SQL PROJECT

SPORT MANAGEMENT SYSTEM

✞ Abstract:-

The article considers sports management as a new sport scientific direction, which is in demand in the sports world today. Sports management in its essence is the sports management, the formation of a coordinated sports team; this is the management of sport organizations at the micro- and macro-level, aimed at promoting the sport development. The purpose of the research is the conceptualization of sports management as an effective factor, which is based on the activity of sports organizations.

✞ Aim of Project –

The Aim Sport is at the forefront of Sport Marketing and Digital Perimeter Advertising, Scottish Disability Sport have developed the activity inclusion model to help sport respond to the needs of disabled people.

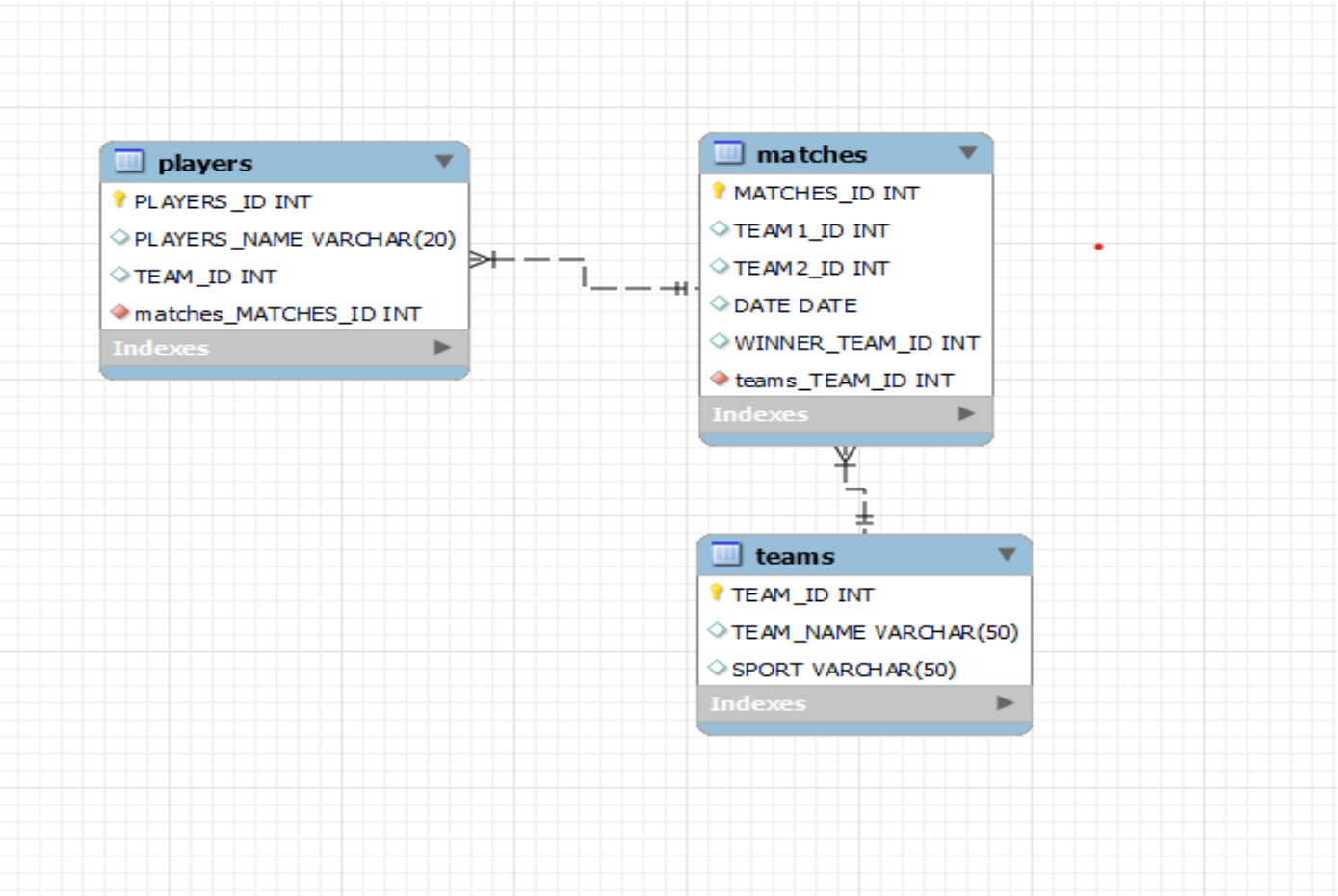
✞ Objective of project –

Develop physical talents to their maximum potential. Engage in competitive activities, while promoting sound health, safety, and physical fitness. Exemplify good conduct as a means for learning good citizenship.

✞ Introduction-

Sport pertains to any form of physical activity or game, often competitive and organized that aims to use, maintain, or improve physical ability and skills while providing enjoyment to participants and, in some cases, entertainment to spectators.

ER Diagram


players
PLAYERS_ID INT
PLAYERS_NAME VARCHAR(20)
TEAM_ID INT
matches_MATCHES_ID INT
Indexes
matches
MATCHES_ID INT
TEAM1_ID INT
TEAM2_ID INT
DATE DATE
WINNER_TEAM_ID INT
teams_TEAM_ID INT
Indexes
teams
TEAM_ID INT
TEAM_NAME VARCHAR(50)
SPORT VARCHAR(50)
Indexes

STRUCTURE OF TABLES

- **PLAYERS TABLE**

```
7 •  DESC PLAYERS;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
PLAYERS_ID	int	NO	PRI	NULL	
PLAYERS_NAME	varchar(20)	YES		NULL	
TEAM_ID	int	YES		NULL	

- **TEAM TABLE**

```
23 •  DESC TEAMS;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
TEAM_ID	int	NO	PRI	NULL	
TEAM_NAME	varchar(50)	YES		NULL	
SPORT	varchar(50)	YES		NULL	

- **MATCHES TABLE**

```
36 •  DESC MATCHES;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
MATCHES_ID	int	NO	PRI	NULL	
TEAM1_ID	int	YES		NULL	
TEAM2_ID	int	YES		NULL	
DATE	date	YES		NULL	
WINNER_TEAM_ID	int	YES		NULL	

Contents of Tables

❖ PLAYERS

```
7 •  SELECT * FROM PLAYERS;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

PLAYERS_ID	PLAYERS_NAME	TEAM_ID
2	sarah	1
3	mike	2
4	emily	2
5	david	3
6	lisa	3
7	tom	4
8	olivia	4
9	daniel	5
10	sophia	5

PLAYERS 4

❖ TEAMS

```
23 •  SELECT * FROM TEAMS;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Cont

TEAM_ID	TEAM_NAME	SPORT
1	Team A	Football
2	Team B	Basketball
3	Team C	Soccer
4	Team D	Volleyball
5	Team E	Tennis
NULL	NULL	NULL

❖ MATCHES

```
37 •    SELECT * FROM MATCHES;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Cont

MATCHES_ID	TEAM1_ID	TEAM2_ID	DATE	WINNER_TEAM_ID
1	1	2	2023-01-01	1
2	2	1	2023-01-15	2
3	3	4	2023-02-01	3
4	4	3	2023-02-01	4
5	5	5	2023-03-01	5
6	1	3	2023-03-15	3
7	2	4	2023-04-15	2
8	4	2	2023-04-01	4
9	5	1	2023-05-01	1

MATCHES 6

AGGREGATE

1. COUNT THE NUMBER OF PLAYERS IN EACH TEAM

```
52 •  SELECT TEAM_ID, COUNT(PLAYERS_NAME) AS PLAYERS
53    FROM PLAYERS
54    GROUP BY  TEAM_ID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TEAM_ID	PLAYERS
1	2
2	2
3	2
4	2
5	2

2. CALCULATE THE AVERAGE NUMBER OF PLAYERS PER TEAM

```
56 •  SELECT AVG(PLAYERCOUNT) AS AVGTEAMSIZE
57    FROM (SELECT TEAM_ID, COUNT(PLAYERS_ID) AS PLAYERCOUNT FROM PLAYERS
58    GROUP BY TEAM_ID) AS TEAMCOUNTS;
59
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AVGTEAMSIZE
2.0000

3. FIND THE TEAM WITH THE HIGHEST NUMBER OF PLAYERS

```
61 •  SELECT TEAM_ID, COUNT(PLAYERS_ID) AS PLAYERCOUNT
62    FROM PLAYERS
63    GROUP BY TEAM_ID ORDER BY PLAYERCOUNT DESC LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

TEAM_ID	PLAYERCOUNT
1	2

4. DETERMINE THE TOTAL NUMBER OF MATCHES

```
65 •    SELECT COUNT(MATCHES_ID) AS TOTALMATCHES
66      FROM MATCHES;
67
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TOTALMATCHES
10

5. CALCULATE THE AVERAGE DATE OF MATCHES

```
69 •    SELECT AVG(DATE) AS AVG_MATCH_DATE
70      FROM MATCHES;
71
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AVG_MATCH_DATE
20230306.6000

JOINTS

INNER JOIN PLAYERS AND TEAMS

- DISPLAY THE NAMES OF PLAYERS AND THEIR RESPECTIVE TEAMS

```
75 •  SELECT PLAYERS.PLAYERS_NAME, TEAMS.TEAM_NAME
76    FROM PLAYERS
77    INNER JOIN TEAMS ON PLAYERS.TEAM_ID= TEAMS.TEAM_ID;
78
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PLAYERS_NAME	TEAM_NAME
mike	Team B
emily	Team B
david	Team C
lisa	Team C
tom	Team D
olivia	Team D
daniel	Team E
sophia	Team E

LEFT JOIN TEAMS AND MATCHES

- GET A LIST OF ALL TEAMS AND THEIR MATCH INFORMATION

```
81 •  SELECT TEAMS.TEAM_NAME, MATCHES.MATCHES_ID, MATCHES.DATE
82    FROM TEAMS
83    LEFT JOIN MATCHES ON TEAMS.TEAM_ID= MATCHES.TEAM1_ID OR TEAMS.TEAM_ID= MATCHES.TEAM2_ID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TEAM_NAME	MATCHES_ID	DATE
Team C	3	2023-02-01
Team D	8	2023-04-01
Team D	7	2023-04-15
Team D	4	2023-02-01
Team D	3	2023-02-01
Team E	10	2023-05-15
Team E	9	2023-05-01
Team E	5	2023-03-01

Result 13

CROSS JOIN PLAYERS

- CREATE LIST OF ALL POSSIBLE PLAYER-TEAM COMBINATIONS

```
87 •  SELECT PLAYERS.PLAYERS_NAME, TEAMS.TEAM_NAME
88    FROM PLAYERS
89    CROSS JOIN TEAMS;
90
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

PLAYERS_NAME	TEAM_NAME
john	Team E
john	Team D
john	Team C
john	Team B
john	Team A
sarah	Team E
sarah	Team D
sarah	Team C
sarah	Team B

Result 14

SELF JOIN PLAYERS

- FIND PAIRS OF PLAYERS FROM THE SAME TEAM

```
93 •  SELECT A.PLAYERS_NAME AS PLAYER1, B.PLAYERS_NAME AS PLAYER2, A.TEAM_ID
94    FROM PLAYERS A
95    INNER JOIN PLAYERS B ON A.TEAM_ID = B.TEAM_ID AND A.PLAYERS_ID < B.PLAYERS_ID;
96
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PLAYER1	PLAYER2	TEAM_ID
john	sarah	1
mike	emily	2
david	lisa	3
tom	olivia	4
daniel	sophia	5

MULTIPLE JOINS (PLAYERS, MATCHES, AND TEAMS)

- DISPLAY PLAYERS NAMES, MATCH DATES, AND THE CORRESPONDING TEAM NAMES FOR MATCHES THEY PARTICIPATED IN

```
99  SELECT PLAYERS.PLAYERS_NAME, MATCHES.DATE, TEAMS.TEAM_NAME
100 FROM PLAYERS
101 INNER JOIN MATCHES ON PLAYERS.TEAM_ID= MATCHES.TEAM1_ID OR PLAYERS.TEAM_ID= MATCHES.TEAM2_ID
102 INNER JOIN TEAMS ON PLAYERS.TEAM_ID = TEAMS.TEAM_ID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PLAYERS_NAME	DATE	TEAM_NAME
emily	2023-01-01	Team B
mike	2023-01-01	Team B
sarah	2023-01-01	Team A
john	2023-01-01	Team A
emily	2023-01-15	Team B
mike	2023-01-15	Team B
sarah	2023-01-15	Team A
john	2023-01-15	Team A
olivia	2023-02-01	Team D

Result 16

QUERIES AND SUBQUERIES

SUBQUERY TO FIND TEAM WITH MOST WINS

- FIND THE TEAM WITH THE MOST WINS IN THE MATCHES TABLE

```
107    SELECT TEAM_NAME
108    FROM TEAMS
109    WHERE TEAM_ID = ( SELECT WINNER_TEAM_ID
110    FROM MATCHES
111    GROUP BY WINNER_TEAM_ID
112     ORDER BY COUNT(*) DESC LIMIT 1);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TEAM_NAME
Team A

SUBQUERY TO CALCULATE PLAYER COUNT PER TEAM

- DETERMINE THE NUMBER OF PLAYERS IN EACH TEAM

```
116    SELECT TEAM_NAME, ( SELECT COUNT(*)
117    FROM PLAYERS
118    WHERE PLAYERS.TEAM_ID = TEAMS.TEAM_ID)
119    AS PLAYERCOUNT FROM TEAMS;
120
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TEAM_NAME	PLAYERCOUNT
Team A	2
Team B	2
Team C	2
Team D	2
Team E	2

SUBQUERY TO FIND MATCHES WITH TEAMS FROM HE SAME SPORT

- LIST MATCHES WHERE BOTH TEAMS BELONG TO THE SAME SPORT

```
123 •   SELECT MATCHES_ID , DATE
124     FROM MATCHES
125     WHERE TEAM1_ID IN
126     (SELECT TEAM_ID FROM TEAMS WHERE SPORT = "Basketball")
127     AND TEAM2_ID IN (SELECT TEAM_ID FROM TEAMS
128     WHERE SPORT = "Basketball");
129
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

MATCHES_ID	DATE
NULL	NULL

SUBQUERY TO CALCULATTE AVERAGE TEAM SIZE

- CALCULATE THE AVERAGE NUMBER OF PLAYERS PER TEAM

```
132 •   SELECT AVG(PLAYERCOUNT) AS AVY_TEAM_SIZE
133     FROM( SELECT TEAM_ID, COUNT(*) AS PLAYERCOUNT
134     FROM PLAYERS
135     GROUP BY TEAM_ID) AS TEAMCOUNTS;
136
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AVY_TEAM_SIZE
2.0000

SUBQUERY TO FIND PLAYERS WITH NO MATCHES

- IDENTIFY PLAYERS WHO HAVEN'T PARTICIPATED IN ANY MATCHES

```
139    SELECT PLAYERS_NAME
140    FROM PLAYERS
141    WHERE PLAYERS_ID NOT IN (SELECT DISTINCT TEAM1_ID AS PLAYERS_ID FROM MATCHES
142    UNION SELECT DISTINCT TEAM2_ID AS PLAYERS_ID FROM MATCHES);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PLAYERS_NAME
lisa
tom
olivia
daniel
sophia