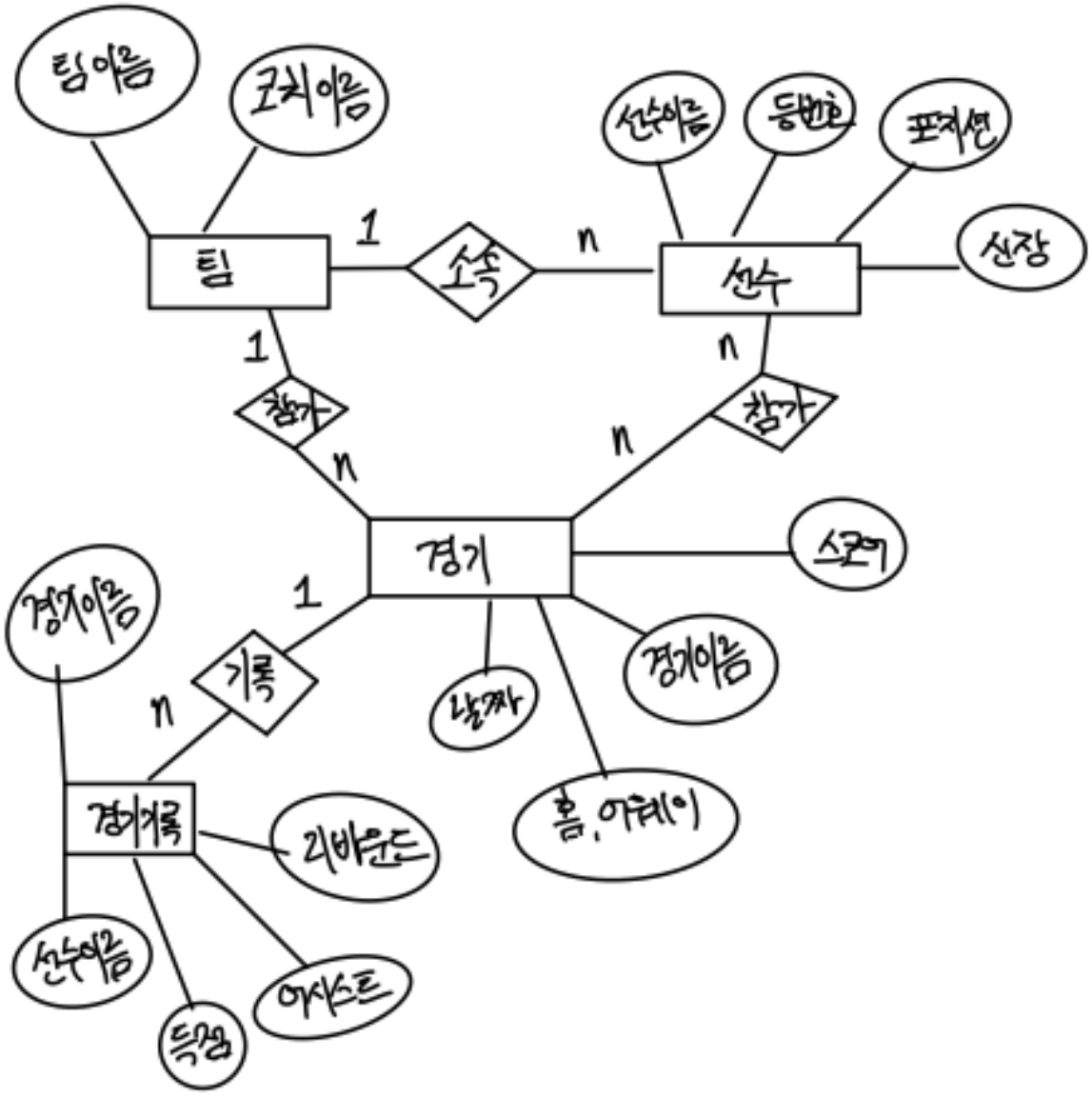
지역 아마추어 농구 리그 관리 시스템 프로젝트는 지역 사회 또는 학교/동호회를 중심으로 운영되는 농구리그를 효율적으로 관리할 수 있는 데이터베이스를 설계하는 것을 목표로 합니다. 이 관리 시스템은 리그에 참가하는 여러 팀과 선수 정보를 체계화해서 저장한 후, 각 경기의 결과와 선수별 경기 기록(득점, 리바운드, 어시스트 등)을 함께 관리 할 수 있도록 구성되어 있습니다. 이 데이터베이스를 통해 다음과 같은 서비스를 제공할 수 있습니다.

- 리그 참가 팀 및 선수 등록
- 경기 일정 및 결과 기록
- 선수별 경기 통계 저장 및 조회
- 팀 간 경기 결과 및 통계
- 특정 기간 혹은 특정 경기 기준 주요 활약 선수 랭킹

또한, SELECT 쿼리를 통해 경기별 주요 선수 확인, 팀 평균 점수 분석, 선수별 누적 기록 등을 효과적으로 조회할 수 있는 구조로 설계했습니다.

이러한 시스템은 학교, 동호회 등에서 실제 운영중인 농구 대회의 행정 효율성과 경기 기록 관리의 정확도 향상에 크게 기여할 수 있습니다.

◆ **ER 다이어그램 이미지**


팀이름
코치이름
팀
1
소속
n
선수
선수이름
등번호
포지션
신장
1
참가
n
n
참가
n
경기
스코어
1
기록
n
경기이름
날짜
경기이름
경기기록
리바운드
흠, 어테이
선수이름
득점
어시스트

◆ 사용 시나리오 쿼리 설명 및 관련 SQL 문

시나리오 1: 특정 경기에서 득점이 가장 높은 선수 조회

```sql
SELECT P.Player_name, T.team_name, GS.points
FROM Game_Stats GS
JOIN Player P ON GS.player_id = P.player_id
JOIN Team T ON P.team_id = T.team_id
WHERE GS.game_id = 1001
ORDER BY GS.points DESC
LIMIT 1;
```

50:118

Result Grid | Filter Rows: Search | Export: | Fetch rows:

Player_name	team_name	points
Kim one	Seoul Eagles	34

시나리오 2: 각 팀의 평균 득점(홈팀) 조회

```sql
SELECT T.team_name, AVG(G.home_score) AS avg_home_score
FROM Game G
JOIN Team T ON G.home_team_id = T.team_id
GROUP BY T.team_name;
```

50:118

Result Grid | Filter Rows: Search | Export:

team_name	avg_home_score
Seoul Eagles	75.0000
Busan Sharks	81.0000

시나리오 3: 선수별 누적 득점, 어시스트, 리바운드 통계 조회

```sql
SELECT P.player_name, T.team_name,
    SUM(GS.points) AS total_points,
    SUM(GS.assists) AS total_assists,
    SUM(GS.rebounds) AS total_rebounds
FROM Game_Stats GS
JOIN Player P ON GS.player_id = P.player_id
JOIN Team T ON P.team_id = T.team_id
GROUP BY P.player_id, P.player_name, T.team_name;
```

50:118

Result Grid | Filter Rows: Search | Export:

player_name	team_name	total_poin...	total_assists	total_rebou...
Kim one	Seoul Eagles	113	20	27
Kim two	Seoul Eagles	15	2	1
Kim three	Seoul Eagles	3	5	5
Kim four	Seoul Eagles	5	5	8
Kim five	Seoul Eagles	18	2	10
Lee six	Busan Sharks	42	7	11
Lee seven	Busan Sharks	29	5	7
Lee eight	Busan Sharks	21	18	14
Lee nine	Busan Sharks	21	11	20
Lee ten	Busan Sharks	36	8	6

◆ MySQL Action Output 및 DB 생성을 위해 사용한 SQL 코드 전체

	Time	Action	Response	Duration / Fetch Time
2	17:34:08	CREATE...	1 row(s) affected	0.00065 sec
3	17:34:08	USE bask...	0 row(s) affected	0.00015 sec
4	17:34:08	CREATE...	0 row(s) affected	0.0057 sec
5	17:34:08	CREATE...	0 row(s) affected	0.0073 sec
6	17:34:08	CREATE...	0 row(s) affected	0.0053 sec
7	17:34:08	CREATE...	0 row(s) affected	0.0060 sec
8	17:34:08	INSERT I...	1 row(s) affected	0.00062 sec
9	17:34:08	INSERT I...	1 row(s) affected	0.00035 sec
10	17:34:08	INSERT I...	1 row(s) affected	0.00073 sec
11	17:34:08	INSERT I...	1 row(s) affected	0.00034 sec
12	17:34:08	INSERT I...	1 row(s) affected	0.00026 sec
13	17:34:08	INSERT I...	1 row(s) affected	0.00023 sec
14	17:34:08	INSERT I...	1 row(s) affected	0.00023 sec
15	17:34:08	INSERT I...	1 row(s) affected	0.00028 sec
16	17:34:08	INSERT I...	1 row(s) affected	0.00022 sec
17	17:34:08	INSERT I...	1 row(s) affected	0.00022 sec
18	17:34:08	INSERT I...	1 row(s) affected	0.00022 sec
19	17:34:08	INSERT I...	1 row(s) affected	0.00022 sec
20	17:34:08	INSERT I...	1 row(s) affected	0.00064 sec
21	17:34:08	INSERT I...	1 row(s) affected	0.00023 sec
22	17:34:08	INSERT I...	1 row(s) affected	0.00043 sec
23	17:34:08	INSERT I...	1 row(s) affected	0.00021 sec
24	17:34:08	INSERT I...	1 row(s) affected	0.00019 sec
25	17:34:08	INSERT I...	1 row(s) affected	0.00023 sec
26	17:34:08	INSERT I...	1 row(s) affected	0.00018 sec
27	17:34:08	INSERT I...	1 row(s) affected	0.00018 sec
28	17:34:08	INSERT I...	1 row(s) affected	0.00020 sec
29	17:34:08	INSERT I...	1 row(s) affected	0.00019 sec
30	17:34:08	INSERT I...	1 row(s) affected	0.00020 sec
31	17:34:08	INSERT I...	1 row(s) affected	0.00022 sec
32	17:34:08	INSERT I...	1 row(s) affected	0.00017 sec
33	17:34:08	INSERT I...	1 row(s) affected	0.00017 sec
34	17:34:08	INSERT I...	1 row(s) affected	0.00019 sec
35	17:34:08	INSERT I...	1 row(s) affected	0.00019 sec
36	17:34:08	INSERT I...	1 row(s) affected	0.00021 sec
37	17:34:08	INSERT I...	1 row(s) affected	0.00019 sec
38	17:34:08	INSERT I...	1 row(s) affected	0.00018 sec
39	17:34:08	INSERT I...	1 row(s) affected	0.00019 sec
40	17:34:08	INSERT I...	1 row(s) affected	0.00020 sec
41	17:34:08	INSERT I...	1 row(s) affected	0.00019 sec
42	17:34:08	SELECT...	1 row(s) returned	0.00038 sec / 0.000...
43	17:34:08	SELECT...	2 row(s) returned	0.00027 sec / 0.0000...
44	17:34:08	SELECT...	10 row(s) returned	0.00025 sec / 0.0000...

```
-- 데이터베이스 생성 및 사용
DROP DATABASE IF EXISTS basketball_league;
CREATE DATABASE basketball_league;
USE basketball_league;



--  팀 테이블
CREATE TABLE Team (
        team_id Int primary key,
    team_name varchar(50) Not null,
    coach_name varchar(50)
);

-- 선수 테이블
CREATE TABLE Player (
        player_id INT primary key,
    player_name VARCHAR(50) NOT NULL,
    position VARCHAR(10),
    height INT,
    jersey_number INT,
    team_id INT,
    FOREIGN KEY (team_id) REFERENCES Team(team_id),
    UNIQUE (team_id, jersey_number) -- 팀 내에서는 등번호 중복 불가
);

-- 경기 테이블
CREATE TABLE Game (
```

```
    game_id INT PRIMARY KEY,
    game_date DATE NOT NULL,
    home_team_id INT,
    away_team_id INT,
    home_score INT,
    away_score INT,
    FOREIGN KEY (home_team_id) REFERENCES Team(team_id),
    FOREIGN KEY (away_team_id) REFERENCES Team(team_id)
    );

-- 경기기록 테이블
CREATE TABLE Game_stats (
    stat_id INT PRIMARY KEY,
    game_id INT,
    player_id INT,
    points INT,
    assists INT,
    rebounds INT,
    FOREIGN KEY (game_id) REFERENCES Game(game_id),
    FOREIGN KEY (player_id) REFERENCES Player(player_id)
);

-- 샘플 데이터 삽입

INSERT INTO Team VALUES (1, 'Seoul Eagles', 'Kim Coach');
INSERT INTO Team VALUES (2, 'Busan Sharks', 'Lee Coach');
```

```
-- 선수 데이터
INSERT INTO Player VALUES (101, 'Kim one', 'Gaurd', 178, 2, 1);
INSERT INTO Player VALUES (102, 'Kim two', 'Gaurd', 180, 7, 1);
INSERT INTO Player VALUES (103, 'Kim three', 'Forward', 183, 11, 1);
INSERT INTO Player VALUES (104, 'Kim four', 'Forward', 186, 15, 1);
INSERT INTO Player VALUES (105, 'Kim five', 'Center',190, 35, 1);

INSERT INTO Player VALUES (201, 'Lee six', 'Gaurd', 180, 2, 2);
INSERT INTO Player VALUES (202, 'Lee seven', 'Gaurd', 181, 4, 2);
INSERT INTO Player VALUES (203, 'Lee eight', 'Forward', 187, 10, 2);
INSERT INTO Player VALUES (204, 'Lee nine', 'Forward', 187, 23, 2);
INSERT INTO Player VALUES (205, 'Lee ten', 'Center', 188, 34, 2);

INSERT INTO Game VALUES (1001, '2025-01-01', 1, 2, 75, 68);
INSERT INTO Game VALUES (1002, '2025-01-02', 2, 1, 81, 79);

-- 경기 1001 기록
INSERT INTO Game_Stats VALUES (1 , 1001, 101, 34, 5, 3);
INSERT INTO Game_Stats VALUES (2 , 1001, 102, 15, 2, 1);
INSERT INTO Game_Stats VALUES (3 , 1001, 103, 3, 5, 5);
INSERT INTO Game_Stats VALUES (4 , 1001, 104, 5, 5, 8);
INSERT INTO Game_Stats VALUES (5 , 1001, 105, 18, 2, 10);

INSERT INTO Game_Stats VALUES (6 , 1001, 201, 19, 5, 3);
INSERT INTO Game_Stats VALUES (7 , 1001, 202, 14, 1, 2);
INSERT INTO Game_Stats VALUES (8 , 1001, 203,11, 8, 9);
INSERT INTO Game_Stats VALUES (9 , 1001, 204, 9, 10, 10);
```

```
INSERT INTO Game_Stats VALUES (10 , 1001, 205, 15, 5, 3);

-- 경기 1002 기록
INSERT INTO Game_Stats VALUES (11, 1002, 101, 21, 4, 2);
INSERT INTO Game_Stats VALUES (12, 1002, 101, 13, 5, 7);
INSERT INTO Game_Stats VALUES (13, 1002, 101, 16, 1, 6);
INSERT INTO Game_Stats VALUES (14, 1002, 101, 11, 3, 4);
INSERT INTO Game_Stats VALUES (15, 1002, 101, 18, 2, 5);

INSERT INTO Game_Stats VALUES (16, 1002, 201, 23, 2, 8);
INSERT INTO Game_Stats VALUES (17, 1002, 202, 15, 4, 5);
INSERT INTO Game_Stats VALUES (18, 1002, 203, 10, 10, 5);
INSERT INTO Game_Stats VALUES (19, 1002, 204, 12, 1, 10);
INSERT INTO Game_Stats VALUES (20, 1002, 205, 21, 3, 3);


SELECT P.Player_name, T.team_name, GS.points
FROM Game_Stats GS
JOIN Player P ON GS.player_id = P.player_id
JOIN Team T ON P.team_id = T.team_id
WHERE GS.game_id = 1001
ORDER BY GS.points DESC
LIMIT 1;

SELECT T.team_name, AVG(G.home_score) AS avg_home_score
FROM Game G
JOIN Team T ON G.home_team_id = T.team_id
```

```
GROUP BY T.team_name;


SELECT P.player_name, T.team_name,
        SUM(GS.points) AS total_points,
    SUM(GS.assists) AS total_assists,
    SUM(GS.rebounds) AS total_rebounds
FROM Game_Stats GS
JOIN Player P ON GS.player_id = P.player_id
JOIN Team T ON P.team_id = T.team_id
GROUP BY P.player_id, P.player_name, T.team_name;
```