Name: Abdulellah Hakim

Student Number: D12122837

Lecturer: Deirdre Lawless

Course: BSc. In Computer Science

Code: DT228

Module: Databases 1

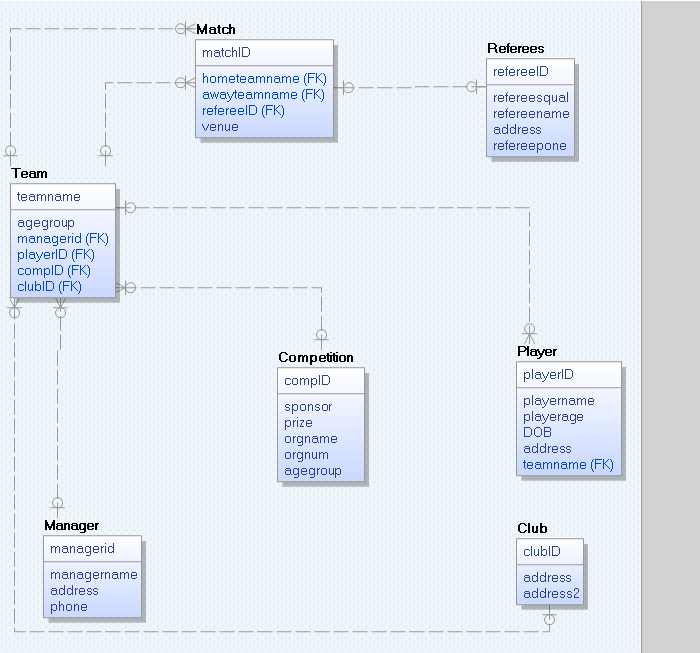
Submission Date: 05\12\2014

Title of Assignment: Database demonstration

**Introduction:**

The case study we had was about football competitions between various teams.

We made seven main entities here is a picture of the erwin diagram which was made to help understand what are the main entities and what is relationships type between them



**Entities**

1-Club, which holds the club ID and the exact address for it. This entity holds three columns of VARCHAR2 so it can hold number or characters ,because the address might have some numbers in it.ClubID is a primary key so it must be unique and it is sent to other tables as a foreign key.

2-Manager entity, which holds the manager ID, name, address, phone. In this entity all the attributes are VARCHAR2 except the phone which is a number(30).The primary key for this entity is the ManagerID which is sent to other tables as a foreign key

3-Competition, this entity holds many information about the competition ID, sponsor, prize, organization name, organization number, and the age group. All the attributes in this entity are VARCHAR2 even the age group. Later on it is discussed why we think agegroup is a VARCHAR2 and not a NUMBER.

4-Playerentity holds the playername, age, DOB, address and holds a foreign key (teamname) from another entity (team). (relationships will be discussed after the entities). DOB is a DATE

and playerage is a NUMBER

5-Refrees entity, this entity holds the referee qualification, name, address, phone and his ID. All this entity attributes are VARCHAR2(30) and the referee ID is a Primary key is sent to Match table as foreign key .

6-Team entity have many foreign keys which comes from another entities(managerID) from (manager),(CompID) which comes from (competition),(clubID) from (club).the primary key in this table is the teamname which is sent to the player table as foreign key. The last attribute in this table is the age group for the team. All of this table attributes are VARCHAR2(30).

7-Match entity, this is the entity which holds each match ID as primary key, and gets two team names from Team table( hometeamname) and (awayteamname) and gets the referee ID from the referees table

**Data types:**

Most of the attributes in the database are VARCHAR2(30) which means they can hold up to 30 number,character. We believe that length is enough for anything required in the database.

In the table competition, we had agegroup as VARCHAR because of the (-) element or if someone entered 20 To 25 for example the database will have no problem in handling this type of agegroup input.

Another datatype was used in the database was NUMBER(30) which iss used for NUMBER columns e.g(prize, phone numbers)

The last datatype we used was DATE which is for the players date of birth as this data type is used to store the players date of birth in a ‘DD-MMM-YYY’ format

**Relationship**:

The only type of relationship was used was a Non-identifying relationship which makes the attribute a foreign key in the other table so it does not have to be unique there. A list of the relationships which were used:

1-Non-identfying relationship which sends one club ID to one team. Relationship name is: club\_team\_fk

2-non-identfying relationship sends one manager ID to one or more team. Relationship name is: manager\_team\_fk

3-non-identfying relationship sends one Competition ID to one or more teams. Relationship name is: comp\_team\_fk

4-Non-identfying relationship sends one teamname to one or many players. Relationship name is: player\_team\_fk

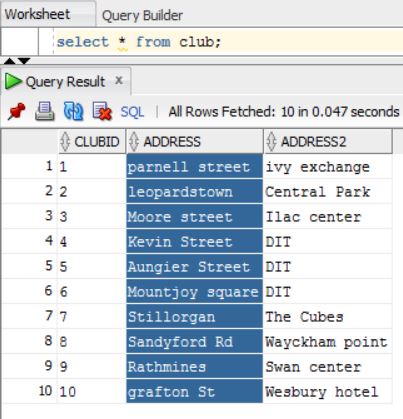
5-Non-idntfying relationship sends one refereeID to one match. Relationship name is : ref\_match\_fk

6-one relationships which sends one teamname to one or many matchs. Relationship name is : team\_match\_fk

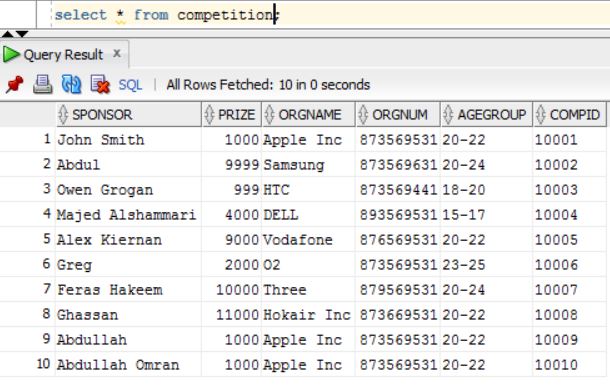
7-Non-idnetfying relationship which sends one teamname to one or many matchs. Relationship name is: team1\_match\_fk

**Tables**:

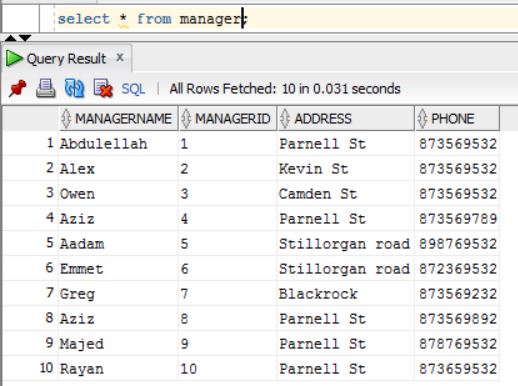
Club table:



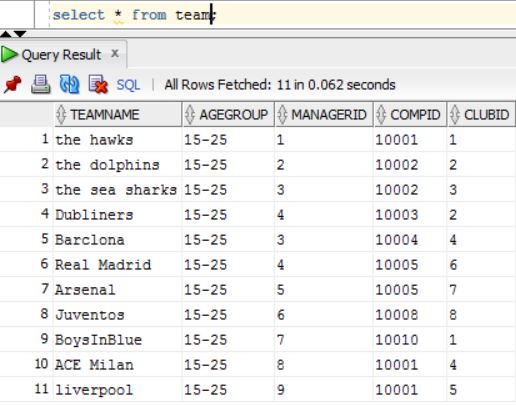
Competition table:



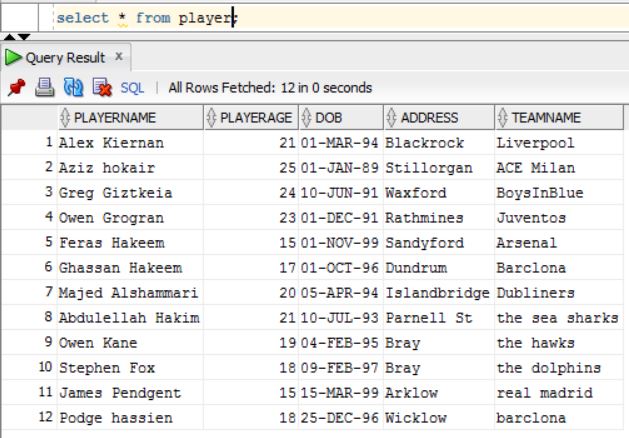
Manager table:



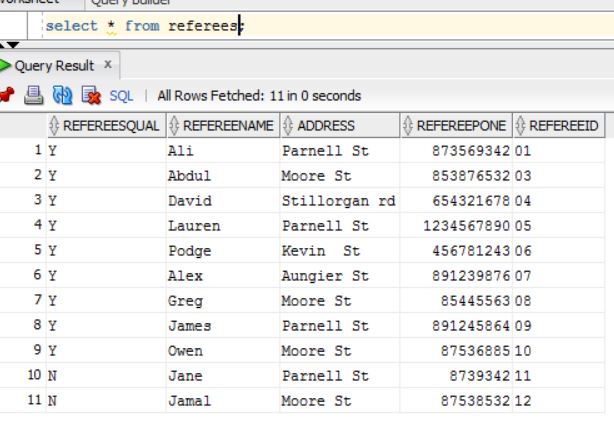
Team table:



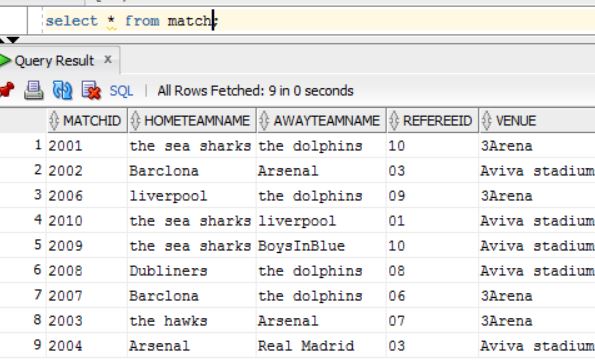
Player table:



Referees table:

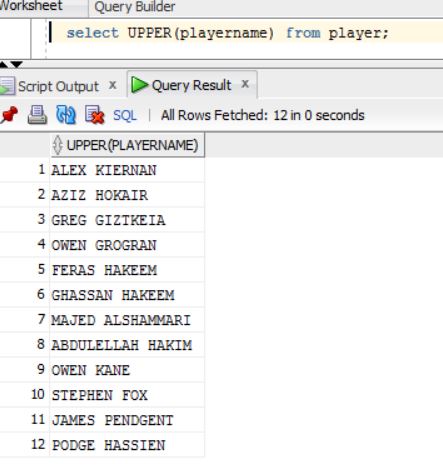


Match table:

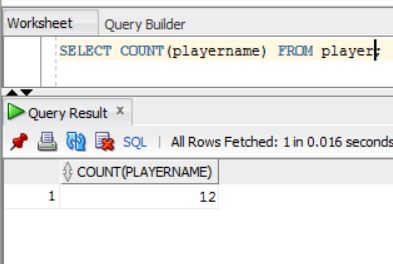


**Queries output:**

1-Single row function we have used UPPER which changes the selected columns data into UPPER case. Here is the output when selecting player names using the UPPER() function from player table

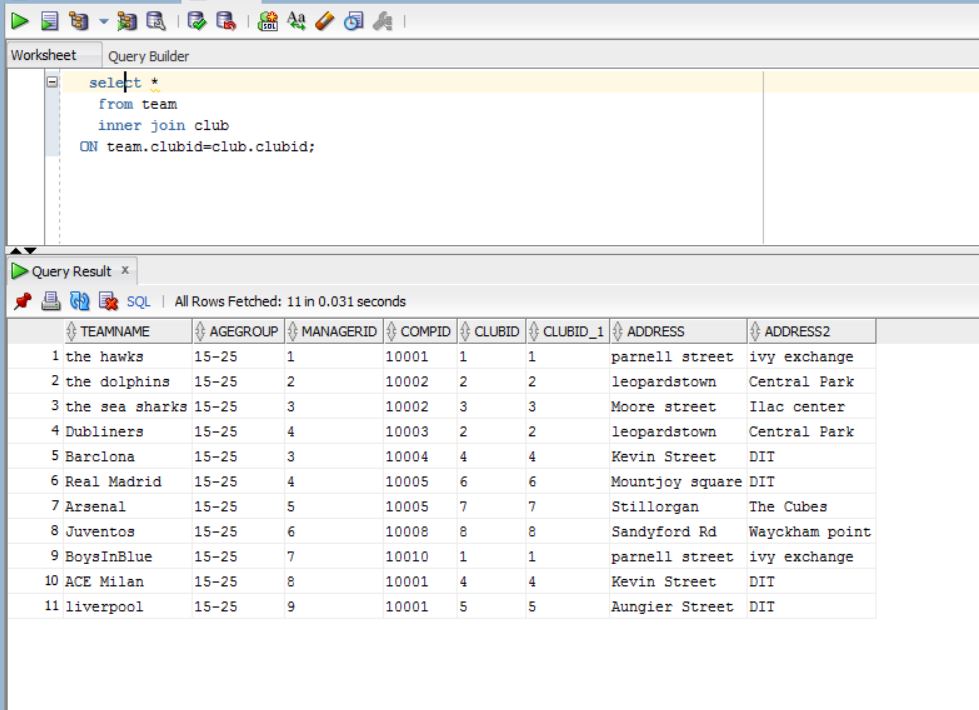


2-aggregating function, we use the COUNT function, which counts how many rows the select statement used in this example we used COUNT() on playernames from the players table and it printed out how many player we had

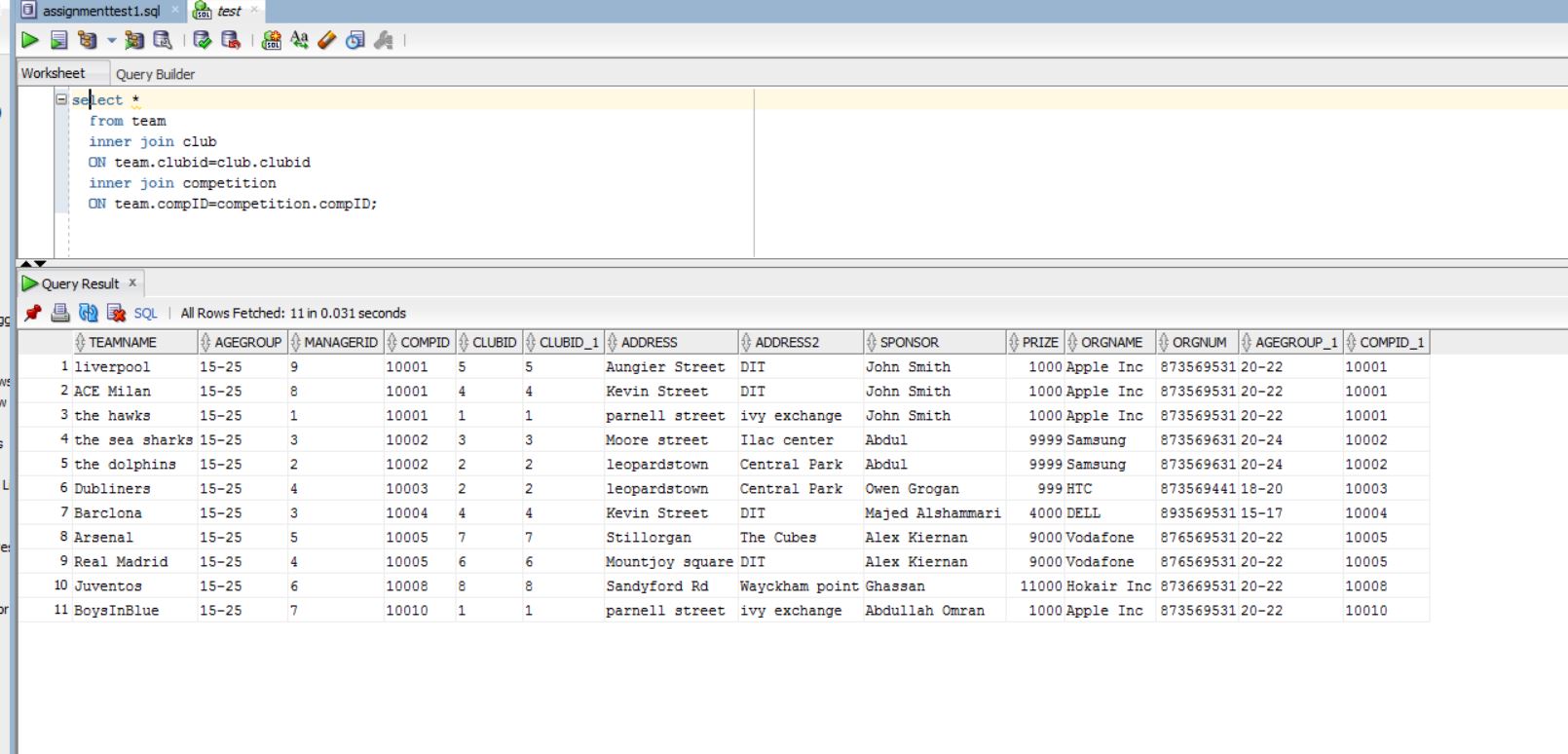


3- INNER JOIN on two tables, we used inner join on all the data in team, joined it with club table where the clubID on team matches the clubID on club and print all the

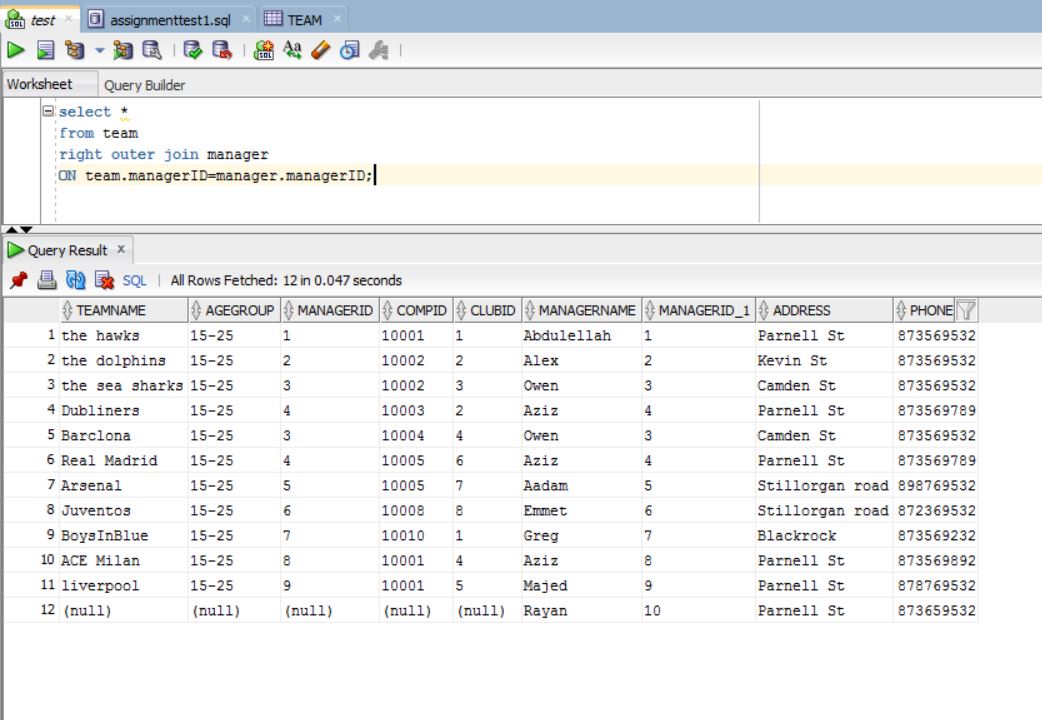
Prints all data in team and all data from club(address1,address2)



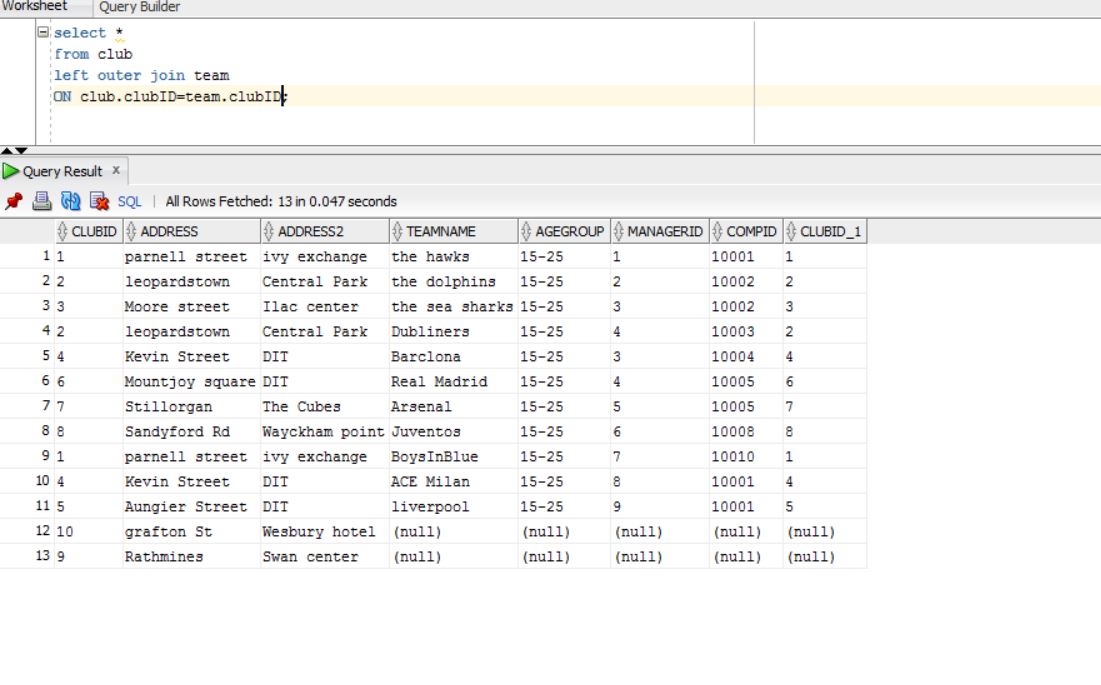
4-inner join on three, selects all data from team and add all data from club to it (where clubID on club matches clubID on team), also join Competition( compID in team matches compID in competition)



5-right outer join, selects all the data from team and right outer join manager, so it brings all the manager details if any managerID matches it on team.managerID

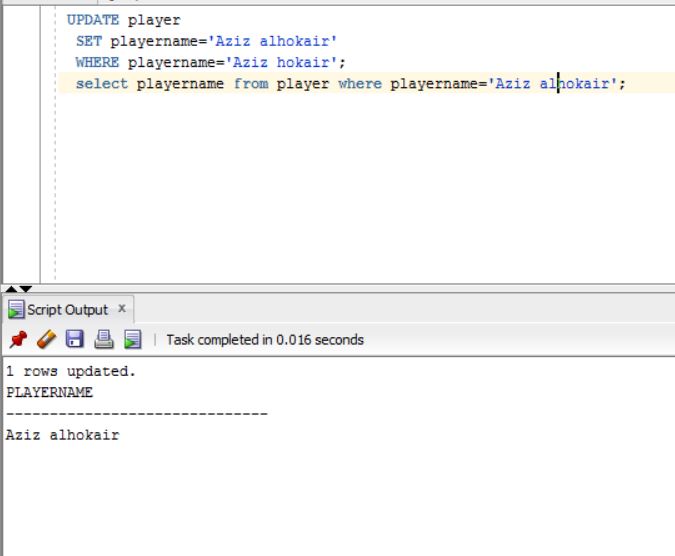


6-left outer join, we selected all the data from club and we check the associated team name where the club ID matchs the club ID in the team.

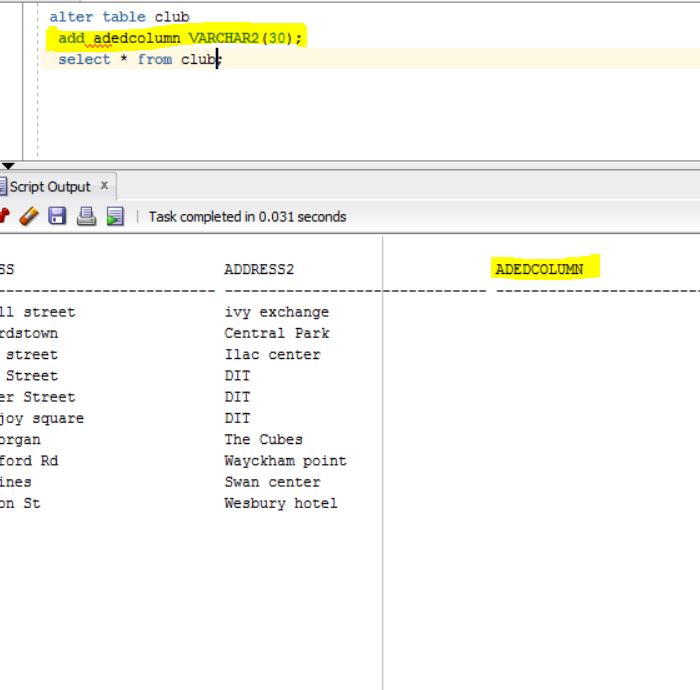


**Alteration:**

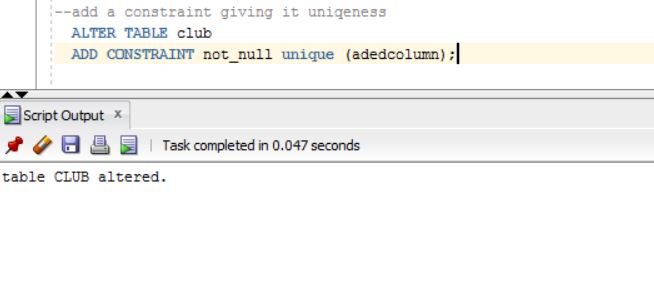
1-Update selected data in this example we updated the player Aziz hokair to Aziz alhokair and we used select statement to show it.



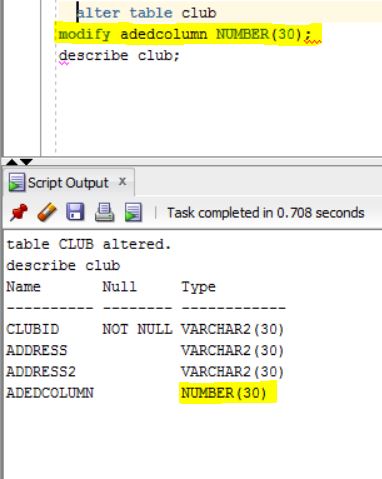
2-Alter table and add a column, we added a column to the club table and we named it adedcolumn with a VARCHAR2(30) datatype



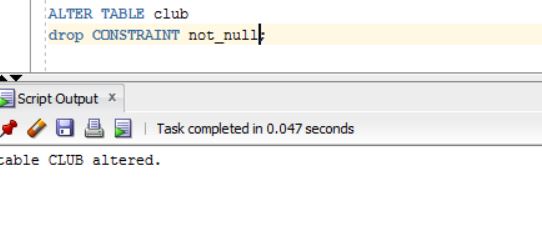
3-add constraint which gives adedcolumn unique value:



4-modfiy a column, in this example we modified adedcolumn to from VARCHAR2 to NUMBER



5-before we do a drop column we need to drop the constraint in it first by using drop constraint statement



6:after droping the constraint we can drop the column

Note the column is gone

