**By,**

**1)ANUSHREE SUGUR VEERAIAH**

**WIU ID:915156931**

**Oracle account: asv104**

**2)SHREE HARSHITHA ALUGUVELLY**

**WIU ID:916618583**

**Oracle account: sha104**

**3)YASHASWINI MANDYA PANCHALINGHIA**

**WIU ID:916151998**

**Oracle account: ym104**

**THE DATABASE CREATED IN THE ACCOUNT ‘ym104’**

1. **DESIGN PHASE:**

a) **Describe the enterprise:**

i). **Introduction and brief enterprise description**.

This project defines over all requirements for international sports tournament system. And even it provides the information regarding a sports tournament and also the necessary details about various sports in a tournament.

ii). **What functions should the system perform?**

The system must be easily able to provide information to the user about the tournament.

iii). **Who are the end users? Remember that the DBA is NOT an end user**.

Here the SPONSORS and PLAYERS are the end users. They can search by the player name, sport or the country the player is representing.

iv). **How will data Obsolescence be handled?**

When a event is held a large amount of data collected for players their records, life time events. When a player gets retired and no longer plays for future events, database should be updated for the present records to save memory and avoid confusion. So all the past records are moved from the database.

v). **Where did you get the idea for this project? Did you make it up, get it from work, or find it in a book? Please site your sources. The idea may NOT be something solved in a book, nor may it be a simple add-on to an existing database.**

The topic we chose is based upon our interest in sports. This came out of discussion with instructor and team mates.

b) **Entity Relationship Design:**

* **Entity Description**

1)Player (P\_ID, P\_name, P\_sex, P\_age)

The Player entity has the information P\_ID, P\_name, P\_sex, P\_age attributes

->PID is the primary key

* P\_ID-This is the unique id for each player
* P\_name-This column gives the name of the player
* P\_age-This column gives the age of the player
* P\_sex-This column gives the sex of the player

2)Country(C\_ID,C\_name)

The country entity has the information C\_ID,C\_name

->C\_ID is the primary key

* C\_ID -This is the unique id for a country
* C\_name-This column gives the country name.

3)Stadium (S\_ID,S\_name,City,Stadium type)

The stadium entity has the information S\_ID,S\_name,City,Stadium type

->S\_ID- This is the primary key

* S\_ID-This gives the stadium ID information
* S\_name-This column gives the name of the stadium
* City-This column gives the name of the city in which the stadium belongs
* Stadium type-This column gives the type of the stadium

4)Sponsor(Sp\_ID, company name)

The sponsor entity has the information Sp\_ID, company name

->Sp\_ID – This is the primary key

* Sp\_ID-This gives the sponsor ID information and it is unique
* Company name-This column gives the information of a company who is sponsoring the sport

5)Sport (St\_ID,St\_Name,St\_Type)

The sport entity has the information St\_ID,St\_Name,St\_Type

->St\_ID- this is the primary key.

* St\_ID- This is the unique ID for the sport
* St\_Name- This column gives the sportname
* St\_Type- This column gives the information about the type of the sport

6)Tournament(T\_name, T\_Date, T\_Venue,T\_ID)

The tournament entity has the information T\_name, T\_Date, T\_Venue

* T\_name- This gives the name of the tournament.
* T\_Date- This column gives the information about on which date the tournament is held.
* T\_Venue- This column gives the venue details where the tournament is held.
* T\_ID-This column gives the information about the tournament ID.
* **LIST OF RELATIONSHIPS**

1.**Player/country** 🡪 Many – One relation.

This is a many to one relation as there will be many players from one country. So a country has many players.

2.**Player /sport** 🡪 Many - one relation.

This is a many to one relation as there will be many players participating for one sport. So a sport will have many players participating.

3.**Stadium/sport** 🡪 One- many relation.

This becomes a many to many relation, many sports in a tournament `are conducted in a stadium. So a stadium will accommodate many sports.

4. **Tournament/sport** 🡪 One-many relation.

This is a one to many relation as many sports are held during one tournament. So a tournament can have many different types of sports.

5.**Sport/sponsor** 🡪 Many –many relation.

This is a many to many relation as a many sponsors can sponsor various sports. So many sponsors can sponsor many sports.

6. **Stadium/Tournament** 🡪 One- one relation.

This is a one- one relation as the tournament has a particular venue i.e., one stadium. So one stadium can have one tournament to take place.

* **E-R Diagram:**

Stadium

Player

m m m

Represent

Played in

1

Will have

Country

Sponsor

m m

Includes

Can Have

1 m

Tournament

Sport

1 m

1 m

Have

**C) CONCEPTUAL LEVEL:**

All the below listed tables are in 3NF.

* **Table :PLAYER**

Player (**P\_ID** NUMBER(5),

P\_NAME CHAR(20),

P\_SEX CHAR(1),

P\_AGE NUMBER(2));

1. **Primary key**: P\_ID
2. **Column domains:**

P\_ID(number) , P\_NAME(char),P\_SEX(char),P\_AGE(number)

1. **Domain Integrity checks**.

P\_ID : This is a unique number assigned to each player. This is primary key.

P\_NAME: This a character and not null.

P\_SEX: This is a character. This is not null.

P\_AGE: This is a number to represent the date of birth of a player. This is not equal to zero.

1. **Functional Dependencies:**

**P\_ID** 🡪 P\_NAME,P\_SEX,P\_AGE

* **Table: COUNTRY**

Country (**C\_ID** NUMBER(5),

C\_NAME CHAR(10),

P\_ID NUMBER(5));

1. **Primary key:** C\_ID
2. **Foreign key:** P\_ID
3. **Column domain:**

C\_ID(Number) ,C\_NAME(char)

1. **Domain Integrity checks:**

C\_ID : This a primary key and this is a unique number.

C\_NAME : This is a character not null.

P\_ID: This is a number.

1. **Functional Dependencies:**

**C\_ID** 🡪 C\_ID, C\_NAME, P\_ID

* **Table: STADIUM**

Stadium (**S\_ID** NUMBER(5),

S\_NAME CHAR(10),

CITY CHAR(10),

ST\_ID NUMBER(5) );

1. **Primary key:** S\_ID
2. **Foreign key:** ST\_ID
3. **Column domain:** S\_ID(Number), S\_NAME(char),CITY(char),ST\_ID(number)
4. **Domain integrity checks:**

S\_ID: This is a unique number given to represent a stadium. This is not null.

S\_NAME: This is a character.

1. **Functional dependencies:**

**S\_ID** 🡪 S\_NAME, CITY, ST\_ID

* **Table: SPONSOR**

Sponsor (**SP\_ID** NUMBER(5)

Company name CHAR(10),

ST\_ID NUMBER(5));

1. **Primary key:** SP\_ID
2. **Foreign key:** ST\_ID
3. **Column domain:** SP\_ID( number), ST\_ID (number)
4. **Domain integrity check:**

SP\_ID: This is a unique number given to sponsor. This is not null.

Company name: This is the name of company of the sponsors.

ST\_ID: This is a number

1. **Functional dependencies:**

**SP\_ID** 🡪 SP\_ID, COMPANY\_NAME, ST\_ID.

* **Table: SPORT**

Sport (**ST\_ID** NUMBER(5),

ST\_name CHAR(10),

ST\_type CHAR(5),

S\_ID NUMBER(5),

SP\_ID NUMBER(5),

P\_ID NUMBER(5),

T\_ID NUMBER(5));

1. **Primary key:** ST\_ID
2. **Foreign key:** S\_ID,SP\_ID,P\_ID,T\_ID
3. **Column Domain:** ST**\_**ID(number),SP\_ID(number),ST\_type(char),

S\_ID(number),SP\_ID(number),P\_ID(number),T\_ID(number).

1. **Domain integrity check:**

ST\_ID: Every sport is given a unique number for representation. This is not null.

ST\_NAME: The sport name is represented in characters. This is not null.

ST\_TYPE: The sport type is represented in characters.

S\_ID:This is a number

SP\_ID:This is a number

P\_ID: This is a number

T\_ID:This is a number

1. **Functional Dependencies:**

**ST\_ID** 🡪 ST\_ID, ST\_NAME, ST\_TYPE, SP\_ID, S\_ID, P\_ID,T\_ID

* **Table: TOURNAMENT**

Tournament( **T\_ID** NUMBER(5),

T\_NAME CHAR(10),

T\_DATE NUMBER(8),

T\_VENUE CHAR(5),

ST\_ID NUMBER(5),

C\_ID NUMBER(5));

1. **Foreign key:** ST\_ID, C\_ID
2. **Column domain:**

T\_NAME(char),T\_DATE(number),T\_VENUE(char),T\_ID(number)

1. **Domain integrity check:**

T\_NAME: This is a character and its unique. This is not null.

T\_DATE: This is a number.

T\_VENUE: This is a character

ST\_ID: This is a number  
C\_ID: This is a number

1. **Functional dependencies:**

None.

* **Table: CAN BE PLAYED**

Can be played (S\_ID NUMBER(5),

ST\_ID NUMBER(5));

1. **Foreign key:** S\_ID,ST\_ID
2. **Column domain:** S\_ID (number), ST\_ID (number)
3. **Domain integrity check:**

S\_ID : This is a number

ST\_ID: This is a number

1. **Functional dependencies:**

None.

* **Table: CAN HAVE**

Can have (SP\_ID NUMBER(5),

ST\_ID NUMBER(5));

1. **Foreign key:** SP\_ID, ST\_ID
2. **Column domain:** SP\_ID (number), ST\_ID( number)
3. **Domain integrity check:**

SP\_ID: This is a number

ST\_ID: This is a number

1. **Functional dependencies:**

None

**D) External view:**

|  |  |  |
| --- | --- | --- |
| **Tables** | **Player** | **Sponsor** |
| PLAYERS | Select | Select  Update  Insert  Delete |
| COUNTRY | Select | Select  Update  Insert  Delete |
| STADIUM | Select | Select  Update  Insert  Delete |
| SPONSORER | Select | Select  Update  Insert  Delete |
| SPORT | Select | Select  Update  Insert  Delete |
| TOURNAMENT | Select | Select  Update  Insert  Delete |
| CAN BE PLAYED | N/A | Select  Update  Insert  Delete |
| CAN HAVE | N/A | Select  Update  Insert  Delete |

**E) INTERNAL VIEW**

**1)** Retrieving sport id and sport type for sport id which is greater than 222

SELECT ST\_ID,ST\_TYPE

FROM SPORT\_IOT

WHERE ST\_ID>222;

Clustered B-tree has been applied on the table SPORT\_IOT

Tables used: SPORT\_IOT

Column retrieved: sport id and sport type

2) Retrieving s\_name and city for the stadium id 101

SELECT S\_NAME,CITY //hashing

FROM STADIUM

WHERE S\_ID=101;

Hashing is used on S\_ID column.

Tables used: Stadium

Column retrieved: Stadium name and city

3) Retrieving player name and age whose sex is F.

SELECT P\_NAME,P\_AGE

FROM PLAYER,COUNTRY

WHERE P\_SEX='F' //searching

AND PLAYER.P\_ID=COUNTRY.P\_ID;

Retrieving player name and age from player, country whose player sex is f by considering the player id from both the tables.

**4)** Retrieving the X- age group of player by using DECODE function.

SELECT P\_NAME,P\_ID,DECODE(FLOOR(P\_AGE/10),0,'FRESHMAN',1,'FRESHMAN',2,'JUNIOR',3,'SENIOR',4,'SUPERSENIOR')"X"

FROM PLAYER;

When the age of a player is divided by 10,gives 0,1 player is considered as FRESHMAN, if it gives 2

then the player is JUNIOR, 3 then SENIOR, 4 player is considered to be a SUPERSENIOR. Retrieving

playername and id from player table using decode function.

**5)** Retrieving the player name, country name, sports name, tournament name of the sport.

SELECT P\_NAME,C\_NAME,ST\_NAME,T\_NAME

FROM PLAYER,COUNTRY,SPORT\_IOT,TOURNAMENT

WHERE PLAYER.P\_ID=COUNTRY.P\_ID

AND PLAYER.P\_ID=SPORT\_IOT.P\_ID

AND SPORT\_IOT.ST\_ID=TOURNAMENT.ST\_ID;

Retrieved the player name,country name ,sport name and the tournament name from player, country, sport

and tournament tables by joining the tables of player and country by their p\_id, player and sport by their

p\_id and sport and tournament table by their sport id(st\_id).

6) Retrieving sport name, sport type from sponsor table using JOIN.

SELECT ST\_NAME,ST\_TYPE

FROM SPONSOR

JOIN SPORT\_IOT

ON SPONSOR.SP\_ID=SPORT\_IOT.SP\_ID;

Here sport name and sport type are retrieved from sponsor table by joining sport\_iot table by the SP\_ID.

Tables used: Sponsor, sport

Columns Retrieved: Sport id and sport type.

7) GROUP BY EXPRESSION Retrieving the tournament date and count of the teams.

SELECT T\_DATE,COUNT(\*) AS NUMOFTEAMS //Aliasing

FROM TOURNAMENT

WHERE T\_DATE IS NOT NULL

GROUP BY T\_DATE; //GROUP BY

Retrieving the date and the team count from tournament by using GROUP BY function on date of the

tournament. Here the team sport type is given a count function by aliasing it as number of teams.

8) Retrieving the sport name and the number of players participating in each sport.

SELECT ST\_NAME,COUNT(P\_ID) //COUNT

FROM SPORT\_IOT

GROUP BY ST\_NAME; //GROUP BY

Here sport name and count of the number of players is retrieved from sport\_iot by using GROUP BY

function on sport name.

**f) DATA DICTIONARY**

1. **TABLE: PLAYER**

|  |  |  |
| --- | --- | --- |
| **NAME** | **NULL?** | **DATA TYPE** |
| P\_ID | NOT NULL | NUMBER(5) |
| P\_NAME |  | CHAR(20) |
| P\_SEX |  | CHAR(1) |
| P\_AGE |  | NUMBER(2) |

**PRIMARY KEY:** P\_ID

**PRIVILIEGES:**

SPONSORS, PLAYERS

**INDEXES:**

Secondary B-tree on P\_ID

CREATE TABLE PLAYER(P\_ID NUMBER(5),

P\_NAME CHAR(20),

P\_SEX CHAR(1),

P\_AGE NUMBER(2));

CREATE INDEX P\_INDEX

ON PLAYER(P\_ID,P\_NAME);

1. **TABLE: COUNTRY**

|  |  |  |
| --- | --- | --- |
| **NAME** | **NULL?** | **DATA TYPE** |
| C\_ID | NOT NULL | NUMBER(5) |
| C\_NAME |  | CHAR(10) |
| P\_ID |  | NUMBER(5) |

**PRIMARY KEY:** C\_ID

**FOREIGN KEY:** P\_ID

**PRIVILEGES:**

SPONSORS, PLAYERS

1. **TABLE: STADIUM**

|  |  |  |
| --- | --- | --- |
| **NAME** | **NULL?** | **DATA TYPE** |
| S\_ID | NOT NULL | NUMBER(5) |
| S\_NAME |  | CHAR(10) |
| CITY |  | CHAR(10) |
| ST\_ID |  | NUM(10) |

**PRIMARY KEY:** S\_ID

**FOREIGN KEY:** ST\_ID

**PRIVILEGES:**

SPONSORS, PLAYERS

**HASH TABLE**:

CREATE TABLE STADIUM(S\_ID NUMBER(5),

S\_NAME CHAR(10),

CITY CHAR(10),

ST\_ID NUMBER(5))

CLUSTER STAD\_HASH(S\_NAME);

Hashing is applied on stadium name from stadium table by creating a cluster.

CREATE CLUSTER STAD\_HASH(S\_NAME CHAR(10))

SIZE 512 SINGLE TABLE HASHKEYS 2000;

1. **TABLE: SPONSOR**

|  |  |  |
| --- | --- | --- |
| **NAME** | **NULL?** | **DATA TYPE** |
| SP\_ID | NOT NULL | NUMBER(5) |
| COMPANY\_NAME |  | CHAR(10) |
| ST\_ID | NOT NULL | NUMBER(5) |

**PRIMARY KEY**: SP\_ID

**FOREIGN KEY**: ST\_ID

**PRIVILEGES:** SPONSORS, PLAYERS

1. **TABLE: SPORT**

|  |  |  |
| --- | --- | --- |
| **NAME** | **NULL?** | **DATA TYPE** |
| ST\_ID | NOT NULL | NUMBER(5) |
| ST\_NAME | NOT NULL | CHAR(10) |
| ST\_TYPE |  | CHAR(10) |
| S\_ID |  | NUMBER(5) |
| SP\_ID |  | NUMBER(5) |
| P\_ID |  | NUMBER(5) |
| T\_ID |  | NUMBER(5) |

**PRIMARY KEY:** ST\_ID

**FOREIGN KEY:** S\_ID, SP\_ID, P\_ID, T\_ID

**INDEXES:**

Clustered B-tree is implemented.

CREATE TABLE SPORT\_IOT(ST\_ID NUMBER(5) PRIMARY KEY,

ST\_NAME CHAR(10)NOT NULL,

ST\_TYPE CHAR(5),

S\_ID NUMBER(5),

SP\_ID NUMBER(5),

P\_ID NUMBER(5))ORGANIZATION INDEX;

**PRIVILEGES:**

SPONSORS,PLAYERS

1. **TABLE: TOURNAMENT**

|  |  |  |
| --- | --- | --- |
| **NAME** | **NULL?** | **DATA TYPE** |
| T\_ID | NOT NULL | NUMBER(5) |
| T\_NAME | NOT NULL | CHAR(10) |
| T\_DATE |  | DATE |
| T\_VENUE |  | CHAR(10) |
| ST\_ID |  | NUMBER(5) |

**PRIMARY KEY**: T\_ID

**FOREIGN KEY**: ST\_ID

**PRIVILEGES:** SPONSORS,PLAYERS

1. **TABLE :CAN BE PLAYED**

|  |  |  |
| --- | --- | --- |
| **NAME** | **NULL?** | **DATA TYPE** |
| S\_ID | NOT NULL | NUMBER(5) |
| ST\_ID | NOT NULL | NUMBER(5) |

1. **TABLE :CAN HAVE**

|  |  |  |
| --- | --- | --- |
| **NAME** | **NULL?** | **DATA TYPE** |
| SP\_ID | NOT NULL | NUMBER(5) |
| ST\_ID | NOT NULL | NUMBER(5) |

**VIEWS**

**CREATE VIEW PLAY** AS ( SELECT P\_NAME,C\_NAME,ST\_NAME,T\_NAME

FROM PLAYER,COUNTRY,SPORT\_IOT,TOURNAMENT

WHERE PLAYER.P\_ID=COUNTRY.P\_ID

AND PLAYER.P\_ID=SPORT\_IOT.P\_ID

AND SPORT\_IOT.ST\_ID=TOURNAMENT.ST\_ID);

From the above view, Joining player,country, sport and tournament

Retrieved column: player name,country name,sport name and tournament name.

**CREATE VIEW QUERY2** AS SELECT P\_NAME,P\_ID,DECODE(FLOOR(P\_AGE/10),0,'FRESHMAN',1,'FRESHMAN',2,'JUNIOR',3,'SENIOR',4,'SUPERSINIOR')"X"

FROM PLAYER;

When the age of a player is divided by 10,gives 0,1 player is considered as FRESHMAN, if it gives 2 then the player is JUNIOR, 3 then SENIOR, 4 player is considered to be a SUPERSENIOR. Retrieving player name and id from player table using decode function.

**CREATE VIEW QUERY3** AS SELECT ST\_NAME,ST\_TYPE

FROM SPONSOR

JOIN SPORT\_IOT //JOIN

ON SPONSOR.SP\_ID=SPORT\_IOT.SP\_ID;

From the above view,

Joining sport and sponsor

Retrieving column: sport name and sport type

**CREATE VIEW QUERY4** AS SELECT T\_DATE,COUNT(\*) AS NUMOFTEAMS //COUNT

FROM TOURNAMENT

WHERE T\_DATE IS NOT NULL

GROUP BY T\_DATE; //GROUP BY

From the above view,

Retrieved columns: date, specified sport type

**CREATE VIEW QUERY5** AS SELECT S\_NAME,CITY

FROM STADIUM

WHERE S\_ID=101;

From the above view,

Retrieved columns: stadium name,city

**CREATE VIEW QUERY6** AS SELECT ST\_NAME,COUNT(P\_ID)AS NUMOFPLAYERS //COUNT

FROM SPORT\_IOT

GROUP BY ST\_NAME; //GROUP BY

From the above view,

Retrieved columns: Sport name and player id by aliasing it as number of players.

**CREATE VIEW QUERY7** AS SELECT ST\_ID,ST\_TYPE

FROM SPORT\_IOT

WHERE ST\_ID>222; //RANGE

From the above view,

Retrieved columns: Sport id and sport type.

**CREATE VIEW QUERY8** AS SELECT P\_NAME,P\_AGE

FROM PLAYER,COUNTRY

WHERE P\_SEX='F'

AND PLAYER.P\_ID=COUNTRY.P\_ID;

From the above view,

Joining Player and country table.

Retrieved columns: player name and age