**PREPARED BY:** **GUIDED BY:**

VASU PATEL(CE-093) PROF. J. M. Pandya(Assistant Prof.)

DHRUVIT PAREKH(CE-082)

PRASHIT PATEL(CE-090)

INDEX

1. Project Overview………………………..

2. E .R. Diagram…………………………….

3. Database Schema/TABLE Definition……..

4. Implementation………………………….

5.Index and Views…………………………

6. SRS Queries…………………………………..

7. Reports……………………………………

8. Deployment Steps……………………..

9.Oracle Features Summary………………

PROJECT OVERVIEW

Online Betting System provides users who are interested in cricket and really want to watch their dream team to play in real , The FANTASY BUZZ is an online cricket betting website that allows all users to create their own teams & compete . Unlike other betting websites , our product allows users to play for each international matches & they also shows who are in leader in this thrilling competition .

**Goals:-**

-To add more entertainment towards the game of cricket.

-To show expertise on selection of complete team to other users.

**Scopes:-**

-Participation will be available for all real cricket matches.

-Updation of scores will be available only after completion of match.

**Features:-**

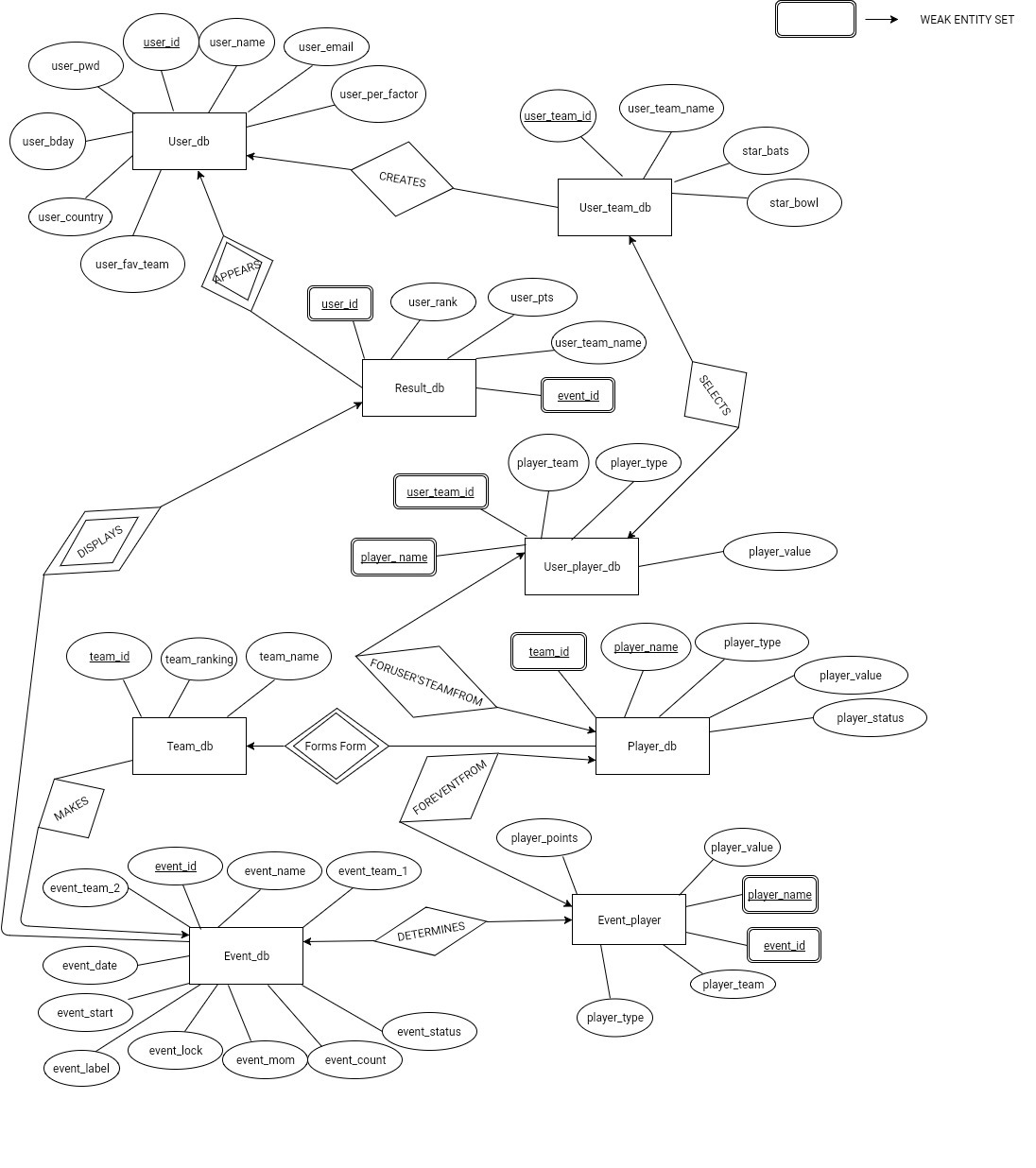
-Create your own league & compete with your friends.

-GUI would be easier to manage your team.

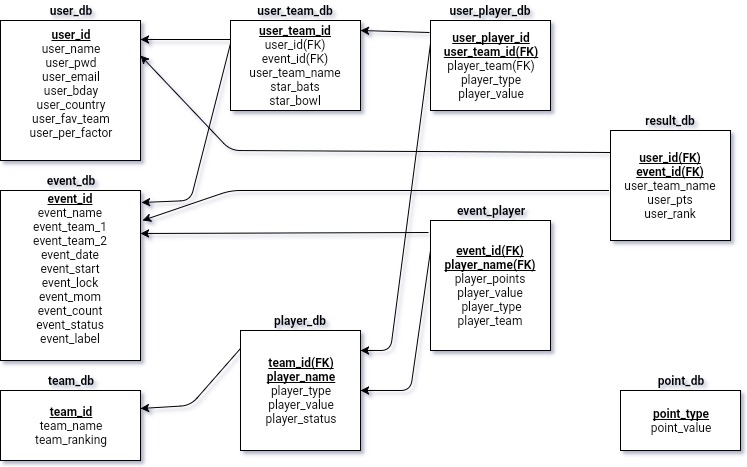
-Individual points of each player will be shown in the result.

|  |  |
| --- | --- |
| *Terms* | *Definition* |
| *Events* | *Upcoming matches between 2 teams.* |
| *League* | *All participants will take part in it and through this league winners will be decided.* |
| *Guide page* | *By which user comes to know about conditions and how points will be given to the participants.* |
| *User* | Reviewer or Author. |
| *Member* | *User who participates in the event* |

ER DIAGRAM



**SCHEMA DIAGRAM**



**IMPLEMENTATION**

**TABLE CREATION**

**CREATE USER\_DB TABLE:-**

CREATE TABLE user\_db (

user\_id INT IDENTITY(1,1) PRIMARY KEY,

user\_name VARCHAR2(50) UNIQUE NOT NULL ,

user\_pwd VARCHAR2(50) NOT NULL,

user\_email VARCHAR2(50) UNIQUE NOT NULL,

user\_bday DATE ,

user\_country VARCHAR2(50),

user\_fav\_team VARCHAR2(50),

user\_per\_factor NUMBER(4,2) default(0)

);

INSERT INTO user\_db(user\_id ,user\_name,user\_pwd,user\_email,user\_bday,user\_country,user\_fav\_team,user\_per\_factor) VALUES (0001,vasu,vasu123,vasu98@gmail.com,”06/02/1998”,”INDIA”,”INDIA”,0.0);

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **DATATYPE** | **SIZE** | **ATTRIBUTE** |
| user\_id | INT |  | PRIMARY KEY |
| user\_name | VARCHAR2 | 50 | UNIQUE |
| user\_pwd | VARCHAR2 | 50 | NOT NULL |
| user\_email | VARCHAR2 | 50 | UNIQUE |
| user\_bday | Date |  | NULL |
| user\_country | VARCHAR2 | 50 | NULL |
| user\_fav\_team | VARCHAR2 | 50 | NULL |
| user\_per\_factor | Float |  | BY DEFAULT ‘0’ |

**CREATE TEAM\_DB TABLE:-**

CREATE TABLE team\_db (

team\_id INT IDENTITY(1,1) PRIMARY KEY ,

team\_name VARCHAR2(50) UNIQUE,

team\_ranking INT UNIQUE ,

);

INSERT INTO team\_db (team\_id,team\_name,team\_ranking) VALUES (001,”INDIA”,1);

INSERT INTO team\_db (team\_id,team\_name,team\_ranking ) VALUES (002,”SOUTH AFRICA”,2);

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **DATATYPE** | **SIZE** | **ATTRIBUTE** |
| team\_id | INT |  | PRIMARY KEY |
| team\_name | VARCHAR2 | 50 | UNIQUE |
| team\_ranking | INT |  | UNIQUE |

**CREATE PLAYER\_DB TABLE:-**

CREATE player\_db(

team\_id INT ,

player\_name VARCHAR2(50),

player\_type VARCHAR2 (50),

player\_value NUMBER(2,2) ,

player\_status VARCHAR2 (50),

PRIMARY KEY ( team\_id , player\_name )

FOREIGN KEY (team\_id) REFERENCES team\_db(team\_id),

CHECK ( player\_status like ‘unavailable’ OR player\_status like ‘available’)

CHECK ( player\_value < 11.0 )

CHECK ( player\_type LIKE ‘batsman’ OR player\_type LIKE ‘bowler’ OR player\_type LIKE ‘allrounder’)

);

INSERT INTO player\_db ( team\_id , player\_name , player\_type , player\_value , player\_status) VALUES (001,”VIRAT KOHLI” , ”BATSMAN” , 10.0 , “AVAILABLE” );

INSERT INTO player\_db ( team\_id , player\_name , player\_type , player\_value , player\_status) VALUES (004,”STEVE SMITH” , ”BATSMAN” , 10.0 , “AVAILABLE” );

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **DATATYPE** | **SIZE** | **ATTRIBUTE** |
| team\_id | INT |  | FOREIGN KEY TEAM\_DB,PRIMARY KEY |
| player\_name | VARCHAR2 | 50 | PRIMARY KEY |
| player\_type | VARCHAR2 | 50 | NULL |
| player\_value | VARCHAR2 |  | NULL |
| player\_status | VARCHAR2 | 50 | NULL |

**CREATE EVENT\_DB TABLE:-**

CREATE TABLE event\_db (

event\_id INT PRIMARY KEY,

event\_name VARCHAR2(50),

event\_team1 VARCHAMAR2(50),

event\_team2 VARCHAR2(50),

event\_date date,

event\_start time\_stamp,

event\_lock time\_stamp,

event\_mom varchar2(50),

event\_count INT DEFAULT ‘0’,

event\_status VARCHAR2(50) NULL,

event\_label VARCHAR2,

FOREIGN KEY (event\_team\_id1) REFERENCES team\_db(team\_name),

CHECK ( event\_status LIKE ‘open’ OR event\_status LIKE ‘hidden’ OR event\_status LIKE ‘locked’ OR event\_status LIKE ‘closed’ )

FOREIGN KEY (event\_team\_id2) REFERENCES team\_db(team\_name),

CHECK(event\_lock > event\_start),

);

INSERT INTO event\_db (event\_id , event\_name , event\_team\_1 , event\_team\_2 , event\_date, event\_start , event\_lock , event\_count,event\_status ) VALUES (1,”indiavsnewzealand”, “INDIA”, “NEWZEALAND” , 18-11-2017 01:00, 19-11-2017 11:00 , 0 , “HIDDEN” );

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **DATATYPE** | **SIZE** | **ATTRIBUTE** |
| event\_id | INT |  | PRIMARY KEY |
| event\_label | VARCHAR2 | 50 | NULL |
| event\_team\_1 | VARCHAR2 | 50 | NULL |
| event\_team\_2 | VARCHAR2 | 50 | NULL |
| event\_date | DATE |  | NULL |
| event\_start | Time\_stamp |  | NOT NULL |
| event\_lock | Time\_stamp |  | NOT NULL |
| event\_mom | VARCHAR2 | 50 | NOT NULL |
| event\_count | INT |  | BY DEFAULT ‘0’ |
| event\_status | VARCHAR2 | 50 | NOT NULL |

**CREATE EVENT\_PLAYER\_DB TABLE:-**

CREATE TABLE event\_player\_db(

event\_id INT ,

player\_name VARCHAR2(50),

player\_points NUMBER(2,2),

player\_value NUMBER(2,2) NOT NULL,

player\_type VARCHAR2(50) ,

player\_team VARCHAR2(50),

PRIMARY KEY(event\_id,player\_name),

FOREIGN KEY (player\_id) references player\_db(player\_id),

FOREIGN KEY(player\_name) references player\_db(player\_name)

);

INSERT INTO event\_player\_db (event\_id , player\_name , player\_points , player\_values , player\_type , player\_team ) VALUES (1,”VIRATKOHLI”, 0, 10.5 , “BATSMAN” , “INDIA” );

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **DATATYPE** | **SIZE** | **ATTRIBUTE** |
| event\_id | INT |  | FOREIGN KEY EVENT\_DB |
| Player\_name | VARCHAR2 | 50 | FOREIGN KEY PLAYER\_DB |
| player\_points | NUMBER |  | BY DEFAULT ‘0’ |
| player\_value | NUMBER |  | NOT NULL |
| player\_type | VARCHAR2 | 50 |  |
| Player\_team | VARCHAR2 | 50 |  |

**CREATE USER\_TEAM\_DB TABLE:-**

CREATE TABLE user\_team\_db(

event\_id INT ,

user\_id INT ,

user\_team\_id INT PRIMARY KEY

user\_team\_name VARCHAR2(50) ,

star\_bats VARCHAR2(50),

star\_bowl VARCHAR2(50),

FOREIGN KEY (user\_id) references user\_db(user\_id),

FOREIGN KEY (event\_id) references event\_db(event\_id)

);

INSERT INTO user\_team\_db (event\_id , user\_id , user\_team\_name , user\_team\_id , star\_bats , star\_bowl ) VALUES (1,0001,”strikers11” , 0101 , “VIRAT KOHLI” , “JASPRIT BUMRAH” );

INSERT INTO user\_team\_db (event\_id , user\_id , user\_team\_name , user\_team\_id , star\_bats , star\_bowl ) VALUES (1,0001,”SUPER11” , 0105 , “ROHIT SHARMA” , “AXAR PATEL” );

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **DATATYPE** | **SIZE** | **ATTRIBUTE** |
| user\_team\_id | INT |  | PRIMARY KEY |
| event\_id | INT |  | FOREIGN KEY EVENT\_DB |
| user\_id | INT |  | FOREIGN KEY USER\_DB |
| user\_team\_name | VARCHAR2 | 50 | NOT NULL |
| star\_bats | VARCHAR2 | 50 |  |
| star\_bowl | VARCHAR2 | 50 |  |

**CREATE USER\_PLAYER TABLE:-**

CREATE TABLE user\_palyer (

user\_team\_id INT ,

player\_name VARCHAR2(50) ,

player\_team VARCHAR2(50) NOT NULL,

player\_type VARCHAR2(50) ,

player\_value NUMBER(2,2),

PRIMARY KEY (user\_team\_id,player\_name),

FOREIGN KEY (player\_name) references player\_db(player\_name),

FOREIGN KEY (user\_team\_id) references user\_team\_db(user\_team\_id),

FOREIGN KEY (player\_team) references team\_db(team\_name),

);

INSERT INTO user\_player\_db (user\_team\_id , player\_name , player\_team , player\_type , player\_value ) VALUES (0101,”VIRATKOHLI” , “INDIA” , “BATSMAN” ,”viratkohli” , 10.5 );

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **DATATYPE** | **SIZE** | **ATTRIBUTE** |
| user\_team\_id | INT |  | FOREIGN KEY USER\_TEAM\_DB |
| player\_name | VARCHAR2 | 50 | FOREIGN KEY PLAYER\_DB |
| player\_team | VARCHAR2 | 50 |  |
| player\_type | VARCHAR2 | 50 |  |
| player\_value | NUMBER |  |  |

**CREATE RESULT\_DB TABLE:-**

CREATE TABLE result\_db(

event\_id INT ,

user\_id INT ,

user\_team\_name VARCHAR2(50),

user\_pts float(53) default(0) ,

user\_rank INT default(1),

PRIMARY KEY ( event\_id , user\_id )

FOREIGN KEY (event\_id) REFERENCES event\_db(event\_id),

FOREIGN KEY (user\_id) REFERENCES user\_db (user\_id)

);

INSERT INTO result\_db (event\_id , user\_id , user\_team \_name , user\_pts , user\_rank ) VALUES (1,0001,”strikers11” , 115 , 7 );

INSERT INTO result\_db (event\_id , user\_id , user\_team \_name , user\_pts , user\_rank ) VALUES (2,0001,”super11” , 225 , 1,10009 );

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **DATATYPE** | **SIZE** | **ATTRIBUTE** |
| user\_id | INT |  | FOREIGN KEY USER\_DB |
| event\_id | INT |  | FOREIGN KEY EVENT\_DB |
| user\_team\_name | VARCHAR2 | 50 | NULL |
| user\_pts | NUMBER |  | NULL |
| user\_rank | INT |  | NULL |

**CREATE POINT\_DB TABLE:-**

CREATE TABLE point\_db (

Point\_type varchar2(50) PRIMARY KEY,

Point\_value NUMBER(2)

);

INSERT INTO point\_db (point\_type,point\_value) VALUES (”RUNS”,1);

INSERT INTO point\_db (point\_type,point\_value) VALUES (”DUCK”,-5);

INSERT INTO point\_db (point\_type,point\_value) VALUES (”FOUR”, 5);

INSERT INTO point\_db (point\_type,point\_value) VALUES (”SIX”,10);

INSERT INTO point\_db (point\_type,point\_value) VALUES (”FIFTY”,20);

INSERT INTO point\_db (point\_type,point\_value) VALUES (”CENTURY”,30);

INSERT INTO point\_db (point\_type,point\_value) VALUES (”WICKET”,20);

INSERT INTO point\_db (point\_type,point\_value) VALUES (”MAIDEN”,5);

INSERT INTO point\_db (point\_type,point\_value) VALUES (”RUNS GIVEN”,-0.25);

INSERT INTO point\_db (point\_type,point\_value) VALUES (”FIVE WICKET ”, 30 );

INSERT INTO point\_db (point\_type,point\_value) VALUES (”MOM ”, 50 );

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **DATATYPE** | **SIZE** | **ATTRIBUTE** |
| Point\_type | VARCHAR2 | 50 | PRIMARY KEY |
| Point\_value | NUMBER | 2 |  |

**Indexes And Views**

Index1

# Index on event\_player table on event\_id and player\_team.

**Create index eve\_team\_player on event\_player(event\_id,player\_team);**

**Index 2**

# Index on user\_team\_db table on event\_id.

**Create index eve\_user on user\_team\_db(event\_id);**

Index 3

# Index on user\_player\_db on user\_team\_id.

**Create index user\_team on user\_player\_db(user\_team\_id);**

**View1**

# View to see players available for selection for current event.

**Create view available\_players as select \* from player\_db where player\_status='available' and team\_id in (select team\_id from team\_db where team\_name in (select event\_team1 from event\_db where event\_id=1) OR team\_name in (select event\_team2 from event\_db where event\_id=1));**

View2

# View for user to see his/her team after creation.

**Create view my\_team as select \* from user\_player\_db where user\_team\_id in (select user\_team\_id from user\_team\_db where user\_id=0001 and event\_id=1);**

**SRS Queries**

**4.1***:****System allows admin to manage events.***

**i/p:-event\_info**

**o/p:-updated\_info**

**des:-System takes information about teams,time etc regarding the event and allows admin to manage that event.**

*4.1.1 System allows admin to CREATE the event.*

i/p:-event\_info

o/p:-updated\_event\_info

des:-System CHECKs for valid teams in master\_TABLE & CREATEs event & make event visible to users.

Query:- INSERT INTO event\_db (event\_id,event\_team1,event\_team2,event\_date,event\_start,event\_lock,event\_count,event\_status,event\_label) VALUES (1,'INDIA','NEW ZEALAND','08/11/2017','08/11/2017 01:00:00','08/11/2017 7:00:00' , 0 , 'locked','3rd T20');

*4.1.2 System allows admin to edit event.*

i/p:- event\_info

o/p:- updated\_event\_info

des:-System updates the event info as specified by the user.

Query:-update event\_id set *[updated\_attribute]*=*”new value”* where event\_id=1;

Eg.-

To update event\_date: update event\_db set event\_date=10-11-2017 where event\_id=1;

*4.1.3 System allows admin to delete event.*

i/p:-event\_info

o/p:-updated\_event\_info

des:-System deletes the event from database as and when required by the admin and makes it invisible for the users.

Query:- delete \* from event\_db where event\_id=1;

## *4.2:-System provides selection of players from 2 teams playing the match with max. 6 players from 1 team.*

**i/p:-Players from participating team**

**o/p:-Final CREATEd team**

**dec:-System provides all the players list from 2 participating teams in the event and allows user to CREATE his team from it.**

*4.2.1 System provides event\_TABLE from master\_TABLE of players for 2 teams.*

i/p:-master\_TABLE of players

o/p:-event\_TABLE

des:-System will CREATE team TABLE for the event from master\_TABLE of players & makes visible to users.

Query: select p.\*,t.team\_name from player\_db p,team\_db t where p.player\_status='available' and t.team\_id in (select team\_id from team\_db where team\_name in (select event\_team1 from event\_db where event\_id=1) OR team\_name in (select event\_team2 from event\_db where event\_id=1));

*4.2.2 System will validate the team on creation by user.*

i/p:-event\_TABLE

o/p:-(valid)team\_TABLE

des:-System will count for total 11 players & max. 6 from each & validates the team.

Query:-select count(\*) from user\_player\_db where user\_team\_id in (select user\_team\_id from user\_team\_db where user\_id=0001 and event\_id=1);

select count(\*) from user\_player\_db where user\_team\_id in (select user\_team\_id from user\_team\_db where event\_id=1 and user\_id=1) group by(player\_team);

*4.2.3 System allows user to CREATE 1 star batsman & 1 star bowler.*

i/p:-(valid)team\_TABLE

o/p:-star\_batsman\_flag=true and star\_bowler\_flag=true

des:-System will mark the star batsman & star bowler enter by the user.

Query:-update user\_team\_db set star\_bats=’Virat Kohli’ and star\_bowl=*’*JASPRIT BUMRAH*’* where user\_id=0001;

*4.2.4 System displays the final team to the user.*

i/p:-(valid)team\_TABLE

o/p:-team\_visible\_flag=true

des:-System CHECKs team \_TABLE if it is valid,it displays the team otherwise error message.

Query:- select \* from user\_player\_db where user\_team\_id in (select user\_team\_id from user\_team\_db where user\_id=0001 and event\_id=1);

*4.2.5 System provides option to edit the team.*

i/p:-(valid)team\_TABLE

o/p:-changed (valid)team\_TABLE

des:-System changes the players edited by the user & CHECKs for valid team.

Query:-update user\_player\_db set player\_name=’ROHIT SHARMA’,

player\_team=’INDIA’,player\_type=’batsman’*,*  where user\_team\_id in (select user\_team\_id from user\_team\_db where user\_id=0001 and event\_id=1);

*4.2.6 System lock the final valid team\_TABLE.*

i/p:-(valid)team\_TABLE

o/p:-team\_lock\_flag=true

des:-System compares system time & event time if same,it locks the team.

Query:- update event\_db set event\_status=’locked’ where event\_id=1;

## *4.3:-PoINTs will be given to players according to his runs and/or wickets & there will be no poINTs given for abandoned match.*

**i/p(valid)team\_TABLE**

***o/p:-result\_TABLE***

***des:-System gives poINTs to the submitted team after completion of match and gives the final result.***

*4.3.1 System allocates poINTs to individual players from each team.*

i/p:-(valid)team\_TABLE

o/p:-total poINTs

des:-System will allocate poINTs to each player & add each of them to calculate total poINTs per each team.

*4.3.1.1 System allocates poINT to batsman.*

i/p:-(valid)team\_TABLE

o/p:-poINTs of batsman

des:-System allocates -5 for duck,1 poINT for each run,5 poINTs for each boundary,10 poINTs for each six,20 poINTs for half century & 30 poINTs for century to each batsman from team\_TABLE.

Query:-

**IF BATSMAN OUT FOR DUCK** : UPDATE RESULT\_DB SET USER\_PTS=(SELECT POINT\_VALUE FROM POINT\_DB WHERE POINT\_TYPE=’DUCK’) WHERE USER\_ID=0001 AND EVENT\_ID=1;

**IF BATSMAN TAKES ONE RUN** : UPDATE RESULT\_DB SET USER\_PTS=(SELECT POINT\_VALUE FROM POINT\_DB WHERE POINT\_TYPE=’RUNS’) WHERE USER\_ID=0001 AND EVENT\_ID=1;

**IF BATSMAN HIT BOUNDARY** : UPDATE RESULT\_DB SET USER\_PTS=(SELECT POINT\_VALUE FROM POINT\_DB WHERE POINT\_TYPE=’FOUR’) WHERE USER\_ID=0001 AND EVENT\_ID=1;

**IF BATSMAN HIT SIX** : UPDATE RESULT\_DB SET USER\_PTS=(SELECT POINT\_VALUE FROM POINT\_DB WHERE POINT\_TYPE=’SIX’) WHERE USER\_ID=0001 AND EVENT\_ID=1;

**IF BATSMAN SCORES HALFCENTURY** : UPDATE RESULT\_DB SET USER\_PTS=(SELECT POINT\_VALUE FROM POINT\_DB WHERE POINT\_TYPE=’FIFTY’) WHERE USER\_ID=0001 AND EVENT\_ID=1;

**IF BATSMAN SCORES CENTURY**  : UPDATE RESULT\_DB SET USER\_PTS=(SELECT POINT\_VALUE FROM POINT\_DB WHERE POINT\_TYPE=’CENTURY’) WHERE USER\_ID=0001 AND EVENT\_ID=1;

*4.3.1.2 System allocates poINT to bowler.*

i/p:-(valid)team\_TABLE

o/p:-poINTs of bowler

des:-System allocates 20 poINTs for each wicket,-0.25 for each run conceded,5 poINTs for maiden over & 30 poINTs for 5 wickets to each bowler from team\_TABLE.

Query:-

**IF BOWLER TAKES WICKET** : UPDATE RESULT\_DB SET USER\_PTS=(SELECT POINT\_VALUE FROM POINT\_DB WHERE POINT\_TYPE=’WICKET’) WHERE USER\_ID=0001 AND EVENT\_ID=1;

**IF BOWLER CONCEDES ONE RUN** : UPDATE RESULT\_DB SET USER\_PTS=(SELECT POINT\_VALUE FROM POINT\_DB WHERE POINT\_TYPE=’RUN GIVEN’) WHERE USER\_ID=0001 AND EVENT\_ID=1;

**IF BOWLER BOWLED MAIDEN OVER** : UPDATE RESULT\_DB SET USER\_PTS=(SELECT POINT\_VALUE FROM POINT\_DB WHERE POINT\_TYPE=’MAIDEN’) WHERE USER\_ID=0001 AND EVENT\_ID=1;

**IF BOWLER TAKES FIVE WICKETS** : UPDATE RESULT\_DB SET USER\_PTS=(SELECT POINT\_VALUE FROM POINT\_DB WHERE POINT\_TYPE=’FIVE WICKETS’) WHERE USER\_ID=0001 AND EVENT\_ID=1; .

*4.3.1.3 System allocates extra poINTs for man of the match award.*

i/p:-(valid)team\_TABLE

o/p:-poINTs to mom player

des:-System CHECKs for mom player to the team\_TABLE & if present , allocates 25 poINTs.

Query:-

UPDATE RESULT\_DB SET USER\_PTS=(SELECT POINT\_VALUE FROM POINT\_DB WHERE POINT\_TYPE=’MOM’) WHERE USER\_ID=0001 AND EVENT\_ID=1;

*4.3.1.4 System allocates bonus poINTs for star batsman & star bowler.*

i/p:-(valid)team\_TABLE

o/p:-poINTs to star bowler & star batsman

des:-System doubles the existing poINTs for star bowler & star batsman.

Query:- UPDATE RESULT\_DB SET USER\_PTS=(select user\_pts from result\_db where USER\_ID=1 AND EVENT\_ID=1)+ (SELECT PLAYER\_POINTS FROM EVENT\_PLAYER\_DB WHERE PLAYER\_NAME IN (SELECT STAR\_BATS FROM USER\_TEAM\_DB WHERE USER\_ID=1 AND EVENT\_ID=1)) WHERE USER\_ID=0001 AND EVENT\_ID=1;

UPDATE RESULT\_DB SET USER\_PTS=(select user\_pts from result\_db where USER\_ID=0001 AND EVENT\_ID=1)+ (SELECT PLAYER\_POINTS FROM EVENT\_PLAYER\_DB WHERE PLAYER\_NAME IN (SELECT STAR\_BOWL FROM USER\_TEAM\_DB WHERE USER\_ID=0001 AND EVENT\_ID=1)) WHERE USER\_ID=0001 AND EVENT\_ID=1;

*4.3.1.5 System calculates total score.*

i/p:-(valid)team\_TABLE

o/p:-total points

des:-System adds the poINTs of each player from the team\_TABLE.

Query:- select user\_pts from result\_db where user\_id=0001 and event\_id=1;

*4.3.2 System CREATEs list of all participants.*

i/p:-team poINTs

o/p:-result\_TABLE

des:-System CREATEs list of all the participated teams and sorts in descending order according to team poINTs.

Query:-

SELECT r.\* FROM RESULT\_DB r left outer join EVENT\_DB e on r.EVENT\_ID=e.event\_id;

*4.3.3 System displays winning teams.*

i/p:-result\_TABLE

o/p:-rank\_TABLE

des:-System gives ranks to top 3 teams and displays the final results.

Query:-

SELECT EVENT\_ID,USER\_ID,USER\_TEAM\_NAME , USER\_PTS, RANK() OVER (ORDER BY USER\_PTS DESC) AS RANK FROM RESULT\_DB where EVENT\_ID=1;

Reports

REPORT 1

# See User Teams after event is locked.

* **Master Report**: See events when locked.

Select \* from event\_db where event\_status=’locked’;

* **Sub Report 1** :See which users have created teams in the respective events by clicking on event row.

Select \* from user\_team\_db where event\_id=:EVENT\_ID;

* **Sub Report 2**:See the selected users team on clicking on the user’s row.

Select \* from user\_player\_db where user\_team\_id=:USER\_TEAM\_ID and event\_id=:EVENT\_ID;

REPORT 2

# See Results on completion of event.

* **Master Report:** See events when closed.

Select \* from event\_db where event\_status=’closed’;

* **Sub Report 1** :See which results for the respective events by clicking on event row.

Select \* from result\_db where event\_id=:EVENT\_ID;

**Deployment steps :--**

1. Create user named OnlineBettingProject.

2. Grant all permission to user OnlineBettingProject.

3. Drop all tables if already exists in given order result\_db , user\_player\_db , user\_team\_db , event\_player\_db , event\_db, player\_db , team\_db , user\_db.

6. Create all tables in given order user\_db , team\_db , player\_db , event\_db , event\_player\_db, user\_team\_db , user\_player\_db , result\_db.

5. Drop all sequences if already exists in given order result\_id , user\_player\_id , user\_team\_id , table\_id , event\_id, player\_id , team\_id , user\_id .

6. Create all sequences user\_id , team\_id, player\_id , event\_id , table\_id , user\_team\_id , user\_player\_id , result\_id .

7. Insert initial data in given order user\_db , team\_db , player\_db , event\_db , event\_player\_db, user\_team\_db , user\_player\_db , result\_db.

8. Commit all changes.

Oracle Features Summary

|  |  |
| --- | --- |
| Features | Used in Project |
| Composite Attributes | No |
| Weak Entity | Yes |
| Strong Entity | Yes |
| Specialization & Generalization | No |
| Index | Yes |
| View | Yes |
| Triggers | No |
| Report | Yes |
| Sub Queries | Yes |
| Grouping | Yes |
| Joins | Yes |
| Functions(rank) | Yes |
| Constraints | Yes |
| Reg\_Exp Like | No |