**PROJECT REPORT**

(DATABASE MANAGEMENT SYSTEM)

### THAPAR OLYMPICS SYSTEM

**SUBMITTED BY:**

PRATISTHA KAMAL (101253002)

PIYUSH GUPTA (101203071)

NAMAN GUPTA (101203060)

NAMAN GARG (101253010)

**UNDER THE GUIDANCE OF:**

**Dr. Parteek Bhatia**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**THAPAR UNIVERSITY, PATIALA**

**REQUIREMENT ANALYSIS**

The automated **THAPAR OLYMPICS SYSTEM** has been so designed that this system automates the registration and the organization of fixtures of different sports held in University. The project reduces the paper work and redundancy encountered during team and member registration.

This system is designed in such a way that the students as well as the faculty members have to get themselves registered in order to be a part of Thapar Olympics. The registration includes **name, roll number (if student), gender, email id, phone number**, **captain (or not), faculty name, department (if faculty), sport.**

The details of the account are sent to the captain of the team. The captain then logs into the account. Login consists of **team\_name** and **password**. After verification of the login, the captain can see all the **fixtures**, **scores** and **points** and also the **details of the team member**, **date** and **time** of the matches.

The **team** table contains the data of the **team\_id**, **sport\_id**, **faculty\_id**, **vc\_id** (vice captain), **captain\_id**.

A team can register in more than one sport and sport must have at least 2 teams. There is a **total participation** of team entity set and also **total participation** of sport entity set in this relationship.

Those who have not registered can only see Fixtures, scores and points, gallery, win/lose for every sport.

The sport contains the **sport\_id** and **sport\_name**. Every sport is covered and has unique **sport\_id**.

There is one separate table for **athletics**. This contains **member\_id**, **sport\_id**, **time**, **distance** and **status**. Events like 100m, 200m, hurdle race, relays, high jump, long jump, shot put, javelin throw, etc are covered.

After the match has been played the admin updates the **score** table containing **match\_id** and **team\_id**, and also updates the **overall\_point** table which contains **team\_id** and **points**.

While the match is going on the **match\_point** table which contains **sport\_id**, **team\_id**, **matches\_played**, **matches\_won**, **draw**, **nrr** are updated by the admin.

All the tables shown below have already been normalized and are present in their best non reducible form.

1Normal Form: All repeatable and non-repeatable values attributes have been separated using the decomposition process.

2 Normal Form: All the attributes are fully functional dependent on their respective primary keys.

3 Normal Form: All transitive dependency in between the attributes has been removed by separating the tables.

BCNF Form: Each and every determinant should be a candidate key.

4 Normal Form: no attribute has multi values dependence.

**IDENTIFYING THE TABLES AND THE CONSTRAINTS**

1. **sport**

The constraints, attributes and the types are as follows:

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| SPORT\_ID | NOT NULL | NUMBER(10) |
| SPORT\_NAME |  | VARCHAR2(20) |

Primary key: **sport\_id**

FD Diagram:

**Sport\_id**

Sport\_name

Normalized Form: 1, 2, 3, BCNF, 4.

1. **team\_master**

The constraints, attributes and the types are as follows:

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| TEAM\_ID | NOT NULL | NUMBER(10) |
| TEAM\_NAME |  | VARCHAR2(20) |
| TEAM\_PASS |  | VARCHAR2(20) |

Primary key: **team\_id**

FD Diagram:

Team\_name

Team\_id

Team\_pass

Normalized Forms: 1, 2, 3, BCNF, 4.

1. **department**

The constraints, attributes and the types are as follows:

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| DEPT\_ID | NOT NULL | NUMBER(10) |
| DEPT\_NAME |  | VARCHAR2(20) |
| TEAM\_ID |  | NUMBER(10) |

Primary key: **dept\_id**

Foreign key: **team\_id**

FD Diagram:

Dept\_name

Dept\_id

Team\_id

Normalized Forms: 1, 2, 3, BCNF, 4.

1. **team**

The constraints, attributes and the types are as follows:

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| TEAM\_ID |  | NUMBER(10) |
| SPORT\_ID |  | NUMBER(10) |
| CAPTAIN\_ID |  | VARCHAR2(20) |
| FACULTY\_ID1 |  | VARCHAR2(20) |
| FACULTY\_ID2 |  | VARCHAR2(20) |
| VC\_ID1 |  | VARCHAR2(20) |
| VC\_ID2 |  | VARCHAR2(20) |

Foreign keys: **team\_id** (referenced to **team\_master**),

**sport\_id** (referenced to **sport**),

**captain\_id, faculty\_id1, faculty\_id2, vc\_id1, vc\_id2**

(referenced to **member\_master**)

FD Diagram:

Captain\_id

Team\_id

Sport\_id

Faculty\_id1

Faculty\_id2

Vc\_id1

Vc\_id2

Normalized Forms: 1, 2, 3, BCNF, 4.

1. **member\_master**

The constraints, attributes and the types are as follows:

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| MEMBER\_ID | NOT NULL | VARCHAR2(20) |
| ROLL\_NO |  | NUMBER(10) |
| FACULTY\_ID |  | NUMBER(10) |
| FNAME |  | VARCHAR2(50) |
| LNAME |  | VARCHAR2(50) |
| GENDER |  | VARCHAR2(2) |
| EMAIL |  | VARCHAR2(20) |
| PH\_NO |  | NUMBER(10) |
| DEPT\_ID |  | NUMBER(10) |

Primary key: **member\_id**

Unique key: **faculty\_id, roll\_no**

Foreign key: **dept\_id** (referenced to **department**)

FDD Diagram:

Roll\_no faculty\_id

Member\_id

f\_name l\_name

Gender dept\_id

Email phn\_no

email and phn\_no are multi-valued attributes so we apply 4th Normal Form and we create separate tables:

Table 1: roll\_no, f\_name, l\_name, gender, faculty\_id, dept\_id, member\_id

Table 2: member\_id, email

Table 3: member\_id, phn\_no

1. **fixture**

The constraints, attributes and the types are as follows:

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| SPORT\_ID |  | NUMBER(10) |
| MATCH\_ID | NOT NULL | NUMBER(10) |
| TEAM\_ID1 |  | NUMBER(10) |
| TEAM\_ID2 |  | NUMBER(10) |
| STATUS |  | VARCHAR2(10) |
| MATCH\_DATE |  | DATE |
| WINNER |  | NUMBER(10) |

Primary key: **match\_id**

Foreign key: **sport\_id** (referenced to **sport**),

**Team\_id1, team\_id2** (referenced to **team\_master**)

FDD Diagram:

sport\_id team\_id1

Match\_id

status winner

match\_date team\_id2

1. **athletics**

The constraints, attributes and the types are as follows:

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| MEMBER\_ID |  | VARCHAR2(20) |
| SPORT\_ID |  | NUMBER(10) |
| TIME |  | DATE |
| DISTANCE |  | NUMBER(10) |
| STATUS |  | VARCHAR2(1) |

Foreign key: **member\_id** (referenced to **member\_master**),

**Sport\_id** (referenced to **sport**)

FDD Diagram:

time

Member\_id

Sport\_id

distance

status

1. **member\_sport**

The constraints, attributes and the types are as follows:

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| MEMBER\_ID |  | VARCHAR2(20) |
| SPORT\_ID |  | NUMBER(10) |

Foreign key: **member\_id** (referenced to **member\_master**),

**Sport\_id** (referenced to **sport**)

FD Diagram:

Member\_id

Sport\_id

1. **score**

The constraints, attributes and the types are as follows:

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| MATCH\_ID |  | NUMBER(10) |
| TEAM\_ID |  | NUMBER(10) |
| SCORE1 |  | NUMBER(10) |
| SCORE2 |  | NUMBER(10) |
| SCORE3 |  | NUMBER(10) |
| SCORE4 |  | NUMBER(10) |
| SCORE5 |  | NUMBER(10) |

Foreign key: **match\_id** (referenced to **fixture**),

**Team\_id** (referenced to **team\_master**)

FD Diagram:

Score1

Match\_id

Team\_id

score2 score3

score4 score5

1. **match\_point**

The constraints, attributes and the types are as follows:

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| SPORT\_ID |  | NUMBER(10) |
| TEAM\_ID |  | NUMBER(10) |
| MATCHES\_PLAYED |  | NUMBER(10) |
| MATCHES\_WON |  | NUMBER(10) |
| POINTS |  | NUMBER(10) |
| NRR |  | NUMBER(10) |
| DRAW |  | NUMBER(10) |
| SCORE\_A |  | NUMBER(10) |
| SCORE\_F |  | NUMBER(10) |
| WINPC |  | NUMBER(10) |

Foreign key: **sport\_id** (referenced to **sport**),

**Team\_id** (referenced to **team\_master**),

FD Diagram:

Sport\_id

Team\_id

winpc points

score\_a draw

score\_f nrr

matches\_won matches\_played

1. **overall\_point**

The constraints, attributes and the types are as follows:

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| TEAM\_ID |  | NUMBER(10) |
| POINTS |  | NUMBER(10) |

Foreign key: **team\_id** (referenced to **team\_master**)

FD Diagram:

Team\_id

Points

**ER DIAGRAM**

**SOURCE CODE….**

create table sport(

sport\_id number(10),

sport\_name varchar(20),

constraint sport\_id\_pk primary key(sport\_id)

);

create table team\_master(

team\_id number(10),

team\_name varchar(20),

team\_pass varchar(20),

constraint team\_id\_pk primary key(team\_id)

);

create table department(

dept\_id number(10),

dept\_name varchar(20),

team\_id number(10),

constraint department\_team\_id\_fk foreign key (team\_id) references team\_master(team\_id) on delete set null,

constraint dept\_id\_pk primary key (dept\_id)

);

create table team(

team\_id number(10),

sport\_id number(10),

captain\_id number(10),

faculty\_id1 number(10),

faculty\_id2 number(10),

vc\_id1 number(10),

vc\_id2 number(10),

constraint team\_team\_id\_fk foreign key(team\_id) references team\_master(team\_id) on delete cascade,

constraint team\_sport\_id\_fk foreign key(sport\_id) references sport(sport\_id) on delete cascade,

constraint team\_captain\_id\_fk foreign key(captain\_id) references member\_master(member\_id) on delete set null,

constraint team\_vc\_id1\_fk foreign key(vc\_id1) references member\_master(member\_id) on delete set null,

constraint team\_vc\_id2\_fk foreign key(vc\_id2) references member\_master(member\_id) on delete set null,

constraint team\_faculty\_id1\_fk foreign key(faculty\_id1) references member\_master(member\_id) on delete set null,

constraint team\_faculty\_id2\_fk foreign key(faculty\_id2) references member\_master(member\_id) on delete set null

);

create table member\_master(

member\_id number(10),

roll\_no number(10) default null,

faculty\_id number(10) default null,

fname varchar(50),

lname varchar(50),

gender varchar(2),

email varchar(20),

ph\_no number(10),

dept\_id number(10),

constraint member\_id\_pk primary key(member\_id),

constraint faculty\_id\_unique unique(faculty\_id),

constraint roll\_no\_unique unique(roll\_no),

constraint member\_master\_dept\_id\_fk foreign key(dept\_id) references department(dept\_id) on delete set null

);

create table fixture(

sport\_id number(10),

match\_id number(10),

team\_id1 number(10),

team\_id2 number(10),

status varchar(10),

match\_date date,

winner number(10) default null,

constraint match\_id\_pk primary key(match\_id) ,

constraint fixture\_sport\_id\_fk foreign key(sport\_id) references sport(sport\_id) on delete cascade,

constraint fixture\_team\_id1\_fk foreign key(team\_id1) references team\_master(team\_id) on delete cascade,

constraint fixture\_team\_id2\_fk foreign key(team\_id2) references team\_master(team\_id) on delete cascade,

constraint fixture\_status check(status in('prelims','semis','finals'))

);

create table athletics(

member\_id number(10),

sport\_id number(10),

time date,

distance number(10),

status varchar(1),

constraint athletics\_member\_id\_fk foreign key(member\_id) references member\_master(roll\_no) on delete cascade,

constraint athletics\_sport\_id\_fk foreign key(sport\_id) references sport(sport\_id) on delete cascade,

constraint athletics\_status check(status in('heat','finals'))

);

create table member\_sport(

member\_id number(10),

sport\_id number(10),

constraint member\_sport\_member\_id\_fk foreign key(member\_id) references member\_master(member\_id) on delete cascade,

constraint member\_sport\_sport\_id\_fk foreign key(sport\_id) references sport(sport\_id) on delete set null

);

create table score(

match\_id number(10),

team\_id number(10),

score1 number(10),

score2 number(10),

score3 number(10),

score4 number(10),

score5 number(10),

constraint score\_match\_id\_fk foreign key(match\_id) references fixture(match\_id) on delete cascade,

constraint score\_team\_id\_fk foreign key(team\_id) references team\_master(team\_id) on delete cascade

);

create table match\_point(

sport\_id number(10),

team\_id number(10),

matches\_played number(10),

matches\_won number(10),

points number(10),

nrr number(10),

draw number(10),

score\_a number(10),

score\_f number(10),

winpc number(10),

constraint match\_point\_sport\_id\_fk foreign key(sport\_id) references sport(sport\_id) on delete cascade,

constraint match\_point\_team\_id\_fk foreign key(team\_id) references team\_master(team\_id) on delete set null

);

create table overall\_point(

team\_id number(10),

points number(10),

constraint overall\_point\_team\_id\_fk foreign key(team\_id) references team\_master(team\_id) on delete set null

);