

Final Project Report For Sports Aggregator Information System

ISM 6218 – Advanced Database Management

Prepared by:
Rohith Movva
Srinivas Raju Nanapuram
Venkata Ravi Teja Lanka
Naif Alawi
Yaassh Rao

USF Muma College of Business

21st November, 2017

Abstract

In this report, our team has developed a sports aggregator information system that offers a unique experience for all types of users. Unlike other sports aggregation systems that are limited in terms of functionalities that they offer to the users; this system was developed to provide a comprehensive experience for all sports enthusiasts. The system is designed from the backend at the database to the frontend of the user interface with the sports enthusiasts' experience in mind. The system will offer a wide variety of functions that are designed using the expert system and user interactive system. Users of the system will have a tailored experience in which they can follow their favourite sports, publish expert authenticated articles, posts opinions about their sports, follow their sports news feeds, and access to purchasing game tickets of their favourite sports games and events. The report will provide a detailed description on the database design and development as well as the development of the actual user interface.

Table of Contents

1.	Introduction.....	5
2.	System Design Plan	6
3.	Database Design.....	7
3.1.	Assumptions.....	7
3.2.	Conceptual ER Diagram.....	7
3.3.	Logical Design of Database	7
4.	Data Integrity	9
4.1.	Constraints.....	9
4.2.	Tables	10
5.	Data Generation and Loading	28
5.1.	Manual Data Generation and Loading	28
5.2.	Sequences	32
5.3.	Date Arithmetic	32
5.4.	DBMS_RANDOM.....	33
5.5.	PROCEDURE	33
6.	Query Writing	34
6.1.	Application Initiated Queries	34
6.1.1.	Registration Page	34
6.1.2.	Login Page	35
6.2.	PL/SQL Package	37
6.2.1.	BASEBALL_DATA_PKG SPECIFICATION	37
6.2.2.	BASEBALL_DATA_PKG BODY	38
6.2.2.1.	SEQ_NEXTVAL_ON_DEMAND FUNCTION	38
6.2.2.2.	BASEBALL_DATA_PRC PROCEDURE	39
6.2.3.	EXECUTION OF PACKAGE.....	41
7.	Performance Tuning.....	43
7.1.	Function-Based Index Experiment.....	43
7.2.	Transaction Level Read consistency	44
8.	Application Development	48
8.1.	Working on Database	49
8.2.	APPLICATION OVERVIEW	50
	System's Home Page	50

Final Project Report for Sports Aggregator Information System

Registration Page	51
Login Page	53
User's Home Page.....	54
User's Posts Page.....	56
Agent's Home Page	59
Agent's Posts Page.....	61
Article Published to all users	62
Agent's Homepage.....	63
User's Home Page.....	64
9. Interesting Queries	64
9.1. For Size of all tables.....	64
9.2. For current open Sessions	65
9.3. For Memory allocated to current sessions.....	66
9.4. For Database time properties.....	67
10. Conclusion	68

1. Introduction

Sports Aggregator Information System is a web interface system that has been developed has a combination of expert system (agents posting their articles being the experts) and user interactive system (here user is also able to post their articles or views, but it would be validated by an expert before publishing it on the website). While there exist many websites that provide sports aggregation systems such as Yahoo Sports, Fox Sports, Sportskeeda.com, and many others; these systems are limited in terms of functionality as they are either expert systems or user interactive system but not the combination of both. On the other hand, the system that our group is developing is comprehensive and provides a combination of functionalities that are provided by the expert system as well as user interactive system.

In the sports aggregator information system, the users will be able to follow the sports that they like. The system initially will offer 9 sports for users to follow in which users can choose. Further expansion of the sports offering can be added in the future. Some of the functionalities that the system will include are articles authentication, interface and news customization, game tickets referral links, and interest sports news feeds.

The development of the system can be divided into three stages: the system design planning stage, the database design stage, and the user interface (UI) stage. Prior to In the database design stage, our team developed an overall system design plan that identifies the requirement of the system and functionalities such as the ones mentioned earlier. Following this stage, initial entity relation diagrams were developed and used as the foundations for developing the actual database that will serve as the backbone for the sports aggregator system. Afterwards, the user interface stage was initiated to develop a user-friendly interface that is integrated with the database to allow users a fluent interaction with the system.

Using tools such as SQL developer for developing the database, the system database is consisting of over 22 entities that stores different information such as users' registration and sports information. In the development of these entities, several issues were carefully considered including the choice of primary keys, constraints, and values. The entities contain dummy data for demonstration. Furthermore, the system was developed with expandability in mind so that further additions of sports is feasible. The UI was developed using ASP.NET, C#, HTML, JavaScript and is fully integrated with the ORACLE database. Moreover, several factors were taken into consideration at the development of the UI to ensure that the interface is user friendly.

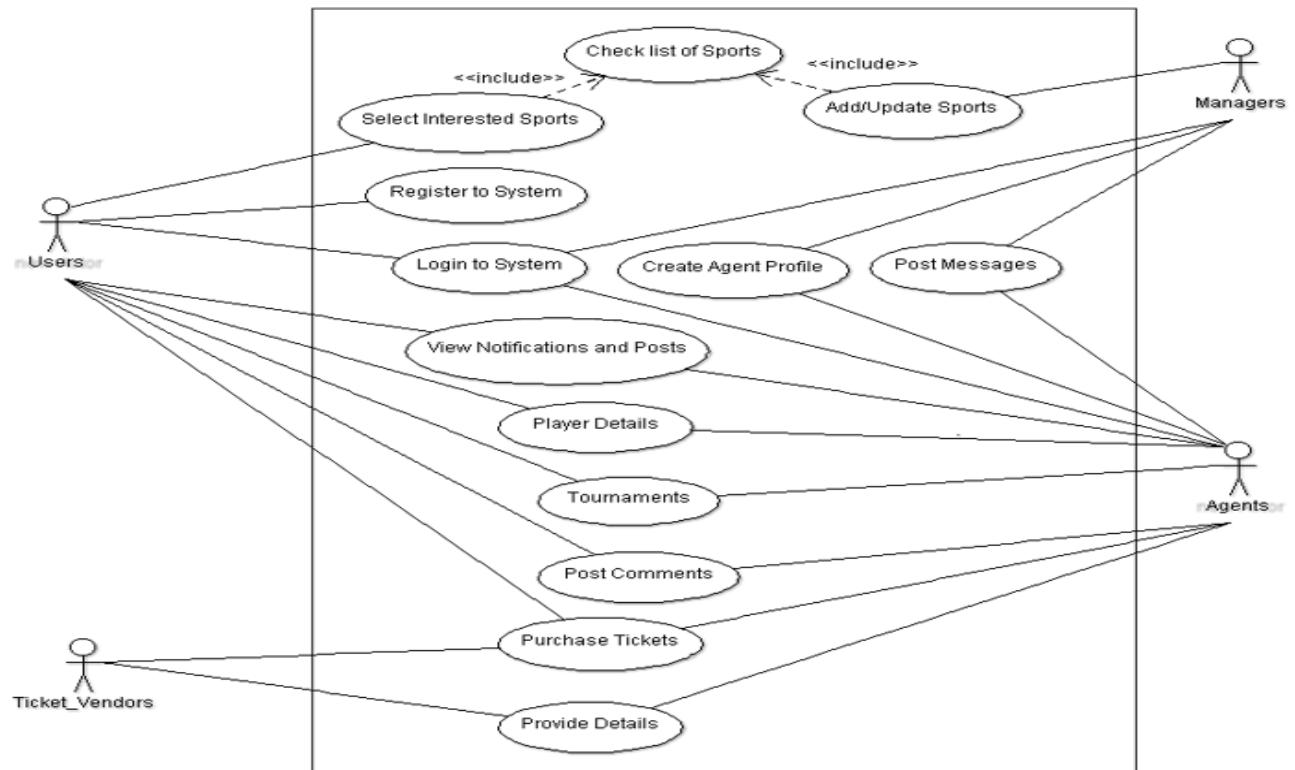
The following sections of this report will discuss in detail the processes that were undertaken for the database design, query writing, performance tuning, and application development. These sections will include actual queries used in the development, screen shots of the database system, and a user story demonstration to provide a detailed view on how the system will interact with users.

2. System Design Plan

During the system design plan, we finalized the requirements from the responses received from the questionnaire posted by us. Finally, we have concluded the actors that are going to utilize the system are Users, Agents, Managers and Ticket Vendors.

- **Users:** The Users should register to Sports Aggregator system by selecting his/her interested sports and log-in to system to view the latest updates in their interested sports, player statistics, tournament details, purchase tickets for the tournaments by navigating to the respective Vendor's page and post the articles.
- **Agents:** The Agents updates the database with the player, tournament and ticket vendors details. Also, Agents posts his/her articles and validates the articles posted by users before publishing it to all users of the system. And agents provide the notifications/latest updates of a sport to the respective users.
- **Managers:** The Manager will recruit the agents for various sports and creates their profile in the system. Manager is the only one who adds/updates/deletes the sports details in the system. Manager posts messages to the agents.
- **Ticket Vendors:** Ticket vendors provide the URL link details for purchasing tickets for the tournaments, which later validated by agent and updated into the system.

Use Case diagram of the system



3. Database Design

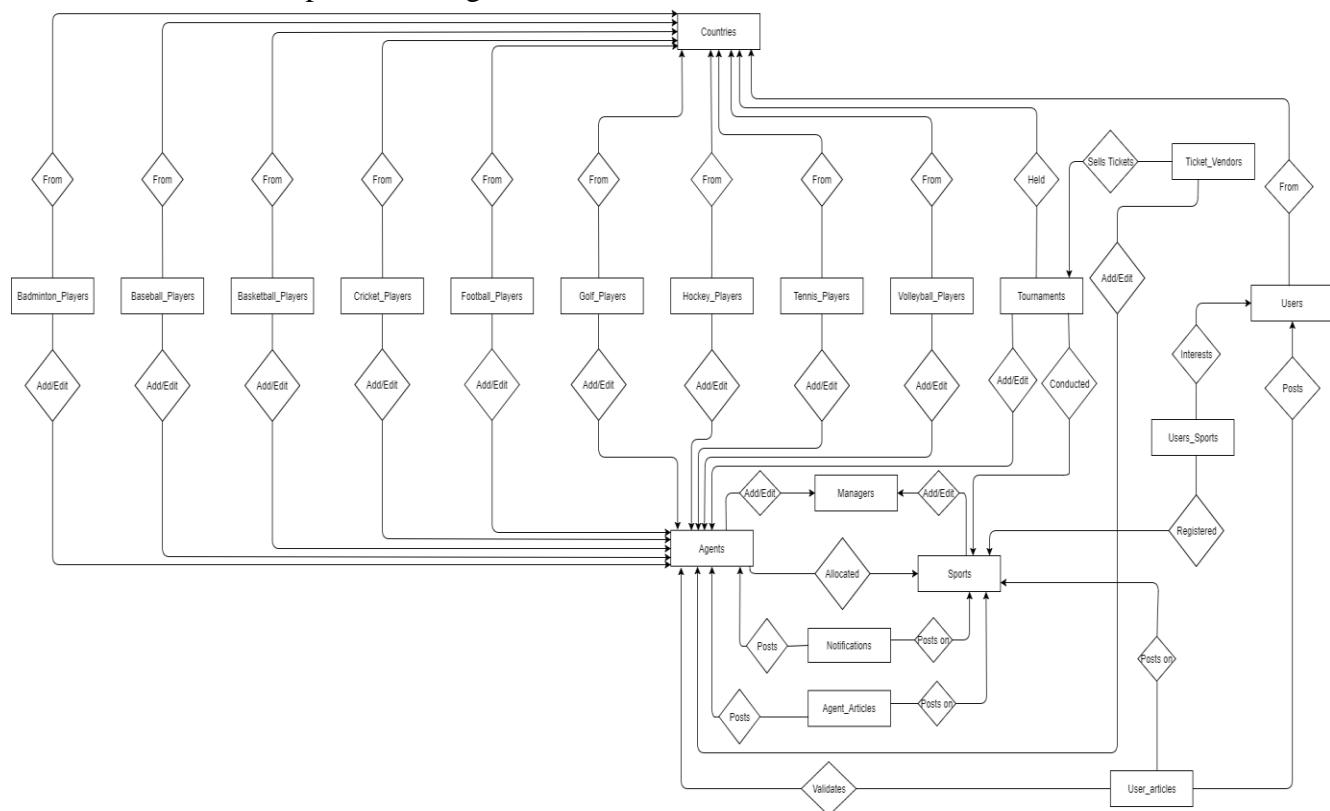
3.1. Assumptions

The following are the assumptions made while designing the database for this system

- User must register for at least one sport
- Same username can't be associated to different users
- Password should be alpha-numeric with minimum of 8 characters & it can have special characters like @, \$, #
- To get registered mandatory fields must be filled
- A user can post many articles
- An agent can validate many articles
- Agents can insert/update one or more tournaments, players, ticket vendors list
- Manager can insert/update one or more agents profile details, sports details

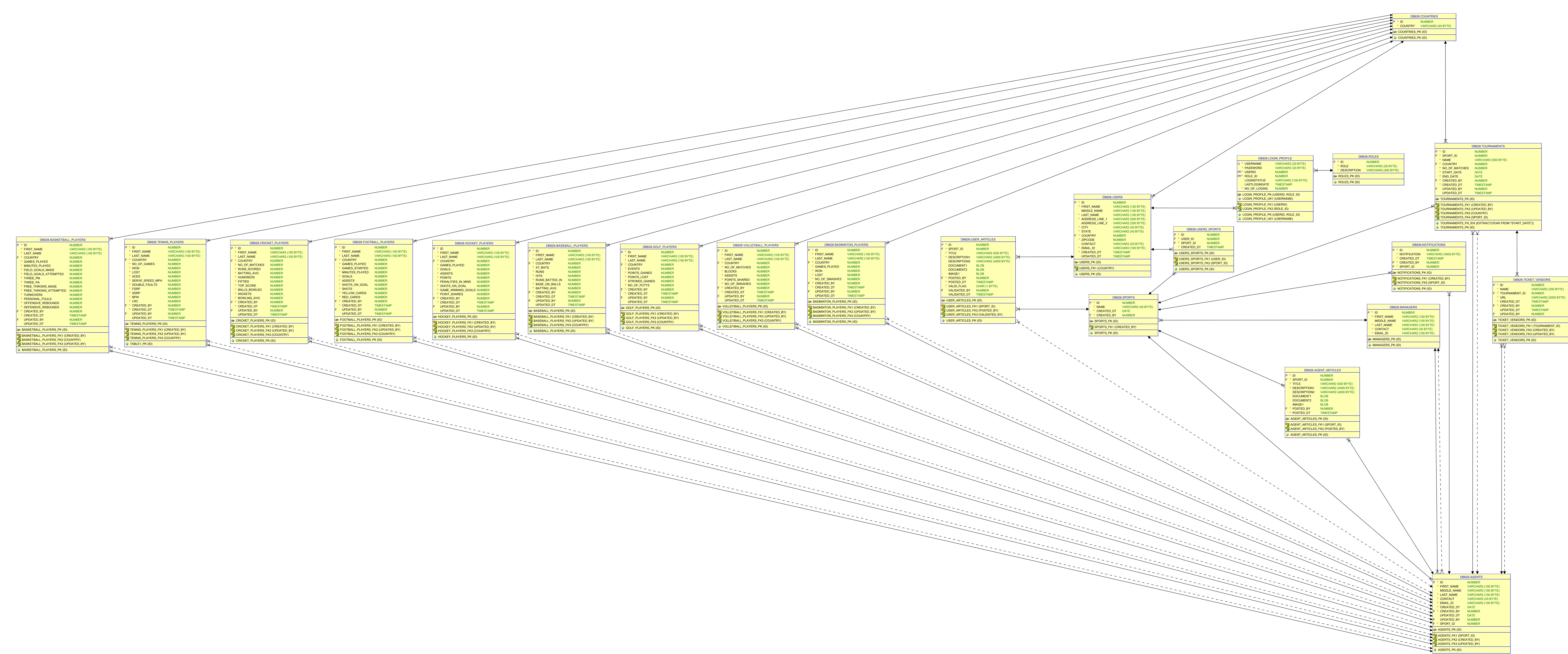
3.2. Conceptual ER Diagram

The below is the conceptual ER diagram of the database



3.3. Logical Design of Database

The following design is the Logical design of the database.



4. Data Integrity

4.1. Constraints

Primary Key: We have primary key constraints available in all our database tables. Here we are demonstrating the purpose of primary key on a single attribute as well as multiple attributes.

(a)Single Attribute: For instance, we are showing the scenario using “**USER_ARTICLES**” table. Here we are having “ID” attribute as primary key to uniquely identify an article posted by a user among many articles recorded in the table.

(b)Multiple Attributes (2): We are showing this scenario using “**LOGIN_PROFILE**” table. Since we have three different types of actors (USERS, AGENTS & MANAGERS), to differentiate these actors we have allocated role ID’s as 3,2,1 respectively and we have three different tables to store the profile details of the actors. So, to differentiate among login details of actors we are having primary key on combination of two attributes namely “USERID” and “ROLE_ID”.

Foreign Key: In the same “**USER_ARTICLES**” table we are having three foreign keys on “SPORT_ID”, “POSTED_BY” and “VALIDATED_BY” attributes which are referring to primary keys of “**SPORTS**”, “**USERS**” and “**AGENTS**” tables respectively. These foreign keys are used to describe about the article to which sport it is related to, the user who has posted this article and the agent who has validated it.

Check Constraint: In the same “**USER_ARTICLES**” table we are using check constraint on “VALID_FLAG” to accept the character values “Y” and “N” only. The purpose of this check constraint is to throw an error if any other character is insert/update to that attribute.

Unique Constraint: In the same “**LOGIN_PROFILE**” table we are using unique constraint on “USERNAME” attribute. The purpose of this constraint to ensure that no two users are having the same username.

Not Null Constraint: In the same “**USER_ARTICLES**” table we are using Not Null constraint on “TITLE”, “DESCRIPTION1” and “POSTED_DT” attributes along with the attributes mentioned while discussing the primary, foreign and check constraints. All these attributes are mandatory to an article posted by a user.

Below are the **DDL STATEMENTS** for the tables mentioned above:

a. USER_ARTICLES DDL:

```
CREATE TABLE "DB626"."USER_ARTICLES"
(
    "ID" NUMBER NOT NULL ENABLE,
    "SPORT_ID" NUMBER NOT NULL ENABLE,
    "TITLE" VARCHAR2(500 BYTE) NOT NULL ENABLE,
```

```
"DESCRIPTION1" VARCHAR2(4000 BYTE) NOT NULL ENABLE,
"DESCRIPTION2" VARCHAR2(4000 BYTE),
"DOCUMENT1" BLOB,
"DOCUMENT2" BLOB,
"IMAGE1" BLOB,
"POSTED_BY" NUMBER NOT NULL ENABLE,
"POSTED_DT" TIMESTAMP (6) NOT NULL ENABLE,
"VALID_FLAG" CHAR(1 BYTE) NOT NULL ENABLE,
"VALIDATED_BY" NUMBER,
"VALIDATED_DT" TIMESTAMP (6),
CONSTRAINT "USER_ARTICLES_PK" PRIMARY KEY ("ID") ENABLE,
CONSTRAINT "USER_ARTICLES_CHK1" CHECK (VALID_FLAG IN
('Y','N')) ENABLE,
CONSTRAINT "USER_ARTICLES_FK1" FOREIGN KEY ("SPORT_ID")
REFERENCES "DB626"."SPORTS" ("ID") ENABLE,
CONSTRAINT "USER_ARTICLES_FK2" FOREIGN KEY ("POSTED_BY")
REFERENCES "DB626"."USERS" ("ID") ENABLE,
CONSTRAINT "USER_ARTICLES_FK3" FOREIGN KEY
("VALIDATED_BY")
REFERENCES "DB626"."AGENTS" ("ID") ENABLE
);
```

b. LOGIN_PROFILE DDL:

```
CREATE TABLE "DB626"."LOGIN_PROFILE"
(
    "USERNAME" VARCHAR2(20 BYTE) NOT NULL ENABLE,
    "PASSWORD" VARCHAR2(20 BYTE) NOT NULL ENABLE,
    "USERID" NUMBER NOT NULL ENABLE,
    "ROLE_ID" NUMBER NOT NULL ENABLE,
    "LOGINSTATUS" VARCHAR2(100 BYTE),
    "LASTLOGINDATE" TIMESTAMP (6),
    "NO_OF_LOGINS" NUMBER DEFAULT 0 NOT NULL ENABLE,
    CONSTRAINT "LOGIN_PROFILE_PK" PRIMARY KEY ("USERID",
    "ROLE_ID") ENABLE,
    CONSTRAINT "LOGIN_PROFILE_UK1" UNIQUE ("USERNAME")
ENABLE,
    CONSTRAINT "LOGIN_PROFILE_FK1" FOREIGN KEY ("USERID")
REFERENCES "DB626"."USERS" ("ID") ENABLE,
    CONSTRAINT "LOGIN_PROFILE_FK2" FOREIGN KEY ("ROLE_ID")
REFERENCES "DB626"."ROLES" ("ID") ENABLE
);
```

4.2. Tables

There are total 22 tables that has been created in database for this Information System. The below mentioned are the details providing the significance, DDL, list of attributes and significance of those attributes.

AGENT_ARTICLES

This table is used to store the details about the articles posted by agents related to all sports. This table is also used to display the agent's articles in the homepage after login to the web application. The data inserted into this table from application when agents post article in their posts page. The list of attributes and significance of each attribute is as follows



The DDL Statement for AGENT_ARTICLES table is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify an article from the table
2	SPORT_ID	NUMBER	No	This attribute is used to identify the sport to which this article belongs to
3	TITLE	VARCHAR2(500 BYTE)	No	This is the title given by agent to the article
4	DESCRIPTION1	VARCHAR2(4000 BYTE)	No	This gives some description about the article
5	DESCRIPTION2	VARCHAR2(4000 BYTE)	Yes	This gives some description about the article
6	DOCUMENT1	BLOB	Yes	This contains the reference document 1 file uploaded by agent while posting article
7	DOCUMENT2	BLOB	Yes	This contains the reference document 2 file uploaded by agent while posting article
8	IMAGE1	BLOB	Yes	This contains the image file uploaded by agent
9	POSTED_BY	NUMBER	No	This contains the ID value of agent who posted the respective article
10	POSTED_DT	TIMESTAMP (6)	No	This refers to the date when the article has been posted

AGENTS

This table stores the details of the agents who were recruited by managers. The data will be stored in this table when manager creates agent profile from his Add/Edit Agents Page in the application. The list of attributes and significance of each attribute is as follows



AGENTS_DDLsql

The DDL Statement for AGENTS table is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify an agent from the table
2	FIRST_NAME	VARCHAR2(100 BYTE)	No	First Name of the agent
3	MIDDLE_NAME	VARCHAR2(100 BYTE)	Yes	Middle Name of the agent
4	LAST_NAME	VARCHAR2(100 BYTE)	No	Last Name of the agent
5	CONTACT	VARCHAR2(20 BYTE)	No	Contact Number of the respective agent. It is in varchar2 since it has some special characters like (, - Example: (813)-340-2610
6	EMAIL_ID	VARCHAR2(100 BYTE)	No	Email Id of the agent
7	CREATED_DT	DATE	No	This refers to date when this agent profile has been created
8	CREATED_BY	NUMBER	No	This refers to the ID of the Manager who created this agents profile
9	UPDATED_DT	DATE	Yes	This refers to date when this agent profile has been last updated
10	UPDATED_BY	NUMBER	Yes	This refers to ID of the Manager who updated this agents profile
11	SPORT_ID	NUMBER	No	This refers to ID of the sport for which this agent has been recruited

BADMINTON_PLAYERS

This table stores the details of the badminton players. The data inserted into this table from the application when agent adds/edits the player details related to Badminton. The list of attributes and significance of each attribute is as follows



The DDL Statement for BADMINTON_PLAYERS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a player from the table
2	FIRST_NAME	VARCHAR2(100 BYTE)	No	First Name of the player

3	LAST_NAME	VARCHAR2(100 BYTE)	No	Last Name of the Player
4	COUNTRY	NUMBER	No	The Id of the country to which the player belongs to
5	GAMES_PLAYED	NUMBER	No	Total number of games played
6	WON	NUMBER	No	Total Number of games won by the player
7	LOST	NUMBER	No	Total Number of games lost by the player
8	NO_OF_SMASHES	NUMBER	No	Total Number of smashes by the player
9	CREATED_BY	NUMBER	No	The Id of the Agent who added this player in the application
10	CREATED_DT	TIMESTAMP (6)	No	This refers to the date when the Player has been added by agent
11	UPDATED_BY	NUMBER	Yes	The Id of the Agent who edited the player details in application
12	UPDATED_DT	TIMESTAMP (6)	Yes	This refers to the date when the Player details has been last updated by an agent

BASEBALL_PLAYERS

This table stores the details of the Baseball players. The data inserted into this table from the application when agent adds/edits the player details related to Baseball. The list of attributes and significance of each attribute is as follows



The DDL Statement for BASEBALL_PLAYERS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a player from the table
2	FIRST_NAME	VARCHAR2(100 BYTE)	No	First Name of the player
3	LAST_NAME	VARCHAR2(100 BYTE)	No	Last Name of the Player
4	COUNTRY	NUMBER	No	The Id of the country to which the player belongs to
5	AT_BATS	NUMBER	No	At_Bats by player
6	RUNS	NUMBER	No	Total runs scored by player
7	HITS	NUMBER	No	Total hits by the player
8	RUNS_BATTED_IN	NUMBER	No	Total runs batted in by the player
9	BASE_ON_BALLS	NUMBER	No	Total base on balls by the player
10	BATTING_AVG	NUMBER	No	Batting Average of the player

11	CREATED_BY	NUMBER	No	The Id of the Agent who added this player in the application
12	CREATED_DT	TIMESTAMP (6)	No	This refers to the date when the Player has been added by agent
13	UPDATED_BY	NUMBER	Yes	The Id of the Agent who edited the player details in application
14	UPDATED_DT	TIMESTAMP (6)	Yes	This refers to the date when the Player details has been last updated by an agent

BASKETBALL_PLAYERS

This table stores the details of the Basketball players. The data inserted into this table from the application when agent adds/edits the player details related to Basketball. The list of attributes and significance of each attribute is as follows



The DDL Statement for BASKETBALL_PLAYERS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a player from the table
2	FIRST_NAME	VARCHAR2(100 BYTE)	No	First Name of the player
3	LAST_NAME	VARCHAR2(100 BYTE)	No	Last Name of the Player
4	COUNTRY	NUMBER	No	The Id of the country to which the player belongs to
5	GAMES_PLAYED	NUMBER	No	Total Number of Games played by the player
6	MINUTES_PLAYED	NUMBER	No	Total Number of Minutes played by the player
7	FIELD_GOALS_MADE	NUMBER	No	Total Number of Field Goals made by the player
8	FIELD_GOALS_ATTEMPTED	NUMBER	No	Total Number of Field Goals attempted by the player
9	THREE_PM	NUMBER	No	Total Number of Three Pointers made by the player
10	THREE_PA	NUMBER	No	Total Number of Three pointers attempted by the player

11	FREE_THROWS_MADE	NUMBER	No	Total Number of Free Throws made by the player
12	FREE_THROWS_ATTEMPTED	NUMBER	No	Total Number of Free Throws attempted by the player
13	TURNOVERS	NUMBER	No	Total Turnovers of the player
14	PERSONAL_FOULS	NUMBER	No	Total Number of Personal fouls by the player
15	OFFENSIVE_REBOUNDS	NUMBER	No	Total Number of Offensive rebounds by the player
16	DEFENSIVE_REBOUNDS	NUMBER	No	Total Number of Defensive rebounds by the player
17	CREATED_BY	NUMBER	No	The Id of the Agent who added this player in the application
18	CREATED_DT	TIMESTAMP (6)	No	This refers to the date when the Player has been added by agent
19	UPDATED_BY	NUMBER	Yes	The Id of the Agent who edited the player details in application
20	UPDATED_DT	TIMESTAMP (6)	Yes	This refers to the date when the Player details has been last updated by an agent

COUNTRIES

This table contains the countries list for whom this Information System available. The list of attributes and significance of each attribute is as follows



The DDL Statement for BADMINTON_PLAYERS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a country from the table
2	COUNTRY	VARCHAR2(40 BYTE)	No	This attribute contains the name of the country

CRICKET_PLAYERS

This table stores the details of the Cricket players. The data inserted into this table from the application when agent adds/edits the player details related to Cricket. The list of attributes and significance of each attribute is as follows



CRICKET_PLAYERS_D
DLsql

The DDL Statement for CRICKET_PLAYERS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a player from the table
2	FIRST_NAME	VARCHAR2(100 BYTE)	No	First Name of the player
3	LAST_NAME	VARCHAR2(100 BYTE)	No	Last Name of the Player
4	COUNTRY	NUMBER	No	The Id of the country to which the player belongs to
5	NO_OF_MATCHES	NUMBER	No	Total number of Matches played by the player
6	RUNS_SCORED	NUMBER	No	Total number of runs scored by the player
7	BATTING_AVG	NUMBER	No	Batting Average of the player
8	HUNDREDS	NUMBER	No	Total number of Hundreds by the Player
9	FIFTIES	NUMBER	No	Total Number of Fifties by the Player
10	TOP_SCORE	NUMBER	No	The Highest runs scored by a player
11	BALLS_BOWLED	NUMBER	No	Total Number of Balls bowled by the player
12	WICKETS	NUMBER	No	Total number of wickets taken by the player
13	BOWLING_AVG	NUMBER	No	Bowling average of the player
14	CREATED_BY	NUMBER	No	The Id of the Agent who added this player in the application
15	CREATED_DT	TIMESTAMP (6)	No	This refers to the date when the Player has been added by agent
16	UPDATED_BY	NUMBER	Yes	The Id of the Agent who edited the player details in application
17	UPDATED_DT	TIMESTAMP (6)	Yes	This refers to the date when the Player details has been last updated by an agent

FOOTBALL_PLAYERS

This table stores the details of the Football players. The data inserted into this table from the application when agent adds/edits the player details related to Football. The list of attributes and significance of each attribute is as follows



FOOTBALL_PLAYERS
_DDLsql

The DDL Statement for FOOTBALL_PLAYERS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a player from the table
2	FIRST_NAME	VARCHAR2(100 BYTE)	No	First Name of the player
3	LAST_NAME	VARCHAR2(100 BYTE)	No	Last Name of the Player
4	COUNTRY	NUMBER	No	The Id of the country to which the player belongs to
5	GAMES_PLAYED	NUMBER	No	Total number of games played by the player
6	GAMES_STARTED	NUMBER	No	Total number of games started by the player
7	MINUTES_PLAYED	NUMBER	No	Total number of minutes played by the player
8	GOALS	NUMBER	No	Total number of Goals by the player
9	ASSISTS	NUMBER	No	Total number of Assists by the player
10	SHOTS_ON_GOAL	NUMBER	No	Total number shots on the Goal by the player
11	SHOTS	NUMBER	No	Total number of shots by the player
12	YELLOW_CARDS	NUMBER	No	Total number of Yellow cards received by the player
13	RED_CARDS	NUMBER	No	Total number of Red cards received by the player
14	CREATED_BY	NUMBER	No	The Id of the Agent who added this player in the application
15	CREATED_DT	TIMESTAMP (6)	No	This refers to the date when the Player has been added by agent
16	UPDATED_BY	NUMBER	Yes	The Id of the Agent who edited the player details in application
17	UPDATED_DT	TIMESTAMP (6)	Yes	This refers to the date when the Player details has been last updated by an agent

GOLF_PLAYERS

This table stores the details of the Golf players. The data inserted into this table from the application when agent adds/edits the player details related to Golf. The list of attributes and significance of each attribute is as follows



GOLF_PLAYERS_DDL
.sql

The DDL Statement for GOLF_PLAYERS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a player from the table
2	FIRST_NAME	VARCHAR2(100 BYTE)	No	First Name of the player
3	LAST_NAME	VARCHAR2(100 BYTE)	No	Last Name of the Player
4	COUNTRY	NUMBER	No	The Id of the country to which the player belongs to
5	EVENTS	NUMBER	No	Total Number of Events of the player
6	POINTS_GAINED	NUMBER	No	Total number of points gained by the player
7	POINTS_LOST	NUMBER	No	Total number of points lost by the player
8	STROKES_GAINED	NUMBER	No	Total number of strokes gained by the player
9	NO_OF_PUTTS	NUMBER	No	Total number of putts into the hole done by the player
10	CREATED_BY	NUMBER	No	The Id of the Agent who added this player in the application
11	CREATED_DT	TIMESTAMP (6)	No	This refers to the date when the Player has been added by agent
12	UPDATED_BY	NUMBER	Yes	The Id of the Agent who edited the player details in application
13	UPDATED_DT	TIMESTAMP (6)	Yes	This refers to the date when the Player details has been last updated by an agent

HOCKEY_PLAYERS

This table stores the details of the Hockey players. The data inserted into this table from the application when agent adds/edits the player details related to Hockey. The list of attributes and significance of each attribute is as follows



HOCKEY_PLAYERS_D
DLSql

The DDL Statement for HOCKEY_PLAYERS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a player from the table
2	FIRST_NAME	VARCHAR2(100 BYTE)	No	First Name of the player
3	LAST_NAME	VARCHAR2(100 BYTE)	No	Last Name of the Player
4	COUNTRY	NUMBER	No	The Id of the country to which the player belongs to
5	GAMES_PLAYED	NUMBER	No	Total Number of games played by the player
6	GOALS	NUMBER	No	Total number of Goals by the player
7	ASSISTS	NUMBER	No	Total number of Assists by the player
8	POINTS	NUMBER	No	Total number of Points scored by the player
9	PENALITIES_IN_MINS	NUMBER	No	Total Number of penalties in mins by the player
10	SHOTS_ON_GOAL	NUMBER	No	Total Number of shots on goal by the player
11	GAME_WINNING_GOALS	NUMBER	No	Total Number of game winning goals by the player
12	POINT SHARES	NUMBER	No	Total Number of points contributed by the player
13	CREATED_BY	NUMBER	No	The Id of the Agent who added this player in the application
14	CREATED_DT	TIMESTAMP (6)	No	This refers to the date when the Player has been added by agent
15	UPDATED_BY	NUMBER	Yes	The Id of the Agent who edited the player details in application
16	UPDATED_DT	TIMESTAMP (6)	Yes	This refers to the date when the Player details has been last updated by an agent

LOGIN_PROFILE

This table stores the login details of Users, Agents and Managers. The user's data inserted into this table from application during registration. The data in this table is updated during login and manage accounts from application. The list of attributes and significance of each attribute is as follows



The DDL Statement for LOGIN_PROFILE is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	USERNAME	VARCHAR2(20 BYTE)	No	Username of the actor
2	PASSWORD	VARCHAR2(20 BYTE)	No	Password of the actor
3	USERID	NUMBER	No	The ID of the actor in their respective tables
4	ROLE_ID	NUMBER	No	The role of the actor
5	LOGINSTATUS	VARCHAR2(100 BYTE)	Yes	Login status of the actor
6	LASTLOGINDATE	TIMESTAMP (6)	Yes	Last successful login date of the actor
7	NO_OF_LOGINS	NUMBER	No	Total number of times logged in by the actor

MANAGERS

This table stores the details of the Managers. The list of attributes and significance of each attribute is as follows



The DDL Statement for MANAGERS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a Manager from the table
2	FIRST_NAME	VARCHAR2(100 BYTE)	No	First Name of the Manager
3	MIDDLE_NAME	VARCHAR2(100 BYTE)	Yes	Middle Name of the Manager
4	LAST_NAME	VARCHAR2(100 BYTE)	No	Last Name of the Manager
5	CONTACT	VARCHAR2(20 BYTE)	No	Contact Number of the Manager. It is in varchar2 since it has some special

				characters like (, - Example: (813)-340-2610
6	EMAIL_ID	VARCHAR2(100 BYTE)	No	Email Id of the Manager

NOTIFICATIONS

This table has the notifications posted by an agent related to a sport. This table is used to display the latest notifications related to their interested sports on the Homepage after login. The list of attributes and significance of each attribute is as follows



The DDL Statement for NOTIFICATIONS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a notification from the table
2	NOTIFICATION	VARCHAR2(4000 BYTE)	No	This contains the notification message
3	CREATED_DT	TIMESTAMP (6)	No	This refers to the date when the notification has been added by agent
4	CREATED_BY	NUMBER	No	The Id of the Agent who added this notification in the application
5	SPORT_ID	NUMBER	No	This attribute is used to identify the sport to which this notification belongs to

ROLES

This table has list of type of actors that uses the system. The list of attributes and significance of each attribute is as follows



The DDL Statement for ROLES is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a type of actor from the table
2	ROLE	VARCHAR2(20 BYTE)	No	This refers to the role of the actor
3	DESCRIPTION	VARCHAR2(400 BYTE)	No	Describes briefly about the actor

SPORTS

This table has list of sports available in the system. The data inserted into this table when Manager add new sports into the system. The list of attributes and significance of each attribute is as follows



SPORTS_DDLsql

The DDL Statement for SPORTS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a sport from the table
2	NAME	VARCHAR2(40 BYTE)	No	Name of the Sport
3	CREATED_DT	DATE	No	This refers to the date when the sport has been added by Manager
4	CREATED_BY	NUMBER	No	The Id of the Manager who added this sport in the application

TENNIS_PLAYERS

This table stores the details of the Tennis players. The data inserted into this table from the application when agent adds/edits the player details related to Tennis. The list of attributes and significance of each attribute is as follows



TENNIS_PLAYERS_D
DLsql

The DDL Statement for TENNIS_PLAYERS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a player from the table
2	FIRST_NAME	VARCHAR2(100 BYTE)	No	First Name of the player
3	LAST_NAME	VARCHAR2(100 BYTE)	No	Last Name of the Player
4	COUNTRY	NUMBER	No	The Id of the country to which the player belongs to
5	NO_OF_GAMES	NUMBER	No	Total number of games played by the player
6	WON	NUMBER	No	Total Number of games won by the player
7	LOST	NUMBER	No	Total Number of games lost by the player

8	ACES	NUMBER	No	Total number of aces by the player
9	SERVE_SPEED_MPH	NUMBER	No	The maximum serve speed of the player in MPH
10	DOUBLE_FAULTS	NUMBER	No	Total number of double faults by the player
11	FSRP	NUMBER	No	The total first serve return points scored by the player
12	SSRP	NUMBER	No	The total second serve return points scored by the player
13	BPW	NUMBER	No	The total break points won by the player
14	LRC	NUMBER	No	The longest rally count of the player
15	CREATED_BY	NUMBER	No	The Id of the Agent who added this player in the application
16	CREATED_DT	TIMESTAMP (6)	No	This refers to the date when the Player has been added by agent
17	UPDATED_BY	NUMBER	Yes	The Id of the Agent who edited the player details in application
18	UPDATED_DT	TIMESTAMP (6)	Yes	This refers to the date when the Player details has been last updated by an agent

TICKET_VENDORS

This table has the details of ticket vendors for various tournaments. The data in this table is inserted from the application when agent adds the details in tickets page of agent. The data is displayed to users in their respective tickets page from which they can redirect to their interested vendor's URL for purchasing the tickets. The list of attributes and significance of each attribute is as follows



TICKET_VENDORS_D
DLsql

The DDL Statement for TICKET_VENDORS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a ticket vendor from the table
2	NAME	VARCHAR2(200 BYTE)	No	Name of the Ticket Vendor
3	TOURNAMENT_ID	NUMBER	No	The tournament ID for which the vendor is providing tickets
4	URL	VARCHAR2(2000 BYTE)	No	The vendor's URL for purchasing tickets

5	CREATED_DT	TIMESTAMP (6)	No	This refers to the date when the details have been added by agent
6	CREATED_BY	NUMBER	No	The Id of the Agent who added these details in the application
7	UPDATED_DT	TIMESTAMP (6)	Yes	This refers to the date when the details have been last updated by an agent
8	UPDATED_BY	NUMBER	Yes	The Id of the Agent who edited these details in application

TOURNAMENTS

This table has the details of tournament details of all sports. The data will be inserted/updated in this table from application when agent adds/edits the details in his respective tournaments page. The data is displayed to the users for their interested sports in their respective tournaments page. The list of attributes and significance of each attribute is as follows



The DDL Statement for TOURNAMENTS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a tournament from the table
2	SPORT_ID	NUMBER	No	The ID of the sport to which the tournament related to
3	NAME	VARCHAR2(500 BYTE)	No	The name of the tournament
4	COUNTRY	NUMBER	No	The Id of the country where the tournament held
5	NO_OF_MATCHES	NUMBER	No	The total number of matches in the tournament
6	START_DATE	DATE	No	The start date of the tournament
7	END_DATE	DATE	No	The end date of the tournament
8	CREATED_BY	NUMBER	No	The Id of the Agent who added the tournament in the application
9	CREATED_DT	TIMESTAMP (6)	No	This refers to the date when the details have been added by agent
10	UPDATED_BY	NUMBER	Yes	The Id of the Agent who edited these details in application

11	UPDATED_DT	TIMESTAMP (6)	Yes	This refers to the date when the details have been last updated by an agent
-----------	-------------------	---------------	-----	---

USER_ARTICLES

This table is used to store the details about the articles posted by users related to all sports. This table is also used to display the valid user articles in the homepage after login to the web application. The data inserted into this table from application when users post article in their posts page. The list of attributes and significance of each attribute is as follows



The DDL Statement for USER_ARTICLES table is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify an article from the table
2	SPORT_ID	NUMBER	No	This attribute is used to identify the sport to which this article belongs to
3	TITLE	VARCHAR2(500 BYTE)	No	This is the title given by user to the article
4	DESCRIPTION1	VARCHAR2(4000 BYTE)	No	This gives some description about the article
5	DESCRIPTION2	VARCHAR2(4000 BYTE)	Yes	This gives some description about the article
6	DOCUMENT1	BLOB	Yes	This contains the reference document 1 file uploaded by user while posting article
7	DOCUMENT2	BLOB	Yes	This contains the reference document 2 file uploaded by user while posting article
8	IMAGE1	BLOB	Yes	This contains the image file uploaded by user
9	POSTED_BY	NUMBER	No	This contains the ID value of user who posted the respective article
10	POSTED_DT	TIMESTAMP (6)	No	This refers to the date when the article has been posted by user
11	VALID_FLAG	CHAR (1 BYTE)	No	This refers whether the article is valid or not
12	VALIDATED_BY	NUMBER	Yes	This contains the ID value of agent who validated the article

13	VALIDATED_DT	TIMESTAMP (6)	Yes	This refers to the date when the article has been validated by an agent
----	---------------------	---------------	-----	---

USERS

This table stores the details of the users who registered to system. The data will be stored in this table when user registers to system from registration page in the application. The data can be updated from application by user from manage account details page. The list of attributes and significance of each attribute is as follows



The DDL Statement for USERS table is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a user from the table
2	FIRST_NAME	VARCHAR2(100 BYTE)	No	First Name of the user
3	MIDDLE_NAME	VARCHAR2(100 BYTE)	Yes	Middle Name of the user
4	LAST_NAME	VARCHAR2(100 BYTE)	No	Last Name of the user
5	ADDRESS_LINE_1	VARCHAR2(200 BYTE)	No	Address line 1 of the user
6	ADDRESS_LINE_2	VARCHAR2(200 BYTE)	Yes	Address line 2 of the user
7	CITY	VARCHAR2(40 BYTE)	No	City of the user
8	STATE	VARCHAR2(40 BYTE)	No	State of the User
9	COUNTRY	NUMBER	No	The ID of the country of the user
10	ZIPCODE	NUMBER	No	Zip code of the user
11	CONTACT	VARCHAR2(20 BYTE)	Yes	Contact Number of the respective agent. It is in varchar2 since it has some special characters like (, - Example: (813)-340-2610
12	EMAIL_ID	VARCHAR2(100 BYTE)	No	Email Id of the agent
13	CREATED_DT	TIMESTAMP (6)	No	This refers to when the user has registered to the system
14	UPDATED_DT	TIMESTAMP (6)	Yes	This refers to when user has last updated their details

USERS_SPORTS

This table contains the sports to which the users have registered. The list of attributes and significance of each attribute is as follows



The DDL Statement for USERS_SPORTS table is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a user, sport combination from the table
2	USER_ID	NUMBER	No	This contains the ID of the users
3	SPORT_ID	NUMBER	No	This contains the ID of the sport to which respective user registered
4	CREATED_DT	TIMESTAMP (6)	No	This contains the date when the user registered to this sport

VOLLEYBALL_PLAYERS

This table stores the details of the Volleyball players. The data inserted into this table from the application when agent adds/edits the player details related to Volleyball. The list of attributes and significance of each attribute is as follows



The DDL Statement for VOLLEYBALL_PLAYERS is

S.No.	Attribute_Name	Data_Type	Nullable	Significance
1	ID	NUMBER	No	This attribute is used to uniquely identify a player from the table
2	FIRST_NAME	VARCHAR2(100 BYTE)	No	First Name of the player
3	LAST_NAME	VARCHAR2(100 BYTE)	No	Last Name of the Player
4	COUNTRY	NUMBER	No	The Id of the country to which the player belongs to
5	NO_OF_MATCHES	NUMBER	No	The total number of matches played by the player
6	BLOCKS	NUMBER	No	The total number of blocks by the player
7	ASSISTS	NUMBER	No	The total number of Assists by the player
8	POINTS_SHARED	NUMBER	No	The total number of points contributed by the player
9	NO_OF_SMASHES	NUMBER	No	The total number of smashes by the player

10	CREATED_BY	NUMBER	No	The Id of the Agent who added this player in the application
11	CREATED_DT	TIMESTAMP (6)	No	This refers to the date when the Player has been added by agent
12	UPDATED_BY	NUMBER	Yes	The Id of the Agent who edited the player details in application
13	UPDATED_DT	TIMESTAMP (6)	Yes	This refers to the date when the Player details has been last updated by an agent

5. Data Generation and Loading

We have used different methodologies for generating and loading the data., these are explained below.

5.1. Manual Data Generation and Loading

We have used www.generatedata.com to generate dummy data for some tables like “TOURNAMENTS” table. Below are the steps followed:

1. The random data required for the attributes “Name”, “Country” and “No_of_matches” in the tournaments table is generated by giving the required datatypes and values in the generatedata.com website.

The screenshot shows the generatedata.com interface for generating a dataset. The user has specified the following parameters:

- Dataset Name:** Your data set name here...
- Country-Specific Data:** All countries
- Data Set:**

Order	Column Title	Data Type	Examples	Options	Help	Del
1	NAME	Custom List	Company Names	Exactly 1 At Most 1 The U.S. Open The British Open Pebble Beach Pro	?	□
2	COUNTRY	Number Range	No examples available.	Between 1 and 197	?	□
3	NO_OF_MATCHES	Number Range	No examples available.	Between 5 and 30	?	□
- Export Types:** CSV, Excel, HTML, JSON, LDIF, Programming Language, SQL, XML
- Generate Options:** Generate 100 rows, Prompt to download, Zip? (checkboxes), Generate button

This generates the below excel file with 100 records for those 3 attributes

Final Project Report for Sports Aggregator Information System

The screenshot shows a Microsoft Excel spreadsheet with the following data:

	A	B	C
1	NAME	COUNTRY	NO_OF_MATCHES
81	PGA European Tour	44	16
82	The Presidents Cup	60	29
83	The Presidents Cup	93	18
84	The British Open	60	23
85	The British Open	66	6
86	Northern Trust (LA) Open	8	5
87	Pebble Beach Pro-Am	148	30
88	PGA Championship	11	13
89	The Players Championship	1	16
90	The Presidents Cup	86	9
91	The U.S. Open	35	25
92	The British Open	22	19
93	Pebble Beach Pro-Am	32	16
94	The British Open	156	26
95	Northern Trust (LA) Open	103	18
96	The British Open	40	15
97	The British Open	60	29
98	The Presidents Cup	100	15
99	Pebble Beach Pro-Am	182	17
100	Northern Trust (LA) Open	189	23
101	PGA European Tour	125	22

2. Now we have added some extra columns to the excel as per the attributes of the Tournaments table and saved it as a csv file.

The screenshot shows a Microsoft Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I	J	K
1	ID	SPORT_ID	NAME	COUNTRY	NO_OF_MATCHES	START_DATE	END_DATE	CREATED_BY	CREATED_DT	UPDATED_BY	UPDATED_DT
2	1	4	Northern Trust (LA) Open	188	27						
3	2	4	The British Open	180	15						
4	3	4	The U.S. Open	28	22						
5	4	4	PGA European Tour	179	6						
6	5	4	Northern Trust (LA) Open	5	28						
7	6	4	The U.S. Open	31	20						
8	7	4	The Players Championship	160	24						
9	8	4	The Players Championship	150	24						
10	9	4	PGA European Tour	118	11						
11	10	4	PGA European Tour	169	16						
12	11	4	The U.S. Open	2	10						
13	12	4	The U.S. Open	47	29						
14	13	4	The U.S. Open	114	5						
15	14	4	Northern Trust (LA) Open	185	17						
16	15	4	Pebble Beach Pro-Am	84	10						
17	16	4	The Presidents Cup	51	16						
18	17	4	Pebble Beach Pro-Am	69	11						
19	18	4	Pebble Beach Pro-Am	173	20						
20	19	4	PGA European Tour	128	8						
21	20	4	The Players Championship	131	10						
22	21	4	The U.S. Open	87	26						
	22	4	The Players Championship	9	25						

Final Project Report for Sports Aggregator Information System

3. Now we have disabled all the constraints on the Tournaments table. And then inserted the data using SQL developer.

The screenshot shows the Oracle SQL Developer interface. In the top right corner, the 'Tournaments' table is visible with its columns: ID, SPORT_ID, NAME, COUNTRY, NO_OF_MA..., START_DATE, END_DATE, CREATED_BY, CREATED_DT, and UP. A context menu is open over the table, with 'Import Data...' highlighted. Below the table, the 'Data Import Wizard - Step 1 of 5' window is open. It shows the 'Data Preview' tab selected, with the import file path set to 'C:\Users\Rohit\Downloads\XwmtQbxIMJtm-p4JEBF1p3dataNov-30-2017.csv'. The 'File Format' section includes options for Header (After Skip), Format (csv), Encoding (Cp1252), Delimiter (,), Left Enclosure ("), Skip Rows (0), Preview Row Limit (100), Line Terminator (standard: CR LF, CR or LF), and Right Enclosure ("). The 'File Contents' section displays the data from the CSV file:

ID	SPORT_ID	NAME	COUNTRY	NO_OF_MA...	START_DATE	END_DATE	CREATED_BY	CREATED_DT	UP
1	4	Northern Tr...	188	27					
2	4	The British ...	180	15					
3	4	The U.S. Open	28	22					
4	4	PGA Europe...	179	6					
5	4	Northern Tr...	5	28					
6	4	The U.S. Open	31	20					
7	4	The Players ...	160	24					
8	4	The Players ...	150	24					
9	4	PGA Europe...	118	11					

At the bottom of the 'Data Preview' window, there are buttons for Help, < Back, Next >, Finish, and Cancel.

Below the preview window, the 'Data Import Wizard - Step 2 of 4' window is open. It shows the 'Import Method' tab selected, with the 'Import Method' dropdown set to 'Insert', 'Table Name' set to 'TOURNAMENTS', and the 'Send Create Script to SQL Worksheet' checkbox checked. The 'File Contents' section displays the same data as above:

ID	SPORT_ID	NAME	COUNTRY	NO_OF_MA...	START_DATE	END_DATE	CREATED_BY	CREATED_DT	UP
1	4	Northern Tr...	188	27					
2	4	The British ...	180	15					
3	4	The U.S. Open	28	22					
4	4	PGA Europe...	179	6					
5	4	Northern Tr...	5	28					
6	4	The U.S. Open	31	20					
7	4	The Players ...	160	24					
8	4	The Players ...	150	24					
9	4	PGA Europe...	118	11					
10	4	PGA Europe...	169	16					
11	4	The U.S. Open	2	10					
12	4	The U.S. Open	47	29					
13	4	The U.S. Open	114	5					
14	4	Northern Tr...	185	17					

At the bottom of the 'Import Method' window, there are buttons for Help, < Back, Next >, Finish, and Cancel.

Final Project Report for Sports Aggregator Information System

Data Import Wizard - Step 3 of 5

Choose Columns

Select the columns to import from the data set and arrange them in the order you want.

Available Columns		Selected Columns	
ID	SPORT_ID	NAME	COUNTRY
NO_OF_MATCHES	START_DATE	END_DATE	CREATED_BY
CREATED_BY	CREATED_DT	UPDATED_BY	UPDATED_DT

File Contents

ID	SPORT_ID	NAME	COUNTRY	NO_OF_MA...	START_DATE	END_DATE	CREATED_BY	CREATED_DT	UP
1	4	Northern Tr...	188	27					
2	4	The British ...	180	15					
3	4	The U.S. Open	28	22					
4									

Help **< Back** **Next >** **Finish** **Cancel**

Data Import Wizard - Step 4 of 5

Column Definition

For each column in the Source Data Columns list on the left, select a Target Table column on the right.

Match By **Position**

Source Data Columns		Target Table Columns	
ID	Name	ID	Name
SPORT_ID	Country	Number	Size/Precision
NAME	NoOfMatches	0	Scale
COUNTRY	StartDate	-127	Nullable?
NO_OF_MATCHES	EndDate		Default
START_DATE	CreatedBy		Comment
END_DATE	CreatedDt		
CREATED_BY	UpdatedBy		
CREATED_DT	UpdatedDt		
UPDATED_BY			
UPDATED_DT			

Help **< Back** **Next >** **Finish** **Cancel**

Data Import Wizard - Step 5 of 5

Finish

Import Summary

- Destination Connection: group14
- Source File: C:\Users\Rohit\Downloads\XwmtQbxIMJtm-p4JEBF1p3dataNov-30-2017.csv
- Selected Fields
- Fields Not Selected
- Import Method: Insert

Save State

Help **< Back** **Next >** **Finish** **Cancel**

Final Project Report for Sports Aggregator Information System

Finally, the data is inserted into the tournaments table from csv file.

ID	SPORT_ID	NAME	COUNTRY	NO_OF_MATCHES	START_DATE	END_DATE	CREATED_BY	CREATED_DT
1	1	4 French Open	194	13 29-01-03		20-03-03	82 04-12-02 09:45:46.0000	
2	2	2 CONCACAF Gold Cup	130	17 28-01-03		28-03-03	59 04-12-02 10:57:37.0000	
3	3	4 French Open	106	9 22-01-03		04-03-03	15 06-12-02 09:49:33.0000	
4	4	9 PGA Championship	151	30 04-02-03		19-03-03	79 08-12-02 05:58:17.0000	
5	5	3 Adriatic League (ABA)	57	17 08-01-03		11-02-03	17 08-12-02 10:33:58.0000	
6	6	3 Russia's VTB United Le...	120	5 06-02-03		28-03-03	53 12-12-02 06:34:42.0000	
7	7	9 The Players Championship	177	28 06-02-03		06-04-03	50 15-12-02 11:36:16.0000	
8	8	9 Pebble Beach Pro-Am	38	18 14-02-03		27-03-03	50 24-12-02 02:54:55.0000	
9	9	9 Pebble Beach Pro-Am	116	28 19-01-03		01-03-03	87 26-12-02 08:53:38.0000	
10	10	1 Mexico International V...	131	8 02-03-03		10-04-03	55 01-01-03 05:03:38.0000	
11	11	4 Fed Cup	54	13 11-02-03		04-03-03	1 04-01-03 03:51:06.0000	
12	12	4 Australian Open	111	14 16-02-03		30-03-03	58 17-01-03 04:15:53.0000	
13	13	6 NHL Challenge	122	26 30-03-03		21-05-03	27 01-02-03 06:00:09.0000	
14	14	2 CONMEBOL Copa América	112	8 26-03-03		18-05-03	81 19-02-03 01:34:45.0000	
15	15	8 Emerald Coast Volleyba...	118	26 17-03-03		06-05-03	28 21-02-03 01:40:07.0000	
16	16	2 UEFA Champions League	100	30 06-04-03		28-04-03	81 27-02-03 09:59:37.0000	
17	17	1 Commonwealth Bank series	154	17 02-04-03		29-04-03	44 11-03-03 02:37:45.0000	
18	18	1 Border-Gavaskar trophy	23	12 29-04-03		28-06-03	95 12-03-03 10:05:12.0000	
19	19	9 The U.S. Open	124	6 21-04-03		05-06-03	80 17-03-03 10:23:00.0000	
20	20	6 Ninety Nine All Stars ...	159	29 15-05-03		11-07-03	63 23-03-03 05:34:31.0000	
21	21	9 Pebble Beach Pro-Am	177	17 15-05-03		21-06-03	37 28-03-03 12:01:08.0000	
22	22	8 Waupaca Boatride	118	15 20-04-03		17-05-03	28 29-03-03 11:31:29.0000	
23	23	9 Northern Trust (LA) Open	101	9 28-04-03		25-06-03	50 02-04-03 06:22:00.0000	
24	24	9 The British Open	161	5 02-06-03		19-07-03	79 05-04-03 05:16:17.0000	
25	25	8 Waupaca Boatride	181	7 21-05-03		09-07-03	84 09-04-03 04:29:53.0000	

- Now we have enabled all the constraints in the table.

5.2. Sequences

We have created sequences to insert the values of attribute ID in all the tables automatically incrementing by 1. Also, we followed naming convention for every sequence as “`<TABLE_NAME>_<ATTRIBUTE_NAME>_SEQ`”. For example, `USERS_ID_SEQ` is the sequence created and used to automatically insert a new record in `USERS` table with next ID value incremented by 1.

Below is the DDL statement for the `USERS_ID_SEQ`

```
CREATE SEQUENCE "DB626"."USERS_ID_SEQ" MINVALUE 1 MAXVALUE  
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE  
NOORDER NOCYCLE NOPARTITION;
```

Below is the Insert query used in the backend code of application to insert the new user record into users table during registration.

```
INSERT INTO USERS VALUES  
(USERS_ID_SEQ.NEXTVAL, ''+fname+'', ''+mname+'', ''+lname+'', ''+Address1+'', ''+Address2+''  
, ''+city+'', ''+state+'', (SELECT ID FROM COUNTRIES WHERE  
UPPER(COUNTRY)=UPPER(''+country+'')), ''+zipcode+'', '', ''+email+'', SYSDATE, SYSDATE)
```

5.3. Date Arithmetic

For generating random data for the attributes of “DATE”/”TIMESTAMP” datatype, we have executed below update queries to set date using date arithmetic.

UPDATE TOURNAMENTS

```
SET CREATED_DT=SYSDATE - DBMS_RANDOM.VALUE(0, 5475);
UPDATE TOURNAMENTS
SET START_DATE= CREATED_DT + DBMS_RANDOM.VALUE(20, 60);
UPDATE TOURNAMENTS
SET END_DATE= START_DATE + DBMS_RANDOM.VALUE(20, 60);
UPDATE TOURNAMENTS
SET UPDATED_DT= CREATED_DT + DBMS_RANDOM.VALUE(0, 30);
```

5.4. DBMS_RANDOM

DBMS_RANDOM package with a variety of methods and parameter settings for generating both numbers and strings. It is an in-built package provided in ORACLE DBMS. Here we are using this package for generating data to the “CONTACT” attribute.

SQL QUERY:

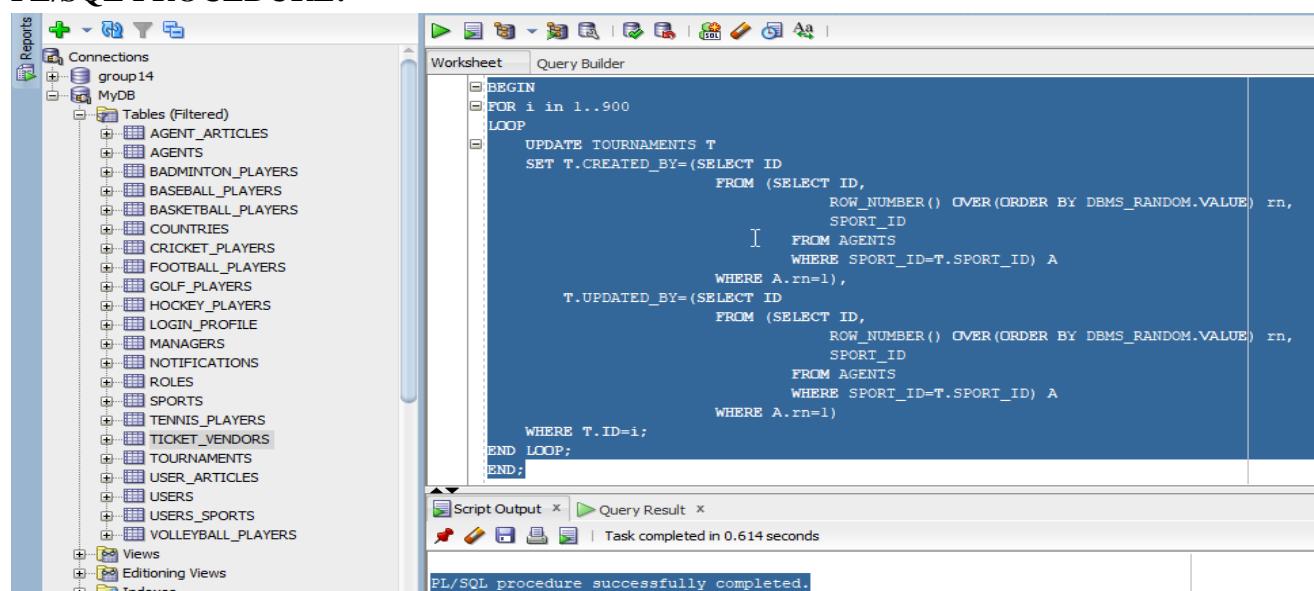
```
UPDATE MANAGERS SET CONTACT = '(' ||
TO_CHAR(TRUNC(DBMS_RANDOM.VALUE(100,999))) || ')' ||
TO_CHAR(TRUNC(DBMS_RANDOM.VALUE(000,999)), 'fm099') || '-' ||
TO_CHAR(TRUNC(DBMS_RANDOM.VALUE(0000,9999)), 'fm0999');
```

The above update query updates the contact details of Manager with the random values generated by DBMS_RANDOM.VALUE from the given ranges. Example: (813)-340-2610

5.5. PROCEDURE

We have created an SQL Block for updating the “CREATED_BY” and “UPDATED_BY” attributes in the “TOURNAMENTS” table with the random values of ID’s of Agents who are related to the same sport of the tournament.

PL/SQL PROCEDURE:



```
BEGIN
FOR i IN 1..900
LOOP
    UPDATE TOURNAMENTS T
    SET T.CREATED_BY=(SELECT ID
                      FROM (SELECT ID,
                                   ROW_NUMBER() OVER(ORDER BY DBMS_RANDOM.VALUE) rn,
                                   SPORT_ID
                            FROM AGENTS
                           WHERE SPORT_ID=T.SPORT_ID) A
                     WHERE A.rn=1),
        T.UPDATED_BY=(SELECT ID
                      FROM (SELECT ID,
                                   ROW_NUMBER() OVER(ORDER BY DBMS_RANDOM.VALUE) rn,
                                   SPORT_ID
                            FROM AGENTS
                           WHERE SPORT_ID=T.SPORT_ID) A
                     WHERE A.rn=1)
    WHERE T.ID=i;
END LOOP;
END;
```

The above PL/SQL procedure executed successfully and updated the created_by and updated_by attributes of tournaments table with random values of respective sport Agent IDs for all the records with tournament ID from 1 to 900.

6. Query Writing

6.1. Application Initiated Queries

We have used lot of queries that were initiated from the application. Out of those, here we are demonstrating for the Registration and Login Pages of the application.

6.1.1. Registration Page

In this page, when the user fills all the mandatory fields and clicks on Register button then

- First it will check whether the USERNAME given by the user is already exists in the database or not, by executing below SELECT query.

Query:

```
SELECT COUNT(DISTINCT USERNAME) AS COUNT FROM LOGIN_PROFILE WHERE  
UPPER(USERNAME)=UPPER('"+username+"')
```

If the COUNT is not equal to 0 then it throws error message to the User as “Username already exists. Please give another username.”.

- Else then it starts inserting the User details into 3 tables (USERS, LOGIN_PROFILE and USERS_SPORTS) by executing the below Insert queries.

Query 1:

```
INSERT INTO USERS VALUES  
(USERS_ID_SEQ.NEXTVAL, '"+fname+"','"+mname+"','"+lname+"','"+Address1+"','"+Address2+"','"+city+"','"+state+"',(SELECT ID FROM COUNTRIES WHERE  
UPPER(COUNTRY)=UPPER('"+country+"')), '"+zipcode+"','"+email+"', SYSDATE,  
SYSDATE)
```

Query 2:

```
INSERT INTO LOGIN_PROFILE VALUES ('"+username+"','"+password+"',  
USERS_ID_SEQ.CURRVAL, (SELECT ID FROM ROLES WHERE UPPER ROLE='USERS'), 'LoggedIn  
Successfully', SYSDATE, 0)
```

Query 3:

```
INSERT INTO USERS_SPORTS VALUES (USERS_SPORTS_ID_SEQ.NEXTVAL,  
USERS_ID_SEQ.CURRVAL, (SELECT ID FROM SPORTS WHERE UPPER NAME=UPPER('"+item+"')),  
SYSDATE)
```

Backend C# Code:

```
protected void CreateUser_Click(object sender, EventArgs e)  
{  
    string fname = FirstName.Text;  
    string mname = MiddleName.Text;  
    string lname = LastName.Text;
```

```

        string Address1 = AddressLine1.Text;
        string Address2 = AddressLine2.Text;
        string city = City.Text;
        string state = State.Text;
        int zipcode = Convert.ToInt32(ZipCode.Text);
        string country = Country.Text;
        string email = Email.Text;
        string username = Username.Text;
        string password = Password.Text;
        OracleConnection conn = new OracleConnection("Data Source=MyDB;User
Id=DB626;Password=*****");
        conn.Open();

        string sql = "SELECT COUNT(DISTINCT USERNAME) AS COUNT FROM LOGIN_PROFILE
WHERE UPPER(USERNAME)=UPPER('" + username + "')";
        OracleCommand cmd = new OracleCommand(sql, conn);
        OracleDataReader dr = cmd.ExecuteReader();

        dr.Read();
        int count = Convert.ToInt32(dr["COUNT"].ToString());
        if(count==0)
        {
            string sql1="INSERT INTO USERS VALUES
(USERS_ID_SEQ.NEXTVAL,'" + fname + "','" + mname + "','" + lname + "','" + Address1 + "','" + Address2 +
',' + city + "','" + state + ',(SELECT ID FROM COUNTRIES WHERE
UPPER(COUNTRY)=UPPER('' + country + '')),'' + zipcode + '' ,'' +email + '' ,SYSDATE,SYSDATE)';
            OracleCommand cmd1 = new OracleCommand(sql1, conn);
            cmd1.ExecuteNonQuery();

            string sql2 = "INSERT INTO LOGIN_PROFILE VALUES ('" + username + "','" +
password + "','" + USERS_ID_SEQ.CURRVAL,(SELECT ID FROM ROLES WHERE
UPPER ROLE='USERS'),'LoggedIn Successfully',SYSDATE,0)";
            OracleCommand cmd2 = new OracleCommand(sql2, conn);
            cmd2.ExecuteNonQuery();

            foreach (ListItem item in Sports.Items)
            {
                if (item.Selected == true)
                {
                    string sql3 = "INSERT INTO USERS_SPORTS VALUES
(USERS_SPORTS_ID_SEQ.NEXTVAL,USERS_ID_SEQ.CURRVAL,(SELECT ID FROM SPORTS WHERE
UPPER(NAME)=UPPER('' + item + '')),SYSDATE)";
                    OracleCommand cmd3 = new OracleCommand(sql3, conn);
                    cmd3.ExecuteNonQuery();
                }
            }
            ErrorTextBox.Visible = false;
            Response.Redirect("~/Account/Login");
        }
        else
        {
            ErrorTextBox.Visible = true;
        }
        dr.Dispose();
        conn.Dispose();
    }
}

```

6.1.2. Login Page

Here when the user enters his username and password details and clicks on Log in button then

- First, Initially it checks whether the entered USERNAME exists in database or not, by executing the below SELECT query.

Query:

```
SELECT COUNT(DISTINCT USERNAME) AS COUNT FROM LOGIN_PROFILE WHERE  
UPPER(USERNAME)=UPPER('" + username + "')
```

If COUNT=0 then it throws an error message to user saying “The Username does not exists, please try with correct Username”.

- Else, then it checks whether the password entered is matching with the password of the respective Username or not, by executing the below SELECT query.

Query:

```
SELECT USERNAME, PASSWORD, ROLE_ID FROM LOGIN_PROFILE WHERE  
UPPER(USERNAME)=UPPER('" + username + "')
```

If password didn't match then it throws another error message to user as “The Username/password is incorrect, please try with correct login details.”

- Else, then it updates the LOGINSTATUS, LASTLOGINDATE and NO_OF_LOGINS attributes in LOGIN_PROFILE table for that USERNAME record by executing below UPDATE query and navigates to their respective Homepage based on Role_Id's.

Query:

```
UPDATE LOGIN_PROFILE SET NO_OF_LOGINS=NO_OF_LOGINS+1, LASTLOGINDATE=SYSDATE,  
LOGINSTATUS='LoggedIn Successfully' WHERE UPPER(USERNAME)=UPPER('" + username + "')
```

Backend C# Code:

```
protected void LogIn(object sender, EventArgs e)  
{  
    string username = Username.Text;  
    string password = Password.Text;  
  
    ErrorTextBox1.Visible = false;  
    ErrorTextBox2.Visible = false;  
    OracleConnection conn = new OracleConnection("Data Source=MyDB;User  
Id=DB626;Password=*****");  
    conn.Open();  
    string sql = "SELECT COUNT(DISTINCT USERNAME) AS COUNT FROM LOGIN_PROFILE  
WHERE UPPER(USERNAME)=UPPER('" + username + "');";  
    OracleCommand cmd = new OracleCommand(sql, conn);  
    OracleDataReader dr = cmd.ExecuteReader();  
    dr.Read();  
    int count = Convert.ToInt32(dr["COUNT"].ToString());  
    if(count==0)  
    {  
        ErrorTextBox1.Visible = true;  
    }  
    else  
    {  
        string sql1 = "SELECT USERNAME,PASSWORD,ROLE_ID FROM LOGIN_PROFILE  
WHERE UPPER(USERNAME)=UPPER('" + username + "');";  
        OracleCommand cmd1 = new OracleCommand(sql1, conn);  
        OracleDataReader dr1 = cmd1.ExecuteReader();  
        dr1.Read();
```

```
        string password_db = dr1["PASSWORD"].ToString();
        if(password==password_db)
        {
            string sql2 = "UPDATE LOGIN_PROFILE SET
NO_OF_LOGINS=NO_OF_LOGINS+1, LASTLOGINDATE=SYSDATE, LOGINSTATUS='LoggedIn Successfully'
WHERE UPPER(USERNAME)=UPPER('" + username + "');");
            OracleCommand cmd2 = new OracleCommand(sql2, conn);
            cmd2.ExecuteNonQuery();
            Session["UserName"] = Username.Text;
            FormsAuthentication.SetAuthCookie(Username.Text, false);

            if (dr1["ROLE_ID"].ToString() == "3")
            {
                Response.Redirect("~/Users/HomePage.aspx");
            }
            else if(dr1["ROLE_ID"].ToString() == "2")
            {
                Response.Redirect("~/Agents/HomePage.aspx");
            }
        }
        else
        {
            ErrorTextBox2.Visible = true;
        }
    }
}
```

6.2. PL/SQL Package

When agent perform add/edit player details in the application, for Inserting/updating those Player details in database we have created PL/SQL packages which will be initiated from the application. For instance, we are showing it using the “**BASEBALL_DATA_PKG**” package created for insert/update the baseball player details. Generally, a package consists of two parts one is **SPECIFICATION** and the other is **BODY**.

6.2.1. BASEBALL_DATA_PKG SPECIFICATION

In this part we just declare the variables, functions, procedures that will be used to implement the package. These declared items can be used anywhere within the package or any other packages.

```

CREATE OR REPLACE PACKAGE BASEBALL_DATA_PKG AS
    FUNCTION SEQ_NEXTVAL_ON_DEMAND (SEQ_NAME_IN IN VARCHAR2) RETURN NUMBER;
    PROCEDURE BASEBALL_DATA_PRC (FIRST_NAME_BB IN VARCHAR2,
                                  LAST_NAME_BB IN VARCHAR2,
                                  COUNTRY_NAME_BB IN VARCHAR2,
                                  AT_BATS_BB IN NUMBER,
                                  RUNS_BB IN NUMBER,
                                  HITS_BB IN NUMBER,
                                  RUNS_BATTED_IN_BB IN NUMBER,
                                  BASE_ON_BALLS_BB IN NUMBER,
                                  BATTING_AVG_BB IN NUMBER,
                                  AGENT_USERNAME IN VARCHAR2);
END BASEBALL_DATA_PKG;
  
```

Script Output | Task completed in 0.299 seconds

Package BASEBALL_DATA_PKG compiled

6.2.2. BASEBALL_DATA_PKG BODY

In this part, we have the implementation of the functions and procedures that declared in Specification part.

This Package consists of a function SEQ_NEXTVAL_ON_DEMAND and a procedure BASEBALL_DATA_PRC.

6.2.2.1. SEQ_NEXTVAL_ON_DEMAND FUNCTION

This function SEQ_NEXTVAL_ON_DEMAND is used to get the next value from the sequence BASEBALL_ID_SEQ for ID attribute while inserting a new record into the BASEBALL_PLAYERS table. We can use the BASEBALL_ID_SEQ.NEXTVAL directly in the insert statement but there is an issue while using SEQ.NEXTVAL in the merge statements, it updates to next value for both Match and Not Match. To overcome this, we have created a separate function which returns the next value of the sequence when called.

Input Parameters: Sequence Name (In this case it is BASEBALL_ID_SEQ)

Output Parameters: Next value of the input sequence

PL/SQL Block:

```

FUNCTION SEQ_NEXTVAL_ON_DEMAND (SEQ_NAME_IN IN VARCHAR2)
    RETURN NUMBER
IS
    SEQ_VAL NUMBER;
BEGIN
    EXECUTE IMMEDIATE 'SELECT ' || SEQ_NAME_IN || '.NEXTVAL FROM DUAL'
        INTO SEQ_VAL;
    RETURN SEQ_VAL;
END SEQ_NEXTVAL_ON_DEMAND;
  
```

6.2.2.2. BASEBALL_DATA_PRC PROCEDURE

This is the procedure that is being called from application when agent perform Add/edit the Baseball players details. Initially it gets the ID values for the country and the agent by using the input arguments COUNTRY_NAME_BB and AGENT_USERNAME respectively from the tables COUNTRIES and LOGIN_PROFILE respectively. And then later it checks whether the player already exists in the BASEBALL_PLAYER table or not based on the attributes FIRST_NAME, LAST_NAME and COUNTRY_ID. If there is any match then it updates the data for that record else it inserts a new record into the table.

Input Parameters: FIRST_NAME_BB, LAST_NAME_BB, COUNTRY_NAME_BB, AT_BATS_BB, RUNS_BB, HITS_BB, RUNS_BATTED_IN_BB, BASE_ON_BALLS_BB, BATTING_AVG_BB, AGENT_USERNAME

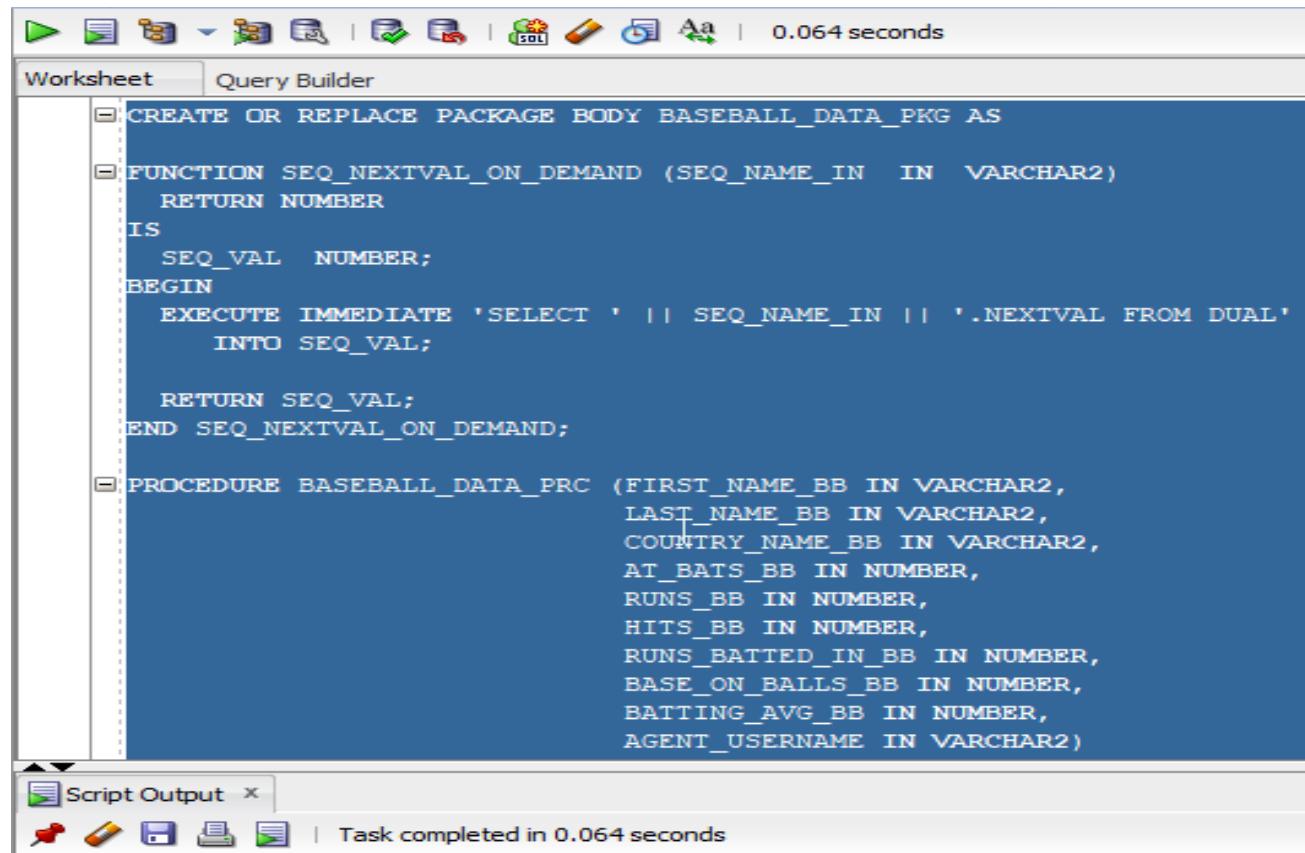
PL/SQL Block:

```
PROCEDURE BASEBALL_DATA_PRC (FIRST_NAME_BB IN VARCHAR2,
                             LAST_NAME_BB IN VARCHAR2,
                             COUNTRY_NAME_BB IN VARCHAR2,
                             AT_BATS_BB IN NUMBER,
                             RUNS_BB IN NUMBER,
                             HITS_BB IN NUMBER,
                             RUNS_BATTED_IN_BB IN NUMBER,
                             BASE_ON_BALLS_BB IN NUMBER,
                             BATTING_AVG_BB IN NUMBER,
                             AGENT_USERNAME IN VARCHAR2)
IS
    COUNTRY_ID NUMBER;
    AGENT_ID NUMBER;
BEGIN
    BEGIN
        SELECT ID INTO COUNTRY_ID FROM COUNTRIES WHERE
UPPER(COUNTRY)=UPPER(COUNTRY_NAME_BB);
        EXCEPTION
        WHEN NO_DATA_FOUND
        THEN
            RAISE_APPLICATION_ERROR(-20250,'An error was encountered'||sqlcode||'-error-'||sqlerrm);
    END;
    BEGIN
        SELECT USERID INTO AGENT_ID FROM LOGIN_PROFILE WHERE
UPPER(USERNAME)=UPPER(AGENT_USERNAME) AND ROLE_ID=2;
        EXCEPTION
        WHEN NO_DATA_FOUND
        THEN
            RAISE_APPLICATION_ERROR(-20250,'An error was encountered'||sqlcode||'-error-'||sqlerrm);
    END;
MERGE INTO BASEBALL_PLAYERS BP
USING (SELECT FIRST_NAME_BB FNB,
             LAST_NAME_BB LNB,
             AGENT_USERNAME AU
        FROM DUAL
       WHERE UPPER(AGENT_USERNAME) IN (SELECT UPPER(USERNAME) FROM
LOGIN_PROFILE)) BA
ON (BA.FNB=BP.FIRST_NAME AND BA.LNB=BP.LAST_NAME AND COUNTRY_ID=BP.COUNTRY)
WHEN MATCHED THEN
UPDATE SET AT_BATS=AT_BATS_BB,
```

Final Project Report for Sports Aggregator Information System

```
RUNS=RUNS_BB,
HITS=HITS_BB,
RUNS_BATTED_IN=RUNS_BATTED_IN_BB,
BASE_ON_BALLS=BASE_ON_BALLS_BB,
BATTING_AVG=BATTING_AVG_BB,
UPDATED_BY=AGENT_ID,
UPDATED_DT=SYSDATE
WHEN NOT MATCHED THEN
INSERT VALUES(SEQ_NEXVAL_ON_DEMAND('baseball_players_id_seq'),
BA.FNB,
BA.LNB,
COUNTRY_ID,
AT_BATS_BB,
RUNS_BB,
HITS_BB,
RUNS_BATTED_IN_BB,
BASE_ON_BALLS_BB,
BATTING_AVG_BB,
AGENT_ID,
SYSDATE,
null,
null);
EXCEPTION
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20250,'An error was encountered'||sqlcode||'-error-'||sqlerrm);
END BASEBALL_DATA_PRC;
```

Now, by combining this function and procedure we get the final package body for the BASEBALL_DATA_PKG.



The screenshot shows the Oracle SQL Developer interface with the 'Worksheet' tab selected. The code editor displays the package body for 'BASEBALL_DATA_PKG'. The code includes a function 'SEQ_NEXVAL_ON_DEMAND' that returns the next value from a sequence based on its name. It uses dynamic SQL to select the next value from the DUAL table. The package also contains a procedure 'BASEBALL_DATA_PRC' that takes several parameters (FIRST_NAME_BB, LAST_NAME_BB, COUNTRY_NAME_BB, AT_BATS_BB, RUNS_BB, HITS_BB, RUNS_BATTED_IN_BB, BASE_ON_BALLS_BB, BATTING_AVG_BB, AGENT_USERNAME) and inserts them into a table. The bottom status bar indicates that the package body has been compiled successfully.

```
CREATE OR REPLACE PACKAGE BODY BASEBALL_DATA_PKG AS
FUNCTION SEQ_NEXVAL_ON_DEMAND (SEQ_NAME_IN IN VARCHAR2)
RETURN NUMBER
IS
SEQ_VAL NUMBER;
BEGIN
EXECUTE IMMEDIATE 'SELECT ' || SEQ_NAME_IN || '.NEXTVAL FROM DUAL'
INTO SEQ_VAL;

RETURN SEQ_VAL;
END SEQ_NEXVAL_ON_DEMAND;

PROCEDURE BASEBALL_DATA_PRC (FIRST_NAME_BB IN VARCHAR2,
LAST_NAME_BB IN VARCHAR2,
COUNTRY_NAME_BB IN VARCHAR2,
AT_BATS_BB IN NUMBER,
RUNS_BB IN NUMBER,
HITS_BB IN NUMBER,
RUNS_BATTED_IN_BB IN NUMBER,
BASE_ON_BALLS_BB IN NUMBER,
BATTING_AVG_BB IN NUMBER,
AGENT_USERNAME IN VARCHAR2)
```

Script Output | Task completed in 0.064 seconds

Package Body BASEBALL_DATA_PKG compiled

6.2.3. EXECUTION OF PACKAGE

Here I am going to demonstrate the various scenarios in execution of package that takes place when triggered from application using SQL developer.

Initially, there are no records in the BaseBall_Players as shown below.

Worksheet | Query Builder

```
SELECT * FROM BASEBALL_PLAYERS;
```

Script Output | Query Result | All Rows Fetched: 0 in 0 seconds

ID	FIRST_NAME	LAST_NAME	COUNTRY	AT_BATS	RUNS	HITS	RUNS_AVG	BASE_ON_BALLS	BATTING_AVG	CREATED_BY	CREATED_DT	UPDATED_BY	UPDATED_DT

INSERT DATA USING PACKAGE:

Now I am executing the package to insert a new record into the table.

Worksheet | Query Builder

```
EXEC BASEBALL_DATA_PKG.BASEBALL_DATA_PRC('ROHITH','MOVVA','India',425,95,134,87,68,.319,'agent123');
```

Script Output | Query Result | Task completed in 0.411 seconds

PL/SQL procedure successfully completed.

Now it has inserted a new record into the table as shown below by fetching the ID values of country ‘India’ and agent ‘agent123’, updated those values for that record for attributes COUNTRY and CREATED_BY.

Worksheet | Query Builder

```
SELECT * FROM BASEBALL_PLAYERS;
```

Script Output | Query Result | All Rows Fetched: 1 in 0.02 seconds

ID	FIRST_NAME	LAST_NAME	COUNTRY	AT_BATS	RUNS	HITS	RUNS_AVG	BASE_ON_BALLS	BATTING_AVG	CREATED_BY	CREATED_DT	UPDATED_BY	UPDATED_DT
1	ROHITH	MOVVA	India	425	95	134	0.319	87	0.525	agent123	2012-12-17 12:26:08....	(null)	(null)

UPDATE DATA USING PACKAGE:

Here I am going to run the same package to update the RUNS, BATTING_AVG attributes for the previously inserted record from 95 to 123 and 0.319 to 0.525 respectively.

Final Project Report for Sports Aggregator Information System

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes icons for file, edit, search, and database operations. The toolbar below has icons for running scripts, saving, and zooming. The main area is titled "Worksheet" and contains the SQL command: `EXEC BASEBALL_DATA_PKG.BASEBALL_DATA_PRC('ROHITH','MOVVA','India',425,123,134,87,.68,.525,'agent123');`. Below the worksheet, the status bar indicates "0.066 seconds". The bottom pane, titled "Query Result", shows the message "PL/SQL procedure successfully completed." twice.

Now, it has updated the data for Runs and Batting_Avg attributes along with the UPDATED_BY and UPDATED_DT attributes.

The screenshot shows the Oracle SQL Developer interface. The top menu bar and toolbar are visible. The main area contains the SQL query: `SELECT * FROM BASEBALL_PLAYERS;`. The bottom pane, titled "Query Result", displays the results of the query. The table structure includes columns: ID, FIRST_NAME, LAST_NAME, COUNTRY, AT_BATS, RUNS, HITS, RUNS_B..., BASE..., BATTI..., UPDATED_BY, UPDATED_DT, CREATED_BY, and CREATE. One row is shown with values: 1, ROHITH, MOVVA, 76, 425, 123, 134, 87, 68, 0.525, 201-12-17 01:08:13.0000..., 201-12-17.

Inserted one more record to check whether the function in package returning correct sequence value or not.

The screenshot shows the Oracle SQL Developer interface. The top menu bar and toolbar are visible. The main area contains the SQL command: `EXEC BASEBALL_DATA_PKG.BASEBALL_DATA_PRC('SRINIVAS','RAJU','China',400,100,200,79,62,.467,'agent123');`. Below the worksheet, the status bar indicates "0.05 seconds". The bottom pane, titled "Query Result", shows the message "PL/SQL procedure successfully completed." twice.

The screenshot shows the Oracle SQL Developer interface. The top menu bar and toolbar are visible. The main area contains the SQL query: `SELECT * FROM BASEBALL_PLAYERS;`. The bottom pane, titled "Query Result", displays the results of the query. The table structure includes columns: ID, FIRST_NAME, LAST_NAME, COUNTRY, AT_BATS, RUNS, HITS, RUNS_B..., BASE..., BATTI..., CREATED_BY, CREATED_DT, UPDATED_BY, and UPDATED_DT. Two rows are shown: Row 1 with values 1, ROHITH, MOVVA, 76, 425, 123, 134, 87, 68, 0.525, 201-12-17 01:07:38...., 201-12-17 01, and Row 2 with values 2, SRINIVAS, RAJU, 36, 400, 100, 200, 79, 62, 0.467, 201-12-17 01:52:43...., (null).

7. Performance Tuning

7.1. Function-Based Index Experiment

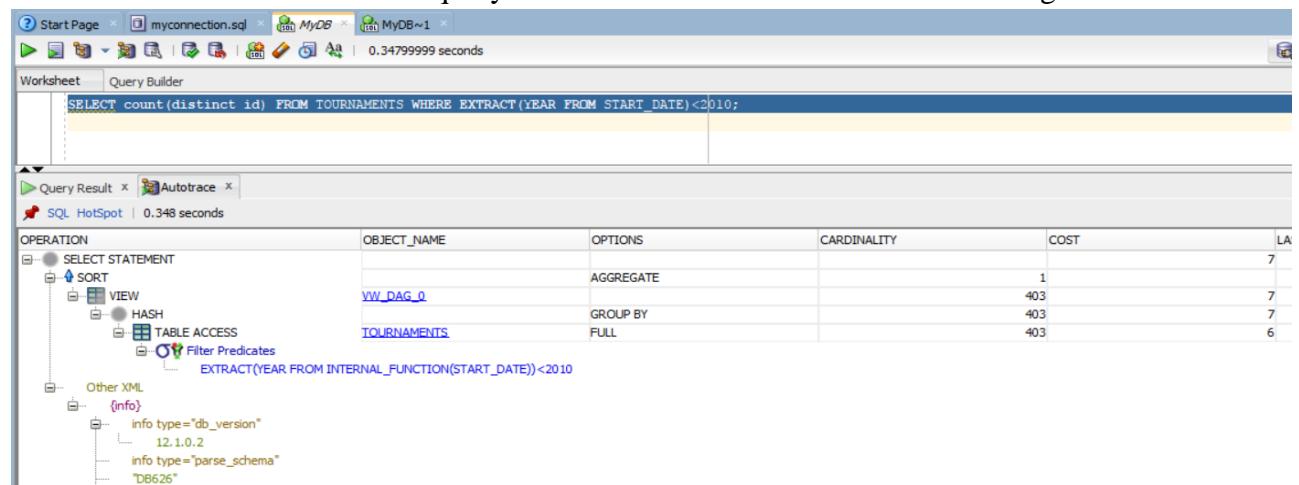
The following query returns the count of distinct tournaments which were started before the year 2010 from the “TOURNAMENTS” table. Here we have used a function named “EXTRACT” to extract only the year from the complete date format wherein it was having day and month along with year. It is imperative for us to use the function “Extract” rather than “Where” condition alone as we cannot filter out only the year from the date format data.

```
SELECT count(distinct id) FROM TOURNAMENTS WHERE EXTRACT(YEAR  
FROM START_DATE)<2010;
```

Total Cost of Execution = $7+7+7+6=27$

Average Execution time taken for 10 observations: 0.35 seconds

The below screenshot shows the query result when it is executed without creating an index.



Creating a function based Index

```
CREATE INDEX TOURNAMEANTS_FN_IDX ON TOURNAMEANTS(EXTRACT(YEAR  
FROM START_DATE));
```

Below screenshot shows the creation of index

Final Project Report for Sports Aggregator Information System

The screenshot shows the Oracle SQL Developer interface. In the top tab bar, there are tabs for 'Start Page', 'myconnection.sql', 'MyDB', and 'MyDB~1'. The main workspace shows a query in the 'Worksheet' tab:

```
CREATE INDEX TOURNAMENTS_FN_IDX ON TOURNAMENTS(EXTRACT(YEAR FROM START_DATE));
```

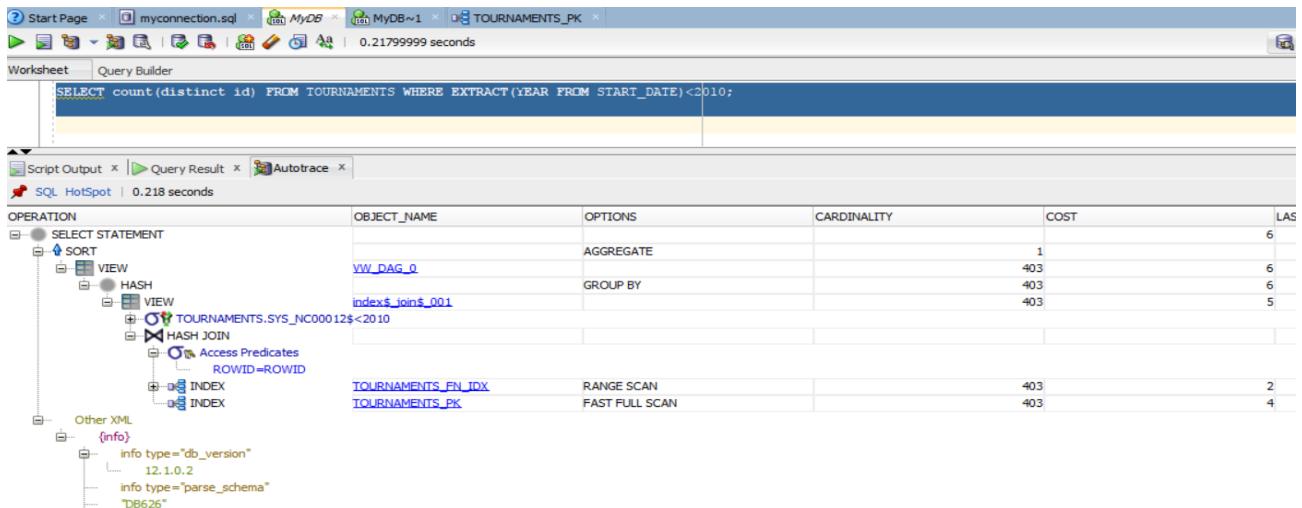
Below the workspace, the 'Script Output' tab displays the result of the execution:

```
Index TOURNAMENTS_FN_IDX created.
```

Now we have run the same query again after creation of index as below:

Here the total cost of execution = $6+6+6+2=20$.

Average Execution time taken for 10 observations: 0.15 seconds



Conclusion

When the index is not created the query is executed by performing a full table scan. But after creating the index, the index is used to process the query hence the execution time and cost are reduced and performed a range scan instead of full scan by using function based index. In our database/information system the business use of this index is to filter out the tournaments according to the year they have started.

7.2. Transaction Level Read consistency

Here we have used “READ ONLY” clause which establishes the current transaction as a read-only transaction. This provides us **Transaction-Level Read Consistency**.

For conducting this experiment, we have opened two SQL developer windows which I am calling as session 1 and session 2.

In session 1, we have committed the statements till now and below are the screenshots of the same.

Final Project Report for Sports Aggregator Information System

The screenshot shows the MySQL Workbench interface. The top tab bar has tabs for 'Start Page', 'myconnection.sql', 'MyDB', 'MyDB~1', and 'LOGIN_PROFILE'. The 'Worksheet' tab is selected, displaying the SQL command: `COMMIT;--Committing all database changes before setting up transaction`. Below the worksheet is a 'Script Output' window showing the message: 'Task completed in 0.06 seconds'. At the bottom left, the status bar says 'Commit complete.'

In session 2, we have tried to conduct an experiment to retrieve the users who have number of logins greater than 20 times. Below is the screenshot showing the same. We have three users who have more than 20 logins.

The screenshot shows the MySQL Workbench interface. The 'Worksheet' tab is selected, displaying the following SQL query:

```
SELECT * FROM USERS WHERE ID IN (SELECT USERID
                                  FROM LOGIN_PROFILE
                                 WHERE NO_OF_LOGINS > 20
                                   AND ROLE_ID=3);
```

Below the query is a 'Query Result' window showing the retrieved data:

ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	STATE	COUNTRY	ZIPCODE	CONTACT	EMAIL_ID	CREATED_DT
1	3 Srinivas	Raju	Nanapuram	14202, Fairway Oaks	Apt No : 202	Tampa	Florida	5	33613 (null)	srinivasraju@gmail.com	14-NOV-17 04.11	
2	4 Venkata	Ravi Teja	Lanka	Ashford Green	Flat No: G	Hyderabad	Telangana	1	500035 (null)	raviteja@gmail.com	15-NOV-17 02.54	
3	5 Rohit	(null)	Movva	4209 Winding Moss Trail, Apt No:105	TAMPA	Florida		4	33613 (null)	rohit1279@gmail.com	15-NOV-17 03.35	

Again, in session 2, we have tried to run a query to display the users along with their number of logins. Below is the screenshot for the same.

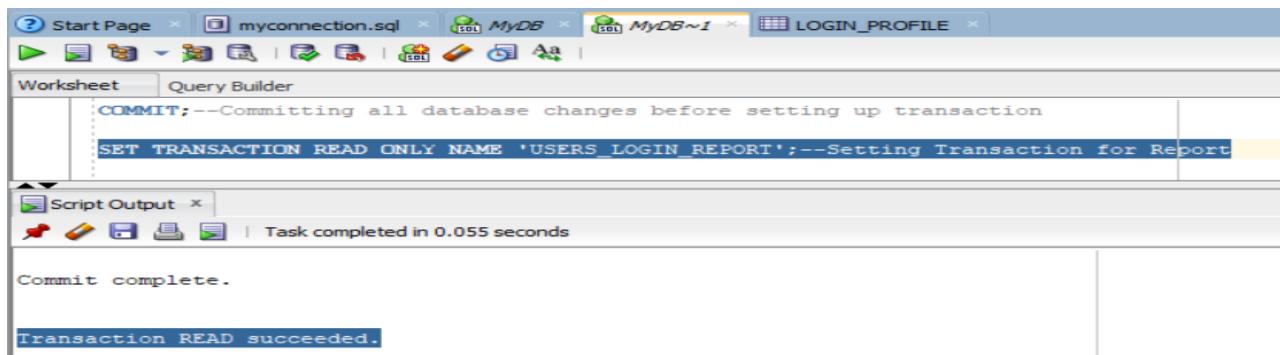
The screenshot shows the MySQL Workbench interface. The 'Worksheet' tab is selected, displaying the SQL command: `SELECT * FROM LOGIN_PROFILE;` Below the query is a 'Query Result' window showing the data from the 'LOGIN_PROFILE' table:

USERNAME	PASSWORD	USERID	ROLE_ID	LOGINSTATUS	LASTLOGINDATE	NO_OF_LOGINS
1 movar	Infra@2017	10	3	LoggedIn Successfully	18-NOV-17 06.07.40.000000000 PM	12
2 agent123	agent123	2	2	LoggedIn Successfully	28-NOV-17 11.05.32.000000000 AM	5
3 movvar	Infra@2017	2	3	LoggedIn Successfully	14-NOV-17 04.05.37.000000000 PM	13
4 srini	Pass@1234	3	3	LoggedIn Successfully	14-NOV-17 04.11.35.000000000 PM	25
5 lankav	Civil@123	4	3	LoggedIn Successfully	15-NOV-17 02.54.22.000000000 PM	33
6 MOVVAR123	Infra@2017	5	3	LoggedIn Successfully	15-NOV-17 03.35.09.000000000 PM	24
7 movvar1279	Infra@1279	6	3	LoggedIn Successfully	29-NOV-17 12.53.25.000000000 PM	20
8 srikan	Infra@1279	7	3	LoggedIn Successfully	15-NOV-17 04.00.08.000000000 PM	8
9 rohit1279	Infra@1234	8	3	LoggedIn Successfully	15-NOV-17 04.42.56.000000000 PM	13
10 sagds	Rohit1279	9	3	LoggedIn Successfully	15-NOV-17 04.49.58.000000000 PM	16

Now transaction is set in another session:

Now in session 1, we have set the read only transaction with name "USER_LOGIN_REPORT"

Final Project Report for Sports Aggregator Information System



Start Page myconnection.sql MyDB MyDB~1 LOGIN_PROFILE

Worksheet Query Builder

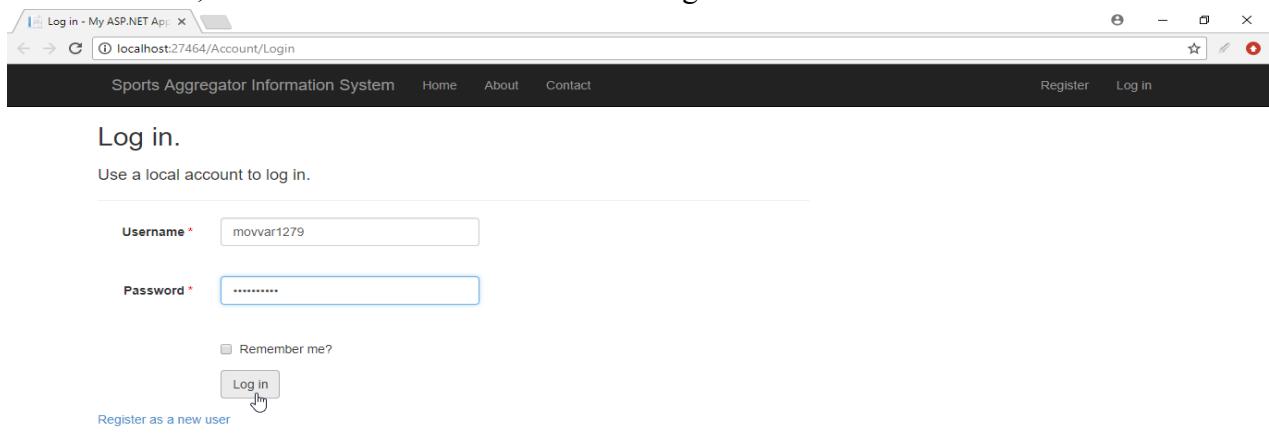
```
COMMIT;--Committing all database changes before setting up transaction
SET TRANSACTION READ ONLY NAME 'USERS_LOGIN_REPORT';--Setting Transaction for Report
```

Script Output Task completed in 0.055 seconds

Commit complete.

Transaction READ succeeded.

Now we have gone to our information system and tried to login using the username “movvar1279”, which will increase the number of login count from 20 to 21.



Log in - My ASP.NET App

localhost:27464/Account/Login

Sports Aggregator Information System Home About Contact Register Log in

Log in.

Use a local account to log in.

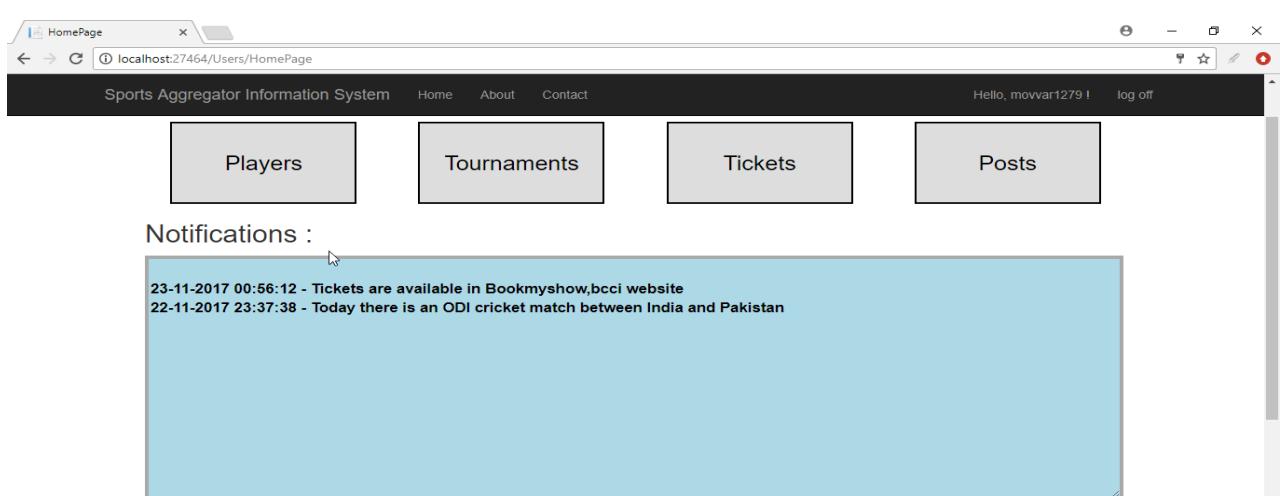
Username *

Password *

Remember me?

[Register as a new user](#)

© 2017 - Group 3 Sports Web Application



HomePage

localhost:27464/Users/HomePage

Sports Aggregator Information System Home About Contact Hello, movvar1279 ! log off

Players Tournaments Tickets Posts

Notifications :

23-11-2017 00:56:12 - Tickets are available in Bookmyshow,bcci website
22-11-2017 23:37:38 - Today there is an ODI cricket match between India and Pakistan

Articles :

Now in session 2, we have again tried to conduct an experiment to retrieve the users who have number of logins greater than 20 times. Below is the screenshot showing the same. Now, We have four users instead of 3 users who have more than 20 logins.

Final Project Report for Sports Aggregator Information System

Session 1 (Top):

```
SELECT * FROM LOGIN_PROFILE;
```

USERNAME	PASSWORD	USERID	ROLE_ID	LOGINSTATUS	LASTLOGINDATE	NO_OF_LOGINS
1 movar	Infra@2017	10	3	LoggedIn Successfully	18-NOV-17 06.07.40.000000000 PM	12
2 agent123	agent123	2	2	LoggedIn Successfully	28-NOV-17 11.05.32.000000000 AM	5
3 movvar	Infra@2017	2	3	LoggedIn Successfully	14-NOV-17 04.05.37.000000000 PM	13
4 sriini	Pass@1234	3	3	LoggedIn Successfully	14-NOV-17 04.11.35.000000000 PM	25
5 lankav	Civil@123	4	3	LoggedIn Successfully	15-NOV-17 02.54.22.000000000 PM	33
6 MOVVAR123	Infra@2017	5	3	LoggedIn Successfully	15-NOV-17 03.35.09.000000000 PM	24
7 movvar1279	Infra@1279	6	3	LoggedIn Successfully	01-DEC-17 01.36.59.000000000 PM	21
8 srikan	Infra@1279	7	3	LoggedIn Successfully	15-NOV-17 04.00.08.000000000 PM	8
9 rohit1279	Infra@1234	8	3	LoggedIn Successfully	15-NOV-17 04.42.56.000000000 PM	13
10 sagds	Rohit1279	9	3	LoggedIn Successfully	15-NOV-17 04.49.58.000000000 PM	16

Session 2 (Bottom):

```
SELECT * FROM USERS WHERE ID IN (SELECT USERID
                                    FROM LOGIN_PROFILE
                                    WHERE NO_OF_LOGINS > 20
                                    AND ROLE_ID=3);
```

ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	STATE	ZIPCODE	CONTACT	EMAIL_ID	CREATED_DT
1	3 Srinivas	Raju	Nanapuram	14202, Fairway Oaks	Apt No : 202	Tampa	Florida	5	33613 (null)	srinivasraju@gmail.com	14-NOV-17 04.11
2	4 Venkata	Ravi Teja	Lanka	Ashford Green	Flat No: G	Hyderabad	Telangana	1	500035 (null)	raviteja@gmail.com	15-NOV-17 02.54
3	5 Rohit	(null)	Movva	4209 Winding Moss Trail, Apt No:105	TAMPA	Florida	4	33613 (null)	rohit1279@gmail.com	15-NOV-17 03.35	
4	6 Rohit	(null)	Movva	4209 Winding Moss Trail, Apt No:105	TAMPA	Florida	2	33613 (null)	movvar@mail.usf.edu	15-NOV-17 03.39	

Now, in session 1, we have executed the same query to get the users who are having more than 20 number of logins. Here as the Read Only transaction is set, the data has not been reflected in this particular session.

Session 1 (Top): Transaction Block

```
COMMIT;--Committing all database changes before setting up transaction
SET TRANSACTION READ ONLY NAME 'USERS_LOGIN_REPORT';--Setting Transaction for Report
```

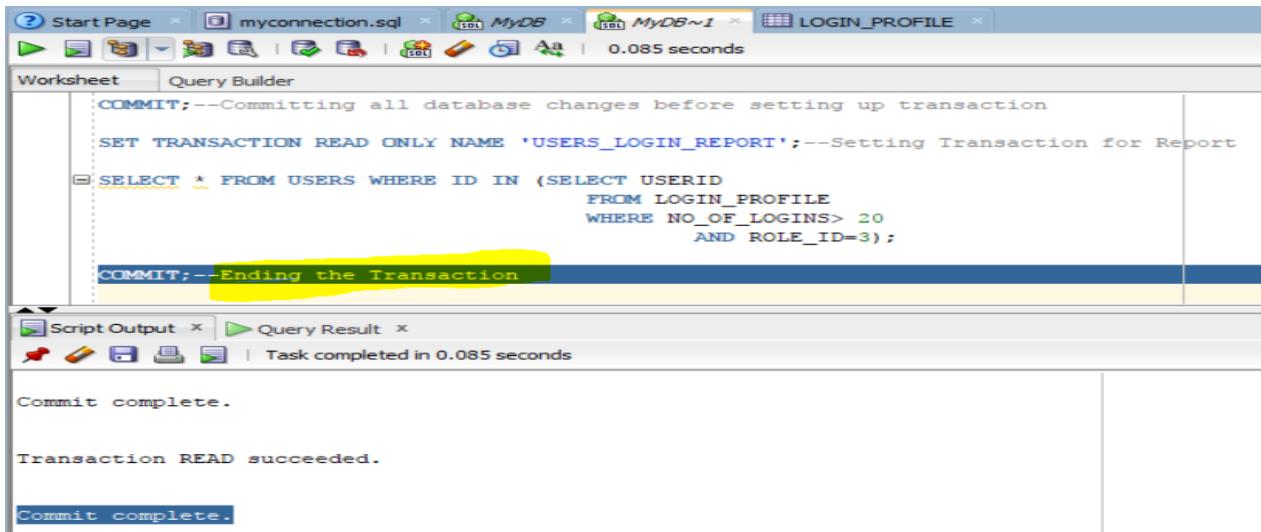
Session 2 (Bottom):

```
SELECT * FROM USERS WHERE ID IN (SELECT USERID
                                    FROM LOGIN_PROFILE
                                    WHERE NO_OF_LOGINS > 20
                                    AND ROLE_ID=3);
```

ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	STATE	ZIPCODE	CONTACT	EMAIL_ID	CREATED_DT
1	3 Srinivas	Raju	Nanapuram	14202, Fairway Oaks	Apt No : 202	Tampa	Florida	5	33613 (null)	srinivasraju@gmail.com	14-NOV-17 04.11
2	4 Venkata	Ravi Teja	Lanka	Ashford Green	Flat No: G	Hyderabad	Telangana	1	500035 (null)	raviteja@gmail.com	15-NOV-17 02.54
3	5 Rohit	(null)	Movva	4209 Winding Moss Trail, Apt No:105	TAMPA	Florida	4	33613 (null)	rohit1279@gmail.com	15-NOV-17 03.35	

Now, we are ending the transaction by executing “COMMIT”. The screenshot is shown as below.

Final Project Report for Sports Aggregator Information System



The screenshot shows a MySQL Workbench interface with a query editor and a script output window. The query editor contains a transaction script:

```
COMMIT;--Committing all database changes before setting up transaction
SET TRANSACTION READ ONLY NAME 'USERS_LOGIN_REPORT';--Setting Transaction for Report
SELECT * FROM USERS WHERE ID IN (SELECT USERID
                                    FROM LOGIN_PROFILE
                                    WHERE NO_OF_LOGINS> 20
                                    AND ROLE_ID=3);
COMMIT;--Ending the Transaction
```

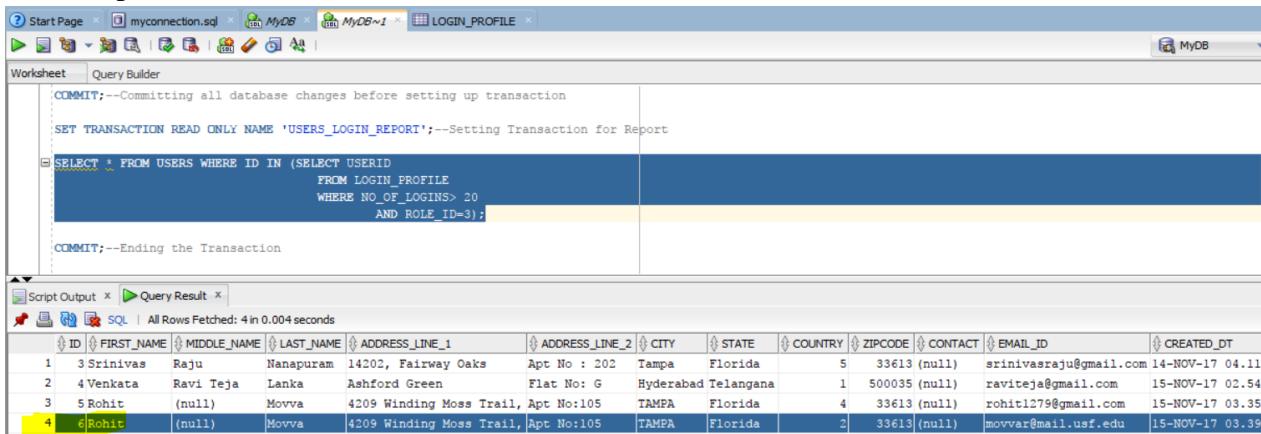
The script output window shows the results of the query:

```
Commit complete.

Transaction READ succeeded.

Commit complete.
```

Now, again when we have executed the same query in the transaction session, we are able to see the updated data i.e., 4th user also.



The screenshot shows a MySQL Workbench interface with a query editor and a script output window. The query editor contains a transaction script:

```
COMMIT;--Committing all database changes before setting up transaction
SET TRANSACTION READ ONLY NAME 'USERS_LOGIN_REPORT';--Setting Transaction for Report
SELECT * FROM USERS WHERE ID IN (SELECT USERID
                                    FROM LOGIN_PROFILE
                                    WHERE NO_OF_LOGINS> 20
                                    AND ROLE_ID=3);
COMMIT;--Ending the Transaction
```

The script output window shows the results of the query:

```
All Rows Fetched: 4 in 0.004 seconds
```

ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	STATE	ZIPCODE	CONTACT	EMAIL_ID	CREATED_DT
1	3 Srinivas	Raju	Nanapuram	14202, Fairway Oaks	Apt No : 202	Tampa	Florida	5 33613 (null)	srinivasraju@gmail.com	14-NOV-17 04.11	
2	4 Venkata	Ravi Teja	Lanka	Ashford Green	Flat No: G	Hyderabad	Telangana	1 500035 (null)	raviteja@gmail.com	15-NOV-17 02.54	
3	5 Rohit	(null)	Movva	4209 Winding Moss Trail, Apt No:105	TAMPA	Florida		4 33613 (null)	rohit1279@gmail.com	15-NOV-17 03.35	
4	6 Rohit	(null)	Movva	4209 Winding Moss Trail, Apt No:105	TAMPA	Florida		2 33613 (null)	movvar@mail.usf.edu	15-NOV-17 03.39	

Conclusion on Transaction Level Read Consistency:

The data seen by all queries within the same transaction is consistent with respect to a single point in time. Transaction-level read consistency produces repeatable reads and does not expose a query to phantoms. SET TRANSACTION READ ONLY is used for getting the transaction level read consistency. So, for business purpose the manager/agents can retrieve the “USER_LOGIN” reports by setting this type of read only transactions which in turn will be used to implement analytics for business development.

8. Application Development

On this database, we have designed a web application using **Microsoft Visual Studio 2013**, **SQL Developer** tools, ASP.NET Framework and languages **C#, HTML, JavaScript, SQL** and **PL/SQL** where the application interacts with the database for the information.

8.1. Working on Database

For connecting to database from Microsoft Visual Studio, first we need to install **Oracle Database Access Components (ODAC)**, where itself it asks for **HOSTNAME**, **PORT** and **SERVICE_NAME** details of the database to generate the **TNSNAMES.ORA** file with below contents and then add Extensions **Oracle.DataAccess** and **Oracle.ManagedDataAccess** in Microsoft Visual Studio.

```
MyDB =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = reade.forest.usf.edu)(PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = cdb9)  
    )  
  )
```

Then in **C#** code we need to add the below references that will be used for connecting to mentioned database

```
using Oracle.DataAccess.Client; // ODP.NET Oracle managed provider  
using Oracle.DataAccess.Types;
```

Then in code we need to add below lines for opening a connection to database.

```
OracleConnection conn = new OracleConnection("Data Source=MyDB;User  
Id=DB626;Password=*****"); //Database details  
conn.Open();
```

For Reading the data from database, use the C# code for **SELECT** queries as below example.

```
string sql = "SELECT COUNT(DISTINCT USERNAME) AS COUNT FROM LOGIN_PROFILE WHERE  
UPPER(USERNAME)=UPPER('" + username + "');  
OracleCommand cmd = new OracleCommand(sql, conn);  
OracleDataReader dr = cmd.ExecuteReader();  
dr.Read();  
int count = Convert.ToInt32(dr["COUNT"].ToString());  
dr.Dispose();
```

For Writing data into database, use the C# code for **INSERT** queries as below example.

```
string sql1="INSERT INTO USERS VALUES  
USERS_ID_SEQ.NEXTVAL,'" + fname + "','" + mname + "','" + lname + "','" + Address1 + "','" + Addr  
ess2 + "','" + city + "','" + state + "',(SELECT ID FROM COUNTRIES WHERE  
UPPER(COUNTRY)=UPPER('" + country + "')),'" + zipcode + "','" + email + "',SYSDATE,SYSDA  
TE)";  
OracleCommand cmd1 = new OracleCommand(sql1, conn);  
cmd1.ExecuteNonQuery();
```

Use the C# code for **UPDATE** queries as below example.

```
string sql2 = "UPDATE LOGIN_PROFILE SET  
NO_OF_LOGINS=NO_OF_LOGINS+1,LASTLOGINDATE=SYSDATE,LOGINSTATUS='LoggedIn'  
Successfully' WHERE UPPER(USERNAME)=UPPER('" + username + "');  
OracleCommand cmd2 = new OracleCommand(sql2, conn);  
cmd2.ExecuteNonQuery();
```

8.2. APPLICATION OVERVIEW

Now here we are demonstrating an end-to-end journey how a User can post their own article and later how it is being validated and published to all other users by Agent along with description of each page in this journey and how those pages interacting with database to read/write the information.

System's Home Page



This page contains the list of sports for which the information will be given by system to the users. When user clicks on the respective **sport's image** it displays basic information about the sport (as of now it is developed to redirect to the Wikipedia page of that sport). Once the user gets awareness about the sports he/she can register to the system by selecting their interested sports for which they require the information.

- When we click on “**Sports Aggregator Information System**” or “**Home**” the page will be in the same System’s Home page.
- When we click on the “**About**”, it will redirect to the page which contains some information about the website.

Final Project Report for Sports Aggregator Information System

- When we click on “**Contact**”, it will redirect to the page which contains the contact details of the supporting team of the website.
- When we click on “**Register**”, it redirects to the registration page where user can register to the system.
- When we click on “**Log in**”, it redirects to the page where user can login to the system.

Registration Page

The screenshot shows a registration form titled "Register" with the sub-instruction "Create a new account". The form includes fields for First Name, Middle Name, Last Name, Address Line1, Address Line2, City, State, Zip Code, Country (a dropdown menu), Interested Sports (checkboxes for Badminton, Baseball, Basketball, Cricket, Football, Golf, Hockey, Tennis, and Volleyball), Email, Username, Password, and Confirm password. At the bottom are "Register" and "Cancel" buttons. The browser address bar shows "localhost:27464/Account/Register". The top navigation bar includes links for Home, About, Contact, Register, and Log in.

First Name *

Middle Name

Last Name *

Address Line1 *

Address Line2

City *

State *

Zip Code *

Country *

Interested Sports *

Badminton Baseball Basketball
Cricket Football Golf
Hockey Tennis Volleyball

Email *

Username *

Password *

Confirm password *

Register Cancel

© 2017 - Group 3 Sports Web Application

This is the page where the users register to the system by entering all Mandatory fields, select their interested sports and clicks on Register button. Then the application validates fields and inserts these details into below database tables as mentioned in [section 6.1.1](#).

The data in the countries dropdown list is read from the database **COUNTRIES** table which has 197 countries list. Also, the interested Sports list is read from the **SPORTS** table which has those 9 sports.

Final Project Report for Sports Aggregator Information System

USERS Table:

SQL Worksheet | History

Worksheet | Query Builder

```
SELECT * FROM USERS order by 1;
```

Query Result | All Rows Fetched: 9 in 0.08 seconds

ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	STATE	ZIPCODE	CONTACT	EMAIL_ID
1	2 Rohith	(null)	Movva	4209 Winding Moss Trail, Apt No:105		TAMPA	Florida	5	33613 (null)	movvarrohit1@gmail.com
2	3 Srinivas	Raju	Nanapuram	14202, Fairway Oaks	Apt No : 202	Tampa	Florida	5	33613 (null)	srinivasraju@gmail.com
3	4 Venkata	Ravi Teja	Lanka	Ashford Green	Flat No: G	Hyderabad	Telangana	1	500035 (null)	raviteja@gmail.com
4	5 Rohit	(null)	Movva	4209 Winding Moss Trail, Apt No:105		TAMPA	Florida	4	33613 (null)	rohit1279@gmail.com
5	6 Rohit	(null)	Movva	4209 Winding Moss Trail, Apt No:105		TAMPA	Florida	2	33613 (null)	movvar@mail.usf.edu
6	7 Srikanth	(null)	Kotamaraju	14202, Fairway Oaks	Apt No : 202	Tampa	Florida	2	33613 (null)	srikanth@gmail.com
7	8 Rohit	(null)	Movva	4209 Winding Moss Trail, 14		TAMPA	Florida	1	33613 (null)	madskmd@gmail.com
8	9 Ravi	(null)	Teja	4209 Winding Moss Trail, Apt No:105		TAMPA	Florida	1	33613 (null)	saf@sads.com
9	10 Rohit	(null)	Movva	4209 Winding Moss Trail, Apt No:105		TAMPA	Florida	1	33613 (null)	movvarrohit1@gmail.com

LOGIN_PROFILE Table:

SQL Worksheet | History

Worksheet | Query Builder

```
SELECT * FROM LOGIN_PROFILE order by 3;
```

Query Result | All Rows Fetched: 10 in 0.032 seconds

USERNAME	PASSWORD	USERID	ROLE_ID	LOGINSTATUS	LASTLOGINDATE	NO_OF_LOGINS
1 agent123	agent123	2	2	LoggedIn Successfully	01-12-17 02:50:01.000000000 PM	8
2 movvar	Infra@2017	2	3	LoggedIn Successfully	14-11-17 04:05:37.000000000 PM	13
3 sriini	Pass@1234	3	3	LoggedIn Successfully	14-11-17 04:11:35.000000000 PM	25
4 lankav	Civil@123	4	3	LoggedIn Successfully	15-11-17 02:54:22.000000000 PM	33
5 MOVVAR123	Infra@2017	5	3	LoggedIn Successfully	15-11-17 03:35:09.000000000 PM	24
6 movvar1279	Infra@1279	6	3	LoggedIn Successfully	01-12-17 02:52:53.000000000 PM	24
7 srikan	Infra@1279	7	3	LoggedIn Successfully	15-11-17 04:00:08.000000000 PM	8
8 rohit1279	Infra@1234	8	3	LoggedIn Successfully	15-11-17 04:42:56.000000000 PM	13
9 sagds	Rohit1279	9	3	LoggedIn Successfully	15-11-17 04:49:58.000000000 PM	16
10 movar	Infra@2017	10	3	LoggedIn Successfully	18-11-17 06:07:40.000000000 PM	12

USERS_SPORTS Table:

SQL Worksheet | History

Worksheet | Query Builder

```
SELECT * FROM USERS_SPORTS order by 3;
```

Query Result | All Rows Fetched: 21 in 0.044 seconds

USER_ID	SPORT_ID	CREATED_DT
2	2	6 15-11-17 03:39:49.000000000 PM
3	3	6 15-11-17 03:39:49.000000000 PM
4	4	6 15-11-17 03:39:49.000000000 PM
5	5	6 15-11-17 03:39:49.000000000 PM
6	6	7 15-11-17 04:00:08.000000000 PM
7	7	7 15-11-17 04:00:08.000000000 PM
8	8	7 15-11-17 04:00:08.000000000 PM
9	9	7 15-11-17 04:00:08.000000000 PM
10	10	7 15-11-17 04:00:08.000000000 PM
11	11	8 15-11-17 04:42:56.000000000 PM
12	12	8 15-11-17 04:42:56.000000000 PM
13	13	8 15-11-17 04:42:56.000000000 PM
14	14	9 15-11-17 04:49:58.000000000 PM
15	15	9 15-11-17 04:49:58.000000000 PM
16	16	9 15-11-17 04:49:58.000000000 PM
17	17	10 18-11-17 06:07:40.000000000 PM
18	18	10 18-11-17 06:07:40.000000000 PM
19	19	10 18-11-17 06:07:40.000000000 PM
20	20	10 18-11-17 06:07:40.000000000 PM
21	21	10 18-11-17 06:07:40.000000000 PM

Final Project Report for Sports Aggregator Information System

Login Page

The screenshot shows a web browser window titled "Log in - My ASP.NET App". The address bar displays "localhost:27464/Account/Login". The page header includes the title "Sports Aggregator Information System" and navigation links for "Home", "About", and "Contact". On the right, there are "Register" and "Log in" buttons. The main content area is titled "Log in." and contains the instruction "Use a local account to log in." Below this, there are two input fields: "Username *" with value "movvar1279" and "Password *" with obscured value. A "Remember me?" checkbox is present, followed by a "Log in" button with a mouse cursor hovering over it. At the bottom left, there is a link "Register as a new user".

© 2017 - Group 3 Sports Web Application

This is the Login page where users enter their Username and Password to log into the system. The validations, reading/writing from/into the database tables from application will be as mentioned in [section 6.1.2](#).

Before Login the Number of logins for username movvar1279 is 24, after logging in it has changed to 25

LOGIN_PROFILE Table:

The screenshot shows a SQL Server Management Studio (SSMS) interface. The top pane is a "Query Builder" window with the query "SELECT * FROM LOGIN_PROFILE;". The bottom pane is a "Query Result" window titled "Query Result" showing the results of the query. The results are presented in a table with the following columns: USERNAME, PASSWORD, USERID, ROLE_ID, LOGINSTATUS, LASTLOGINDATE, and NO_OF_L... (truncated). The data shows 10 rows of login information for various users.

USERNAME	PASSWORD	USERID	ROLE_ID	LOGINSTATUS	LASTLOGINDATE	NO_OF_L...
1 movar	Infra@2017	10	3	LoggedIn Successf...	18-11-17 06:07:40.000000000 PM	12
2 agent123	agent123	2	2	LoggedIn Successf...	01-12-17 02:50:01.000000000 PM	8
3 movvar	Infra@2017	2	3	LoggedIn Successf...	14-11-17 04:05:37.000000000 PM	13
4 sriini	Pass@1234	3	3	LoggedIn Successf...	14-11-17 04:11:35.000000000 PM	25
5 lankav	Civil@123	4	3	LoggedIn Successf...	15-11-17 02:54:22.000000000 PM	33
6 MOVVAR123	Infra@2017	5	3	LoggedIn Successf...	15-11-17 03:35:09.000000000 PM	24
7 movvar1279	Infra@1279	6	3	LoggedIn Successf...	02-12-17 10:54:56.000000000 AM	25
8 srikan	Infra@1279	7	3	LoggedIn Successf...	15-11-17 04:00:08.000000000 PM	8
9 rohit1279	Infra@1234	8	3	LoggedIn Successf...	15-11-17 04:42:56.000000000 PM	13
10 sagds	Rohit1279	9	3	LoggedIn Successf...	15-11-17 04:49:58.000000000 PM	16

Final Project Report for Sports Aggregator Information System

User's Home Page

The screenshot shows a web browser window titled "HomePage" at "localhost:27464/Users/HomePage". The header includes a "Sports Aggregator Information System" logo, "Home", "About", and "Contact" links, and a "Hello, movvar1279 !" log off link. Below the header are four buttons: "Players", "Tournaments", "Tickets", and "Posts". A "Notifications :" section displays two messages: "23-11-2017 00:56:12 - Tickets are available in Bookmyshow,bcci website" and "22-11-2017 23:37:38 - Today there is an ODI cricket match between India and Pakistan".

Articles :

The article list includes two items:

- Title:** Rahul Dravid
Description: As of December 2016, Dravid is the fourth-highest run scorer in Test cricket, after Sachin Tendulkar, Ricky Ponting and Jacques Kallis.^{[8][9]} In 2004, after completing his century against Bangladesh in Chittagong, he became the first and the only player till date to score a century in all the ten Test-playing countries.^[10] As of October 2012, he holds the record for the most number of catches taken by a player (non-wicket-keeper) in Test cricket, with 210.^[11] Dravid holds a unique record of never getting out for a Golden duck in the 28 Test innings which he has played. He faced 31256 balls, which is highest number of balls face by any player in test cricket. He also spent 44152 minutes at the crease, which is highest time spent on crease by any player in test cricket.
Posted_by: Rohit Movva [Document 1](#) [Document 2](#)
- Title:** Nadal
Description: The highest run scorer of all time in International cricket, Tendulkar took up cricket at the age of eleven, made his Test debut on 15 November 1989 against Pakistan in Karachi at the age of sixteen, and went on to represent Mumbai domestically and India internationally for close to twenty-four years. He is the only player to have scored one hundred international centuries, the first batsman to score a double century in a One Day International, the holder of the record for the most number of runs in both ODI and Test cricket, and the only player to complete more than 30,000 runs in international cricket.
Posted_by: Rohit Movva [Document 1](#) [Document 2](#)

© 2017 - Group 3 Sports Web Application

This page contains the latest Notifications, valid articles posted by users and agents related to the user registered sports. In this Page,

- When User clicks on “**Sports Aggregator Information System**”, the application redirects to Systems Home Page.
- When User Clicks on “**Home**”, it redirects to User’s Home Page.
- When User clicks on “Hello, <USERNAME> !” then it redirects to Manage account details page where user can edit their details.
- When user clicks on “**logoff**” then it logs out the user from application and redirects to Systems Home Page.
- When User require Players details they can navigate to players page by clicking the “**Players**” button.
- When User require Tournaments details they can navigate to tournaments page by clicking the “**Tournaments**” button.
- When User wants to purchase tickets, they can navigate to tickets page by clicking the “**Tickets**” button where the list of ticket vendors with their respective URL details provided.

Final Project Report for Sports Aggregator Information System

- Users can post their article in Posts page, where they can navigate to that page by clicking on “**Posts**” button.

All pages of users are Read only pages other than Posts page where User can write their own articles.

NOTIFICATIONS Table:

The screenshot shows the MySQL Workbench interface with the 'Query Result' tab selected. A query has been run: `SELECT * FROM NOTIFICATIONS;`. The results are displayed in a table with columns: ID, NOTIFICATION, CREATED_DT, CREATED_BY, and SPORT_ID. There are two rows of data.

ID	NOTIFICATION	CREATED_DT	CREATED_BY	SPORT_ID
1	1Today there is an ODI cricket match between India and ...	22-11-17 01:07:38.000000000 PM	1	1
2	2Tickets are available in Bookmyshow,bcci website	22-11-17 02:26:12.000000000 PM	1	1

USER_ARTICLES Table:

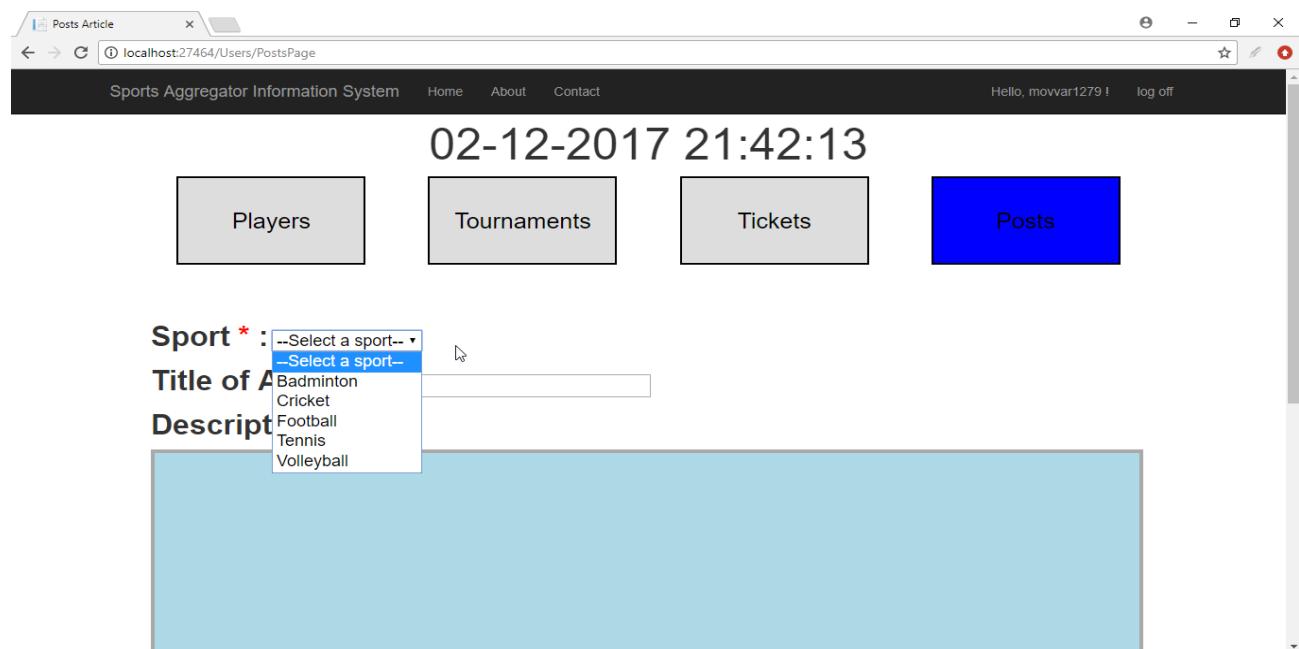
The screenshot shows the MySQL Workbench interface with the 'Query Result' tab selected. Two queries have been run. The first query is: `SELECT * FROM USER_ARTICLES WHERE VALID_FLAG='Y';`. The results are displayed in a table with columns: ID, SPORT_ID, TITLE, DESCRIPTION1. There are two rows of data. The second query is also: `SELECT * FROM USER_ARTICLES WHERE VALID_FLAG='Y';`. The results are displayed in a table with columns: DESCRIPTION2, DOCUMENT1, DOCUMENT2, IMAGE1, POSTED_BY, POSTED_DT, VALID_FLAG, VALIDATED_BY, VALIDATED_DT. There are two rows of data.

ID	SPORT_ID	TITLE	DESCRIPTION1
1	2	1Rahul Dravid	As of December 2016, Dravid is the fourth-highest run scorer in Test cricket, after Sachin Tendulkar, Ricky Ponting and Brian Lara.
2	4	4Nadal	The highest run scorer of all time in International cricket, Tendulkar took up cricket at the age of eleven, made his debut for the Indian national team in 1989 and became captain in 1994.

	DESCRIPTION2	DOCUMENT1	DOCUMENT2	IMAGE1	POSTED_BY	POSTED_DT	VALID_FLAG	VALIDATED_BY	VALIDATED_DT
1	cket.	(null)	(BLOB)	(BLOB)		26-11-17 04:35:02.000000000 PM	Y		29-11-17 11:01:45.000000000 AM
2		(null)	(BLOB)	(BLOB)		27-11-17 05:50:51.000000000 PM	Y		01-12-17 02:50:54.000000000 PM

Final Project Report for Sports Aggregator Information System

User's Posts Page



This is the page where user can post their own articles for their registered sports. The list of sports in the dropdown are read from database using below SELECT query.

Query for list of sports:

```
SELECT NAME FROM SPORTS WHERE ID IN (SELECT SPORT_ID FROM USERS_SPORTS WHERE USER_ID IN (SELECT USERID FROM LOGIN_PROFILE WHERE UPPER(USERNAME) = UPPER('" + Session["Username"].ToString() + "'))) ORDER BY 1
```

The screenshot shows the SQL Server Management Studio interface. The "Query Builder" tab is active, displaying the previously provided SELECT query. The "Query Result" tab shows the execution results, which is a table with one column "NAME" containing five rows: Badminton, Cricket, Football, Tennis, and Volleyball.

NAME
1 Badminton
2 Cricket
3 Football
4 Tennis
5 Volleyball

Final Project Report for Sports Aggregator Information System

Now I am posting an article related to “Cricket” sport by uploading an image file and 2 document files.

The screenshot shows a web browser window titled "Posts Article" at the URL "localhost:27464/Users/PostsPage". The page belongs to the "Sports Aggregator Information System". The navigation bar includes links for Home, About, Contact, and a log-off option. Below the navigation bar are four buttons: Players, Tournaments, Tickets, and Posts, where "Posts" is highlighted in blue. The main content area contains fields for "Sport * : Cricket", "Title of Article * : MS Dhoni", and "Description * :". The "Description" field contains a detailed text about Mahendra Singh Dhoni. Below this is a section for "Upload an Image:" with a file input field showing "Dhoni.jpg" has been added successfully. There are "Add" and "Remove" buttons. Below that is a section for "Upload Documents:" with a file input field showing "M3.docx" and "M3.pdf" selected. There are "Add" and "Remove" buttons, and a "Post" button at the bottom right.

Once user clicks on the post button at the bottom of the page, the application validates all mandatory fields and then insert new record into USER_ARTICLES table along with the files uploaded as BLOB datatype.

Backend C# Code:

```
protected void Post_Click(object sender, EventArgs e)
{
    string sport = Sport.Text;
    string title = Article_Title.Text;
    string description = Description.Text;
    string description1 = string.Empty;
    string description2 = string.Empty;
    OracleConnection conn = new OracleConnection("Data Source=MyDB;User
Id=DB626;Password=infra@2017;");
    conn.Open();
    if (description.Length > 4000)
```

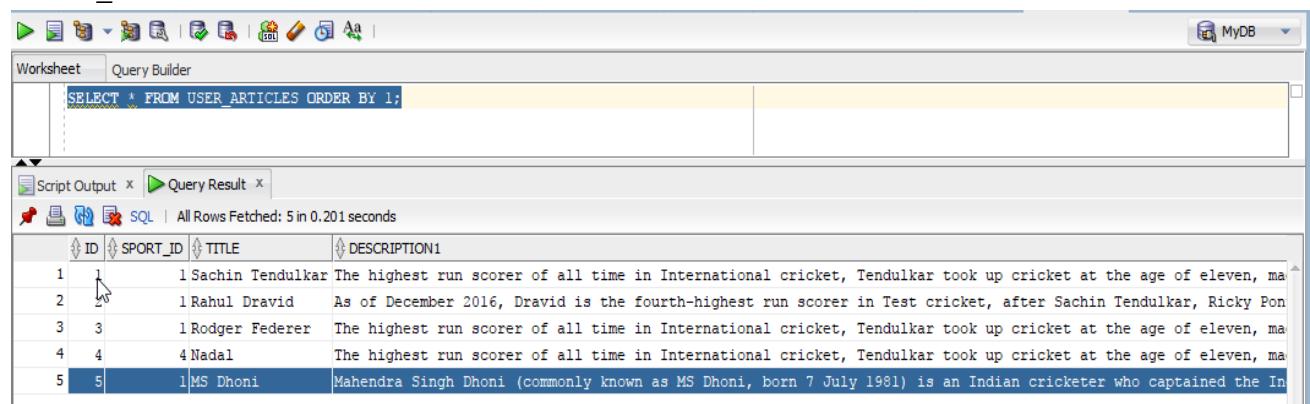
Final Project Report for Sports Aggregator Information System

```
{  
    description1 = description.Substring(1, 4000);  
    description2 = description.Substring(4001, description.Length);  
}  
else  
{  
    description1 = description;  
    description2 = string.Empty;  
}  
string sql = "INSERT INTO USER_ARTICLES VALUES  
(USER_ARTICLES_ID_SEQ.NEXTVAL,(SELECT ID FROM SPORTS WHERE UPPER(NAME)=UPPER('" +  
sport + "')),'" + title + "','" + description1 + "','" + description2 +  
"',:iDocument1,:iDocument2,:iImage,(SELECT USERID FROM LOGIN_PROFILE WHERE  
UPPER(USERNAME) = UPPER('" + Session["Username"].ToString() +  
"')),SYSDATE,'N',null,null)";  
OracleCommand cmd = new OracleCommand(sql, conn);  
cmd.Parameters.Add(":iDocument1", OracleDbType.Blob,  
required.bytes2.Length).Value = required.bytes2;  
cmd.Parameters.Add(":iDocument2", OracleDbType.Blob,  
required.bytes2.Length).Value = required.bytes3;  
cmd.Parameters.Add(":iImage", OracleDbType.Blob,  
required.bytes2.Length).Value = required.bytes1;  
cmd.ExecuteNonQuery();  
conn.Dispose();  
  
Response.Write("<script LANGUAGE='JavaScript' >alert('Article Posted  
successfully')</script>");  
Response.Redirect("HomePage.aspx");  
}
```

Where required.bytes1, required.bytes2, required.bytes3 are the binary data of the image, document1 and document2 files that are uploaded by user respectively.

```
class required{  
    public static Stream fs1, fs2, fs3;  
    public static BinaryReader br1, br2, br3;  
    public static Byte[] bytes1 = null;  
    public static Byte[] bytes2 = null;  
    public static Byte[] bytes3 = null;  
}  
required.fs1 = FileUpload1.PostedFile.InputStream;  
required.br1 = new BinaryReader(required.fs1);  
required.bytes1 = required.br1.ReadBytes((Int32)required.fs1.Length);
```

USER_ARTICLES Table:



The screenshot shows the Oracle SQL Developer interface. The top window displays a SQL query: "SELECT * FROM USER_ARTICLES ORDER BY 1;". The bottom window, titled "Query Result", shows the results of the query. The table has four columns: ID, SPORT_ID, TITLE, and DESCRIPTION1. The data is as follows:

ID	SPORT_ID	TITLE	DESCRIPTION1
1	1	Sachin Tendulkar	The highest run scorer of all time in International cricket, Tendulkar took up cricket at the age of eleven, ma
2	1	Rahul Dravid	As of December 2016, Dravid is the fourth-highest run scorer in Test cricket, after Sachin Tendulkar, Ricky Pon
3	1	Roger Federer	The highest run scorer of all time in International cricket, Tendulkar took up cricket at the age of eleven, ma
4	4	Nadal	The highest run scorer of all time in International cricket, Tendulkar took up cricket at the age of eleven, ma
5	5	MS Dhoni	Mahendra Singh Dhoni (commonly known as MS Dhoni, born 7 July 1981) is an Indian cricketer who captained the In

Final Project Report for Sports Aggregator Information System

Screenshot of a MySQL Workbench interface showing the results of a SQL query:

```
SELECT * FROM USER_ARTICLES ORDER BY 1;
```

The results table has the following columns and data:

	DESCRIPTION2	DOCUMENT1	DOCUMENT2	IMAGE1	POSTED_BY	POSTED_DT	VALID_FLAG	VALIDATED_BY	VALIDATED_DT
1	(null)	(BLOB)	(BLOB)	(BLOB)		6 26-11-17 12:37:35.000000000 AM N		(null)	(null)
2	(null)	(BLOB)	(BLOB)	(BLOB)		6 26-11-17 04:35:02.000000000 PM Y		2 29-11-17 11:01:45.000000000 AM	
3	(null)	(BLOB)	(BLOB)	(BLOB)		6 27-11-17 05:34:28.000000000 PM N		(null)	(null)
4	(null)	(BLOB)	(BLOB)	(BLOB)		6 27-11-17 05:50:51.000000000 PM Y		2 01-12-17 02:50:54.000000000 PM	
5 20.	(null)	(BLOB)	(BLOB)	(BLOB)		6 02-12-17 11:42:17.000000000 AM N		(null)	(null)

Screenshot of a MySQL Workbench interface showing the results of a SQL query and a modal dialog for viewing an image:

```
SELECT * FROM USER_ARTICLES ORDER BY 1;
```

The results table has the same structure as above, with the last row containing a BLOB value for IMAGE1.

A modal dialog titled "View Value" is open, showing a thumbnail image of a man in a suit holding a trophy. The dialog has tabs for "Information" and "Saved Image".

So after posting the article, immediately the article will be sent for validation to the agents who are related to that article's sport. Once the agent validates it then that article will be published to all the Users, Agents and Managers.

Agent's Home Page

Screenshot of a web browser showing the login page of the "Sports Aggregator Information System":

The URL is <http://localhost:27464/account/login>.

The page title is "Log in - My ASP.NET App".

The content area displays:

Log in.

Use a local account to log in.

Username:

Password:

Remember me?

[Register as a new user](#)

© 2017 - Group 3 Sports Web Application

Agent Logging in to the system with his credentials.

Final Project Report for Sports Aggregator Information System

The screenshot shows a web browser window for 'Sports Aggregator Information System' at 'localhost:27464/Agents/HomePage'. The header includes a logo, navigation links for 'Home', 'About', and 'Contact', and a user session 'Hello, agent123 ! log off'. Below the header is a date and time display '02-12-2017 22:39:06'. There are four main buttons: 'Players', 'Tournaments', 'Tickets', and 'Posts (3)'. A 'Notifications' section displays two messages: '23-11-2017 00:56:12 - Tickets are available in Bookmyshow,bcci website' and '22-11-2017 23:37:38 - Today there is an ODI cricket match between India and Pakistan'. The 'Articles' section lists two entries. The first entry is about 'Rahul Dravid' with a photo, a title, a detailed description, and author information 'Posted_by: Rohit Movva' with links to 'Document 1' and 'Document 2'. The second entry is about 'Nadal' with a photo, a title, a description, and author information 'Posted_by: Rohit Movva' with links to 'Document 1' and 'Document 2'. At the bottom left is a copyright notice: '© 2017 - Group 3 Sports Web Application'.

This page contains the latest Notifications, valid articles posted by users and agents related to the user registered sports. In this Page,

- When Agent clicks on “**Sports Aggregator Information System**”, the application redirects to Systems Home Page.
- When Agent Clicks on “**Home**”, it redirects to Agent’s Home Page.
- When Agent clicks on “**Hello, <USERNAME> !**” then it redirects to Manage account details page where agents can edit their details.
- When Agent clicks on “**logoff**” then it logs out the user from application and redirects to Systems Home Page.
- When Agent require to View/Add/Edit Players details they can navigate to players page by clicking the “**Players**” button.
- When Agent require to View/Add/Edit Tournaments details they can navigate to tournaments page by clicking the “**Tournaments**” button.
- When Agent wants to purchase tickets or View/Add/Edit the ticket vendor’s details, they can navigate to tickets page by clicking the “**Tickets**” button where the list of ticket vendors with their respective URL details provided.

Final Project Report for Sports Aggregator Information System

- Agent can Validate the User's articles and post their articles in Posts page, where they can navigate to that page by clicking on “Posts” button. The “Posts” button will contain the count of articles that needs to be validated by the Agent

All pages of Agents are both Read and Write pages.

Agent's Posts Page

The screenshot shows a web application window titled "Posts Article". The URL is "localhost:27464/Agents/PostsPage". The top navigation bar includes "Sports Aggregator Information System", "Home", "About", "Contact", "Hello, agent123 !", and "log off". Below the navigation is a large date and time display: "02-12-2017 22:41:00". A grid of buttons is centered: "Players", "Tournaments", "Tickets", "Posts (3)" (highlighted in blue), "Validate Posts (3)" (highlighted in blue), and "Add Posts". Below the grid, three article cards are listed:

- Sachin Tendulkar**
Title: Sachin Tendulkar
Description: The highest run scorer of all time in International cricket, Tendulkar took up cricket at the age of eleven, made his Test debut on 15 November 1989 against Pakistan in Karachi at the age of sixteen, and went on to represent Mumbai domestically and India internationally for close to twenty-four years. He is the only player to have scored one hundred international centuries, the first batsman to score a double century in a One Day International, the holder of the record for the most number of runs in both ODI and Test cricket, and the only player to complete more than 30,000 runs in international cricket.
Posted_by: Rohit Movva
Document 1 Document 2 Publish Delete
- Roger Federer**
Title: Rodger Federer
Description: The highest run scorer of all time in International cricket, Tendulkar took up cricket at the age of eleven, made his Test debut on 15 November 1989 against Pakistan in Karachi at the age of sixteen, and went on to represent Mumbai domestically and India internationally for close to twenty-four years. He is the only player to have scored one hundred international centuries, the first batsman to score a double century in a One Day International, the holder of the record for the most number of runs in both ODI and Test cricket, and the only player to complete more than 30,000 runs in international cricket.
Posted_by: Rohit Movva
Document 1 Document 2 Publish Delete
- MS Dhoni**
Title: MS Dhoni
Description: Mahendra Singh Dhoni (commonly known as MS Dhoni, born 7 July 1981) is an Indian cricketer who captained the Indian team in limited-overs formats from 2007 to 2016 and in Test cricket from 2008 to 2014. An attacking right-handed middle-order batsman and wicket-keeper, he is widely regarded as one of the greatest finishers in limited-overs cricket. He is also regarded to be one of the best wicket-keepers in world cricket and is known to have very fast hands. He made his One Day International (ODI) debut in December 2004 against Bangladesh, and played his first Test a year later against Sri Lanka. Dhoni holds numerous captaincy records such as most wins by an Indian captain in Tests and ODIs, and most back-to-back wins by an Indian captain in ODIs. He took over the ODI captaincy from Rahul Dravid in 2007 and led the team to its first-ever bilateral ODI series wins in Sri Lanka and New Zealand. Under his captaincy, India won the 2007 ICC World Twenty20, the CB Series of 2007–08, the 2010 Asia Cup, the 2011 ICC Cricket World Cup and the 2013 ICC Champions Trophy. In the final of the 2011 World Cup, Dhoni scored 91 not out off 79 balls handing India the victory for which he was awarded the Man of the Match. In June 2013, when India defeated England in the final of the Champions Trophy in England, Dhoni became the first captain to win all three ICC limited-overs trophies (World Cup, Champions Trophy and the World Twenty20).
Posted_by: Rohit Movva
Document 1 Document 2 Publish Delete

At the bottom left is a copyright notice: "© 2017 - Group 3 Sports Web Application".

When Agent clicks on Posts button it automatically redirects to the Validate Posts page, where it contains the list of articles that needs to be validated. So, the agent goes through the articles one after the other and clicks on respective Publish button if they are valid else clicks on Delete button. And finally clicks on Save button. Then in database the **VALID_FLAG** attribute in **USER_ARTICLES** table for those published articles will be changed to **Y** from **N**. Also, updates the **VALIDATED_BY** and **VALIDATED_DT** with the respective Agent's ID and the

Final Project Report for Sports Aggregator Information System

timestamp when it validated respectively. Now here agent published the article with the title “MS Dhoni” that has been posted previously by the user.

Script Output x Query Result x

All Rows Fetched: 5 in 0.284 seconds

ID	SPORT_ID	TITLE	DESCRIPTION1
1	1	Sachin Tendulkar	The highest run scorer of all time in International cricket, Tendulkar took up cricket at the age of eleven, ma
2	2	Rahul Dravid	As of December 2016, Dravid is the fourth-highest run scorer in Test cricket, after Sachin Tendulkar, Ricky Don
3	3	Roger Federer	The highest run scorer of all time in International cricket, Tendulkar took up cricket at the age of eleven, ma
4	4	Nadal	The highest run scorer of all time in International cricket, Tendulkar took up cricket at the age of eleven, ma
5	5	MS Dhoni	Mahendra Singh Dhoni (commonly known as MS Dhoni, born 7 July 1981) is an Indian cricketer who captained the In

Script Output x Query Result x

All Rows Fetched: 5 in 0.284 seconds

	DESCRIPTION2	DOCUMENT1	DOCUMENT2	IMAGE1	POSTED_BY	POSTED_DT	VALID_FLAG	VALIDATED_BY	VALIDATED_DT
1	(null)	(BLOB)	(BLOB)	(BLOB)	6	26-11-17 12:37:35.000000000 AM N		(null)	(null)
2	(null)	(BLOB)	(BLOB)	(BLOB)	6	26-11-17 04:35:02.000000000 PM Y		2	29-11-17 11:01:45.000000000 AM
3	(null)	(BLOB)	(BLOB)	(BLOB)	6	27-11-17 05:34:28.000000000 PM N		(null)	(null)
4	(null)	(BLOB)	(BLOB)	(BLOB)	6	27-11-17 05:50:51.000000000 PM Y		2	01-12-17 02:50:54.000000000 PM
5	(null)	(BLOB)	(BLOB)	(BLOB)	6	02-12-17 11:42:17.000000000 AM Y		2	02-12-17 12:07:21.000000000 PM

Article Published to all users

Once the article’s Valid_Flag is marked as ‘Y’ automatically it displays that article to all the users who had registered for that sport in their homepage along with the name of user who posted that article. Here now the article with title “MS Dhoni” will be published in everyone’s homepage who registered for “Cricket” Sport.

Final Project Report for Sports Aggregator Information System

Agent's Homepage

Home Page of Agent “agent123”. Now we can see the Posts button of Agent has count of 2 instead of 3 since one article (MS Dhoni) has been validated by the Agent.

HomePage ×

localhost:27464/Agents/HomePage

Sports Aggregator Information System Home About Contact Hello, agent123 ! log off

02-12-2017 22:43:45

Players Tournaments Tickets Posts (2)

Notifications :

23-11-2017 00:56:12 - Tickets are available in Bookmyshow,bcci website
22-11-2017 23:37:38 - Today there is an ODI cricket match between India and Pakistan

Articles :

Title: Rahul Dravid

Description: As of December 2016, Dravid is the fourth-highest run scorer in Test cricket, after Sachin Tendulkar, Ricky Ponting and Jacques Kallis.^{[8][9]} In 2004, after completing his century against Bangladesh in Chittagong, he became the first and the only player till date to score a century in all the ten Test-playing countries.^[10] As of October 2012, he holds the record for the most number of catches taken by a player (non-wicket-keeper) in Test cricket, with 210.^[11] Dravid holds a unique record of never getting out for a Golden duck in the 286 Test innings which he has played. He faced 31258 balls, which is highest number of balls face by any player in test cricket. He also spent 44152 minutes at the crease, which is highest time spent on crease by any player in test cricket.

Posted_by: Rohit Movva [Document 1](#) [Document 2](#)

Title: MS Dhoni

Description: Mahendra Singh Dhoni (commonly known as MS Dhoni, born 7 July 1981) is an Indian cricketer who captained the Indian team in limited-overs formats from 2007 to 2016 and in Test cricket from 2008 to 2014. An attacking right-handed middle-order batsman and wicket-keeper, he is widely regarded as one of the greatest finishers in limited-overs cricket. He is also regarded to be one of the best wicket-keepers in world cricket and is known to have very fast hands. He made his One Day International (ODI) debut in December 2004 against Bangladesh, and played his first Test a year later against Sri Lanka. Dhoni holds numerous captaincy records such as most wins by an Indian captain in Tests and ODIs, and most back-to-back wins by an Indian captain in ODIs. He took over the ODI captaincy from Rahul Dravid in 2007 and led the team to its first-ever bilateral ODI series wins in Sri Lanka and New Zealand. Under his captaincy, India won the 2007 ICC World Twenty20, the CBR Series of 2007–08, the 2010 Asia Cup, the 2011 ICC Cricket World Cup and the 2013 ICC Champions Trophy. In the final of the 2011 World Cup, Dhoni scored 91 not out off 79 balls handing India the victory for which he was awarded the Man of the Match. In June 2013, when India defeated England in the final of the Champions Trophy in England, Dhoni became the first captain to win all three ICC limited-overs trophies (World Cup, Champions Trophy and the World Twenty20).

Posted_by: Rohit Movva [Document 1](#) [Document 2](#)

Title: Nadal

Description: The highest run scorer of all time in International cricket, Tendulkar took up cricket at the age of eleven, made his Test debut on 15 November 1989 against Pakistan in Karachi at the age of sixteen, and went on to represent Mumbai domestically and India internationally for close to twenty-four years. He is the only player to have scored one hundred international centuries, the first batsman to score a double century in a One Day International, the holder of the record for the most number of runs in both ODI and Test cricket, and the only player to complete more than 30,000 runs in international cricket.

Posted_by: Rohit Movva [Document 1](#) [Document 2](#)

© 2017 - Group 3 Sports Web Application

The article has been published with the Image, Title, Description, documents uploaded by the User for that article and the concatenation of User's First_name and Last_Name at Posted_by field.

Final Project Report for Sports Aggregator Information System

User's Home Page

Home Page of User “movvar1279”.

The screenshot shows a web browser window titled "HomePage" with the URL "localhost:27464/Users/HomePage". The page header includes the title "Sports Aggregator Information System" and navigation links for "Home", "About", and "Contact". On the right, it displays a greeting "Hello, movvar1279!" and a "log off" link. The main content area features a large date and time display "02-12-2017 22:48:03". Below this are four rectangular buttons labeled "Players", "Tournaments", "Tickets", and "Posts". A section titled "Notifications :" contains two messages: "23-11-2017 00:56:12 - Tickets are available in Bookmyshow,bcci website" and "22-11-2017 23:37:38 - Today there is an ODI cricket match between India and Pakistan". A section titled "Articles :" lists three news items. Each item includes a thumbnail image, a title, a description, and a "Posted_by" field. The first article is about "Rahul Dravid", the second about "MS Dhoni", and the third about "Nadal". Each article has two "Document" links at the bottom right.

02-12-2017 22:48:03

Players Tournaments Tickets Posts

Notifications :

23-11-2017 00:56:12 - Tickets are available in Bookmyshow,bcci website
22-11-2017 23:37:38 - Today there is an ODI cricket match between India and Pakistan

Articles :

Title: Rahul Dravid
Description: As of December 2016, Dravid is the fourth-highest run scorer in Test cricket, after Sachin Tendulkar, Ricky Ponting and Jacques Kallis.[8][9] In 2004, after completing his century against Bangladesh in Chittagong, he became the first and the only player till date to score a century in all the ten Test-playing countries.[10] As of October 2012, he holds the record for the most number of catches taken by a player (non-wicket-keeper) in Test cricket, with 210.[11] Dravid holds a unique record of never getting out for a Golden duck in the 286 Test innings which he has played. He faced 31258 balls, which is highest number of balls face by any player in test cricket. He also spent 44152 minutes at the crease, which is highest time spent on crease by any player in test cricket.

Posted_by: Rohit Movva [Document 1](#) [Document 2](#)

Title: MS Dhoni
Description: Mahendra Singh Dhoni (commonly known as MS Dhoni, born 7 July 1981) is an Indian cricketer who captained the Indian team in limited-overs formats from 2007 to 2016 and in Test cricket from 2008 to 2014. An attacking right-handed middle-order batsman and wicket-keeper, he is widely regarded as one of the greatest finishers in limited-overs cricket. He is also regarded to be one of the best wicket-keepers in world cricket and is known to have very fast hands. He made his One Day International (ODI) debut in December 2004 against Bangladesh, and played his first Test a year later against Sri Lanka. Dhoni holds numerous captaincy records such as most wins by an Indian captain in Tests and ODIs, and most back-to-back wins by an Indian captain in ODIs. He took over the ODI captaincy from Rahul Dravid in 2007 and led the team to its first-ever bilateral ODI series wins in Sri Lanka and New Zealand. Under his captaincy, India won the 2007 ICC World Twenty20, the CWC Series of 2007–08, the 2010 Asia Cup, the 2011 ICC Cricket World Cup and the 2013 ICC Champions Trophy. In the final of the 2011 World Cup, Dhoni scored 91 not out off 79 balls handing India the victory for which he was awarded the Man of the Match. In June 2013, when India defeated England in the final of the Champions Trophy in England, Dhoni became the first captain to win all three ICC limited-overs trophies (World Cup, Champions Trophy and the World Twenty20).

Posted_by: Rohit Movva [Document 1](#) [Document 2](#)

Title: Nadal
Description: The highest run scorer of all time in International cricket, Tendulkar took up cricket at the age of eleven, made his Test debut on 15 November 1989 against Pakistan in Karachi at the age of sixteen, and went on to represent Mumbai domestically and India internationally for close to twenty-four years. He is the only player to have scored one hundred international centuries, the first batsman to score a double century in a One Day International, the holder of the record for the most number of runs in both ODI and Test cricket, and the only player to complete more than 30,000 runs in international cricket.

Posted_by: Rohit Movva [Document 1](#) [Document 2](#)

© 2017 - Group 3 Sports Web Application

9. Interesting Queries

9.1. For Size of all tables

Query:

```
SELECT DTA.TABLE_NAME,  
       DTA.TABLESPACE_NAME,  
       DTA.NUM_ROWS,  
       DTA.AVG_ROW_LEN,
```

Final Project Report for Sports Aggregator Information System

```
DTA.BLOCKS,  
DTA.EMPTY_BLOCKS,  
ROUND(DTA.BLOCKS * DTS.BLOCK_SIZE/1024,2) AS SIZE_KB  
FROM DBA_TABLES DTA  
JOIN DBA_TABLESPACES DTS ON  
DTA.TABLESPACE_NAME=DTS.TABLESPACE_NAME  
WHERE DTA.OWNER='DB626'  
ORDER BY 7 DESC;
```

The screenshot shows the Oracle SQL Developer interface. In the top window (Worksheet), a SQL query is displayed:

```
SELECT DTA.TABLE_NAME,  
       DTA.TABLESPACE_NAME,  
       DTA.NUM_ROWS,  
       DTA.AVG_ROW_LEN,  
       DTA.BLOCKS,  
       DTA.EMPTY_BLOCKS,  
       ROUND(DTA.BLOCKS * DTS.BLOCK_SIZE/1024,2) AS SIZE_KB  
  FROM DBA_TABLES DTA  
 JOIN DBA_TABLESPACES DTS ON DTA.TABLESPACE_NAME=DTS.TABLESPACE_NAME  
 WHERE DTA.OWNER='DB626'  
 ORDER BY 7 DESC;
```

In the bottom window (Query Result), the results of the query are shown in a grid:

	TABLE_NAME	TABLESPACE_NAME	NUM_ROWS	AVG_ROW_LEN	BLOCKS	EMPTY_BLOCKS	SIZE_KB
1	TOURNAMENTS	STUDENTS	900	95	16	0	128
2	USERS_SPORTS	STUDENTS	21	17	5	0	40
3	USERS	STUDENTS	9	117	5	0	40
4	AGENTS	STUDENTS	100	93	5	0	40
5	USER_ARTICLES	STUDENTS	5	1116	5	0	40
6	BASEBALL_PLAYERS	STUDENTS	1	69	5	0	40
7	COUNTRIES	STUDENTS	197	13	5	0	40
8	LOGIN_PROFILE	STUDENTS	10	61	5	0	40
9	MANAGERS	STUDENTS	20	70	5	0	40
10	NOTIFICATIONS	STUDENTS	2	76	5	0	40
11	ROLES	STUDENTS	3	43	5	0	40
12	SPORTS	STUDENTS	9	23	5	0	40

9.2. For current open Sessions

Query:

```
SELECT * FROM V$SESSION WHERE USERNAME='DB626';
```

Final Project Report for Sports Aggregator Information System

SADDR	SID	SERIAL#	AUDSID	PADDR	USER#	USERNAME	COMMAND	OWNERID	TADDR	LOCKWAIT	STATUS	SERVER	SCHEMA#
1 000007FF65EC3968	32	16251	15781367	000007FF65A589	3045	DB626		3	65571	(null)	(null)	ACTIVE	DEDICATED
2 000007FF65E8D6F8	35	27171	15781367	000007FF65A5EFB8	3045	DB626		3	2147483644	(null)	(null)	INACTIVE	DEDICATED
3 000007FF65E8E448	58	9456	15781362	000007FF65A567F8	3045	DB626		3	2147483644	(null)	(null)	ACTIVE	DEDICATED
4 000007FF65E8A2A8	60	11562	15781367	000007FF65A59538	3045	DB626		3	65571	(null)	(null)	ACTIVE	DEDICATED
													3045 D

Machines where the DB session is opened.

ANAME	OSUSER	PROCESS	MACHINE	PORT	TERMINAL	PROGRAM	TYPE	SQL_ADDRESS	SQL_HASH_VALUE	SQL_ID	SQL_CHILD_NUMBER
1 srini	2784	DESKTOP-993J6IE	56669	unknown	ORACLE.EXE (P000)	USER	000007FF5978D708	2099223842 93xdvxpjz792			C
2 srini	4448	DESKTOP-993J6IE	56669	unknown	SQL Developer	USER	000007FF5978D708	2099223842 93xdvxpjz792			C
3 Rohit	14564	Rohith	2518	unknown	SQL Developer	USER	000007FF2D6EC418	976012320 5pgrg9cx2tj10			C
4 srini	2788	DESKTOP-993J6IE	56669	unknown	ORACLE.EXE (P001)	USER	000007FF5978D708	2099223842 93xdvxpjz792			C

9.3. For Memory allocated to current sessions

Query:

```

SELECT a.inst_id,
       NVL(a.username,'(oracle)') AS username,
       a.module,
       a.program,
       Trunc(b.value/1024) AS memory_kb
FROM   gv$session a,
       gv$sesstat b,
       gv$statname c
WHERE  a.sid = b.sid
AND    a.inst_id = b.inst_id
AND    b.statistic# = c.statistic#
AND    b.inst_id = c.inst_id
AND    c.name = 'session pga memory'
AND    a.program IS NOT NULL
ORDER BY b.value DESC;

```

Final Project Report for Sports Aggregator Information System

The screenshot shows the Oracle SQL Developer interface. The top window is a Worksheet containing a SQL query:

```
SELECT a.inst_id,
       NVL(a.username,'(oracle)') AS username,
       a.module,
       a.program,
       Trunc(b.value/1024) AS memory_kb
  FROM gv$session a,
       gv$sesstat b,
       gv$statname c
 WHERE a.sid = b.sid
   AND a.inst_id = b.inst_id
   AND b.statistic# = c.statistic#
   AND b.inst_id = c.inst_id
   AND c.name = 'session pga memory'
   AND a.program IS NOT NULL
 ORDER BY b.value DESC;
```

The bottom window is a Query Result grid showing the results of the query. The columns are INST_ID, USERNAME, MODULE, PROGRAM, and MEMORY_KB. The data includes sessions for Oracle processes like MMON, DBW0, CJQ0, and various SQL Developers.

INST_ID	USERNAME	MODULE	PROGRAM	MEMORY_KB
1	1 (oracle)	(null)	ORACLE.EXE (MMON)	11593
2	1 DB613	SQL Developer	SQL Developer	9776
3	1 (oracle)	(null)	ORACLE.EXE (DBW0)	8056
4	1 (oracle)	(null)	ORACLE.EXE (CJQ0)	3145
5	1 DB626	SQL Developer	SQL Developer	2928
6	1 DB626	SQL Developer	SQL Developer	2672
7	1 (oracle)	(null)	ORACLE.EXE (DIA0)	2160
8	1 (oracle)	(null)	ORACLE.EXE (DBRM)	1857
9	1 (oracle)	KTSJ	ORACLE.EXE (W002)	1857
10	1 (oracle)	WTST	ORACLE EXE /M0011	1040

9.4. For Database time properties

Query:

```
SELECT property_name,
       property_value
  FROM database_properties
 WHERE UPPER(PROPERTY_NAME) LIKE '%TIME%'
 ORDER BY property_name;
```

The screenshot shows the Oracle SQL Developer interface. The top window is a Worksheet containing a SQL query:

```
SELECT property_name,
       property_value
  FROM database_properties
 WHERE UPPER(PROPERTY_NAME) LIKE '%TIME%'
 ORDER BY property_name;
```

The bottom window is a Query Result grid showing the results of the query. The columns are PROPERTY_NAME and PROPERTY_VALUE. The data includes properties like DBTIMEZONE, Flashback Timestamp TimeZone, and various NLS_TIMESTAMP_* and NLS_TIME_* properties.

PROPERTY_NAME	PROPERTY_VALUE
1 DBTIMEZONE	00:00
2 Flashback Timestamp TimeZone	GMT
3 NLS_TIMESTAMP_FORMAT	DD-MON-RR HH.MI.SSXFF AM
4 NLS_TIMESTAMP_TZ_FORMAT	DD-MON-RR HH.MI.SSXFF AM TZR
5 NLS_TIME_FORMAT	HH.MI.SSXFF AM
6 NLS_TIME_TZ_FORMAT	HH.MI.SSXFF AM TZR

10. Conclusion

In Conclusion, “Sports Aggregator” Information System addresses the exact requirements of users by providing the blend of user interaction along with an Expert Authentication. In this information era as there is huge piles of data regarding various sports it has become imperative to provide information in a focused manner based on the selectivity of the user and this is explicitly served by our information system.

Unique Selling Propositions:

- Both Expert and User Interactive System
- Content Specific to interests of User
- Provision for purchasing tickets

The post article function of this system ensures that the users of published articles will be recognized by other users of the system if the article is well crafted with critical insights.