**AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH (AIUB)**

**FACULTY OF SCIENCE & TECHNOLOGY**

A picture containing calendar

Description automatically generated

Course Title

**INTRODUCTION TO DATABASE**

**TITLE**

**Club Management System**

**Supervised By**

MD Sajid Bin Faisal

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| TOPICS | *Page no*. |
| 1. Title Page | **1** |
| 1. Table of Content | **2** |
| 1. Introduction | **3** |
| 1. Case Study | **3** |
| 1. ER Diagram | **4** |
| 1. Normalization | **5-6** |
| 1. Finalization | **7** |
| 1. Table Creation | **8-19** |
| 1. Data insertion | **20-25** |
| 1. Query testing | **26-34** |
| 1. Database connection | **35-39** |
| 1. Conclusion | **40** |

1. **Introduction**

The Football Club Management Database System is a comprehensive solution for efficient club operations. It centralizes player details with other employees as well as other information. The system enhances decision-making of the administration or other body of the club t. It ensures compliance, transparency, and security, providing flow of data of the club and predict some future outcome.

1. **Case Study**

In the world of club football, a club has a unique name, license no, club valuation (considered as 5 times multiple of total club asset), and country of origin. The club might participate in multiple leagues based on their country of origin for the local leagues. But for participating in the regional leagues, they must be ranked top in their local leagues. A club management system stores information about the club, which leagues they are eligible to play, and their present rank in every individual league.

In a club, there must be many players. A player might play in multiple positions. The system stores the details of the players like position, jersey number, physical state, contract expiration date, number of matches. A club might have many organizations to become partners with. Some of them are sponsors, technology partners, kit suppliers, merchandise and so on. The system stores the name, category, signing date, expiration date, and fund details of the partners.

The players are also employees of the club. The system also stores the information about other employees like the staff and admins. There can be diverse types of staff such as coaching staff, medical staff, working staff. In the admins there must be one team director, one team manager, and other officials. In the management system, it stores employee’s ID, salary, name, date of birth, address, phone number, and joining date.

A club has personal assets such as team vehicles, stadium, exercise equipment, residences. In the management system it always stores the asset id, name, category, quantity & value of the assets.

One club might have many glories in its history. As the count of their achievement, the system also stores information about all the achievements of the club. In the system's database, there is the category of the awards, name of achievement, achievement date, event description, and place of the event.

A club at least owned by one owner. But there can also be multiple owners or investors. The database system also stores information about the shareholders of the club with the name and percentage of share in the club & contact information. Investors can be individuals, or it also can be any company.

1. **ER Diagram**

A diagram of a company

Description automatically generated

Image 3.1:ER Diagram of Case Study

1. **Normalization**
   1. **Owns**
      1. UNF

oShare, oName, oType, oContact, cName, cLicNo, cValuation, cOrigin

* + 1. 1NF

oshare, oName, oType, oContact, cName, cLicNo, cValuation, cOrigin

* + 1. 2NF

oShare, oName, oType, oContact(pk)

cName, cLicNo(pk), cValuation, cOrigin

cLicNO(pk),oContact(fk)

* + 1. 3NF

Same as 2NF

* 1. **Participates** 
     1. UNF

cName, cLicNo, cValuationk, cOrigin, LigName, LigDiv, cRanking, LigOrigin

* + 1. 1NF

cName, cLicNo, cValuationk, cOrigin, LigName, LigDiv, cRanking, LigOrigin

* + 1. 2NF
       1. LigName(pk), LigDiv, cRanaking, LigOrigin
       2. cName, cLicNo(pk), cValuation, cOrigin
       3. LigName(pk), cLicNo(fk)
    2. 3NF

Same as 2NF

* 1. **Sponsers**
     1. UNF

cName, cLicNo, cValuation, cOrigin, cValuation, cOrigin, contStatus, expDate, signingDate, prtnrName,Category, fundDetails

* + 1. 1NF

cName, cLicNo, cValuation, cOrigin, cValuation, cOrigin, contStatus, expDate, signingDate, prtnrName,Category, fundDetails

* + 1. 2NF
       1. prtnrName(pk), contStatus, expDate, signingDate, Category, fundDetails, cLicNo(fk).
       2. cName, cLicNo(pk), cValuation, cOrigin.
    2. 3NF

Same as 2NF

* 1. **Has**
     1. UNF

cName, cLicNo, cValuation, cOrigin, astName, astID, astCate, astQuant, astValue

* + 1. 1NF

cName, cLicNo, cValuattion, cOrigin, astName, astID,astCate, astQuant, astValue

* + 1. 2NF
       1. astName, astID(pk), astCate, astQuant, astValue, cLicNo(fk)
       2. cName, cLicNo(pk), cValuation, cOrigin
    2. 3NF

Same as 2NF

* 1. **Achieved**
     1. UNF

achName, achCate, achDesc, achDate, achPlace, cName, cLicNo, cValuation, cOrigin

* + 1. 1NF

achName, achCate, achDesc, achDate, achPlace, cName, cLicNo, cValuation, cOrigin

* + 1. 2NF
       1. achName(pk), achCate, achDesc, achDate, achPlae, cLicNou(fk)
       2. cName,cLicNo(pk),cValuation, cOrigin, LigName,LigDiv
    2. 3NF

Same as 2NF

* 1. **Works**
     1. UNF

eID, eName, eNationality, eJoinDate, eSal, eDOB, ePhnNo, eAddress, aDept, aDesignation, pNumberMatch, pExpDate, pPosition, pJerseyNo, aPhyState, sDept, SDesignation, cLicNo, cValuation, cName, cOrigin

* + 1. 1NF

eID, eName, eNationality, eJoinDate, eSal, eDOB, ePhnNo, eAddress, aDept, aDesignation, pNumberMatch, pExpDate, pPosition, pJerseyNo, aPhyState, sDept, SDesignation, cLicNo, cValuation, cName, cOrigin

* + 1. 2NF
       1. eID(pk),eName,eNationality,eJoinDate,eSal,eDOB,ePhnNo,address, cLicNo(fK). [employee]
       2. eID(pk),pNumbeMatch,pExpDate,pPostion,pJerseyNo, pPhyState. [player]
       3. eID(Pk),sDept, sDesignation. [staff]
       4. eID(pk),aDept, aDesignation. [admin]
       5. cName, cLicNo(pk), cValuation, cOrigin.[club]
    2. 3NF

Same as 2NF

1. **Finalization**
2. oName, oShare, oType, oContact(pk). [owner]
3. cName, cLicNo(pk), cValuation, cOrigin. [club]
4. cLicNo(pk), oContact(fk). [owns]
5. prtnrName(pk), contStatus, expDate, signingDate, Category, fundDetails, cLicNo(fk). [organization]
6. ~~cName, cLicNo(pk), cValuation, cOrigin. [club]~~
7. ligName(pk), ligDiv, cRanking, LigOrigin. [league]
8. ~~cName, cLicNo(pk), cValuation, cOrigin. [club]~~
9. cLicNo(pk), ligName(fk). [participates]
10. astName,astId(pk),astCate,astQuant,astValue,cLicNo(fk). [asset]
11. ~~cName, cLicNo(pk), cValuation, cOrigin. [club]~~
12. achName(pk), achCate, achDesc, achDate, achePlace, cLicNo(fk). [achievement]
13. ~~cName, cLicNo(pk), cValuation, cOrigin. [club]~~
14. eID(pk), eName,eNationality, eJoinDate,eSal,eDOB,ePhnNo,address, cLicNo(fK). [employee]
15. eID(pk), pNumbeMatch, pExpDate, pPostion, pJerseyNo, pPhyState. [player]
16. eID(Pk), sDept, sDesignation. [staff]
17. eID(pk), aDept, aDesignation. [admin]
18. ~~cName, cLicNo(pk), cValuation, cOrigin.[club]~~
19. **Table Creation**
    1. **Owner**

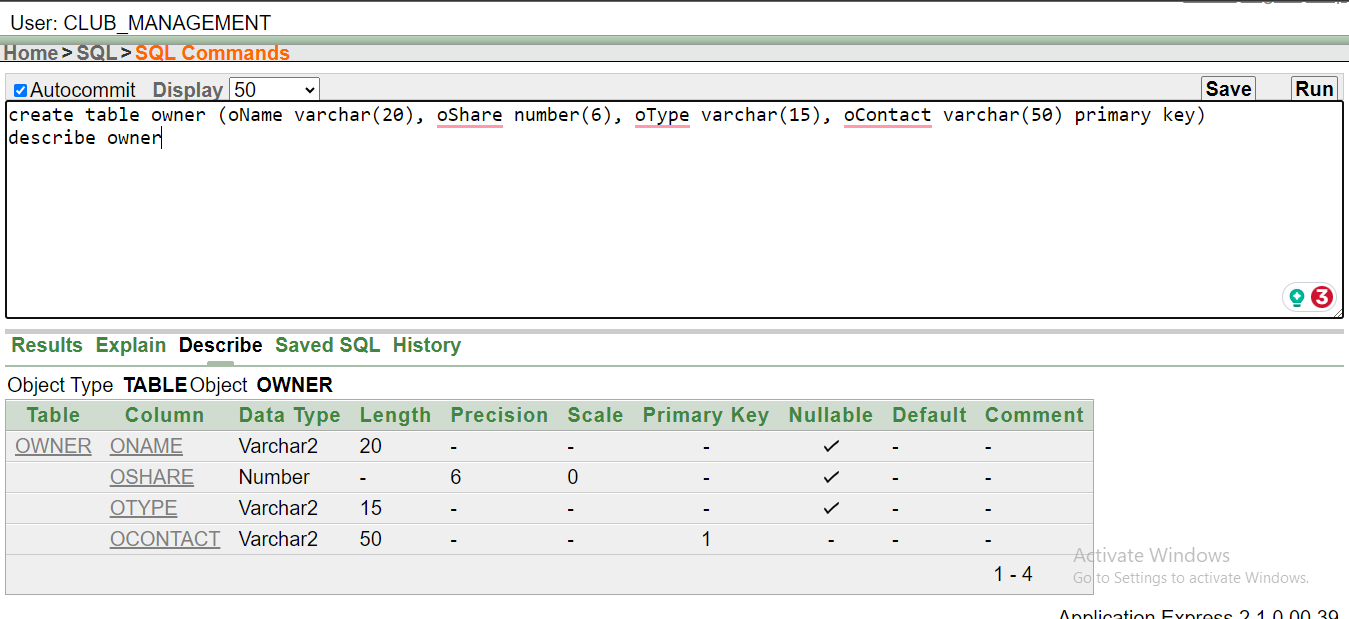


Image 3.1: commnand to create owner table and description of the created table

* 1. **Club**

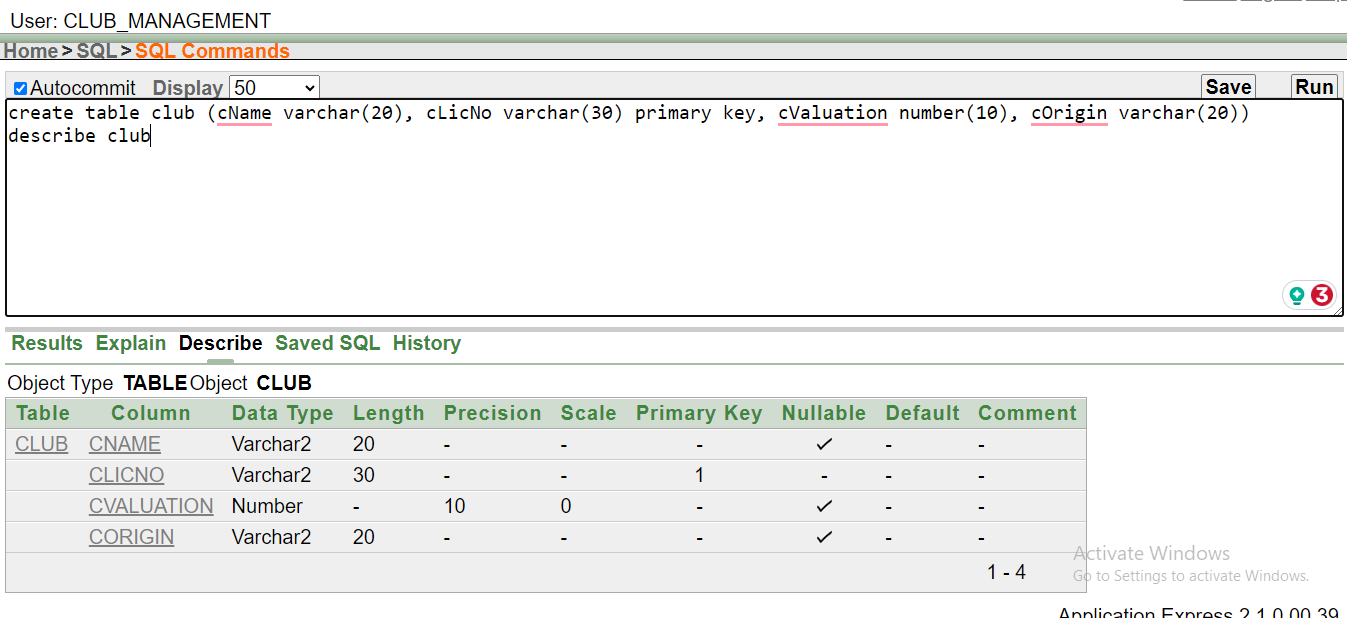


Image 6.2.1: commnand to create owner club and description of the created table

* 1. **Owns**

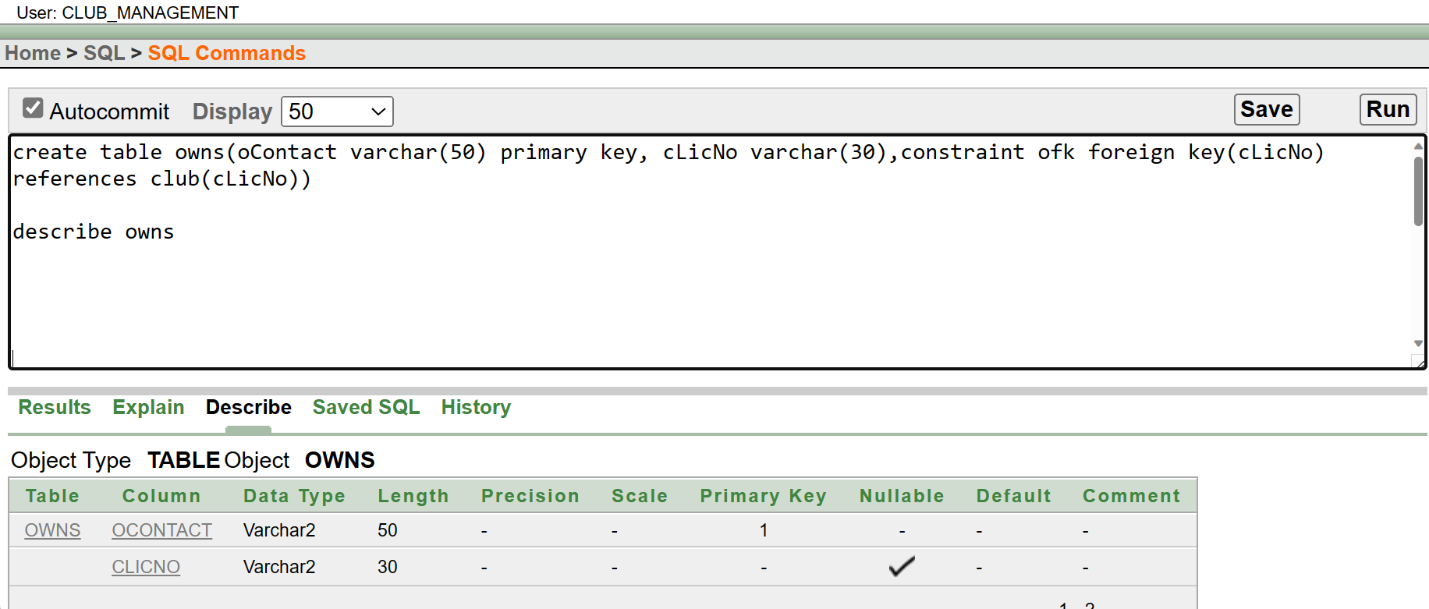


Image 6.3.1: commnand to create owns table and description of the created table

* 1. **Organization**

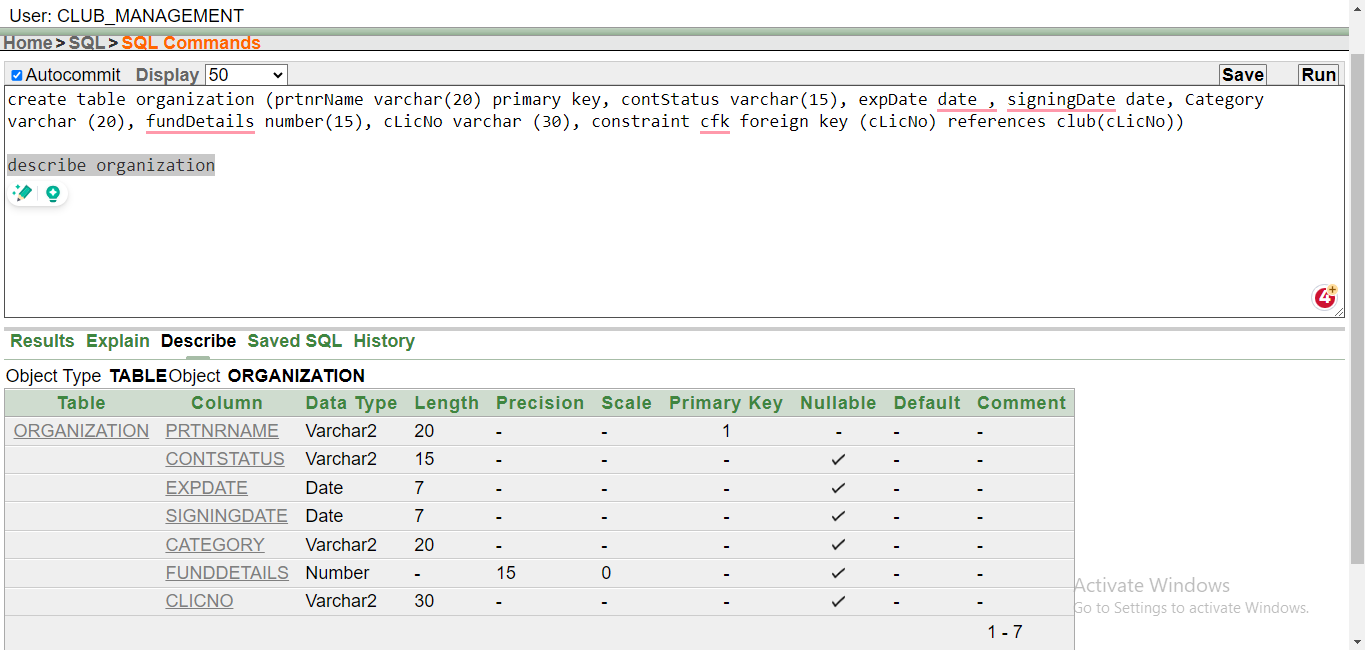


Image 6.4.1: commnand to create organization table and description of the created table

* 1. **League**

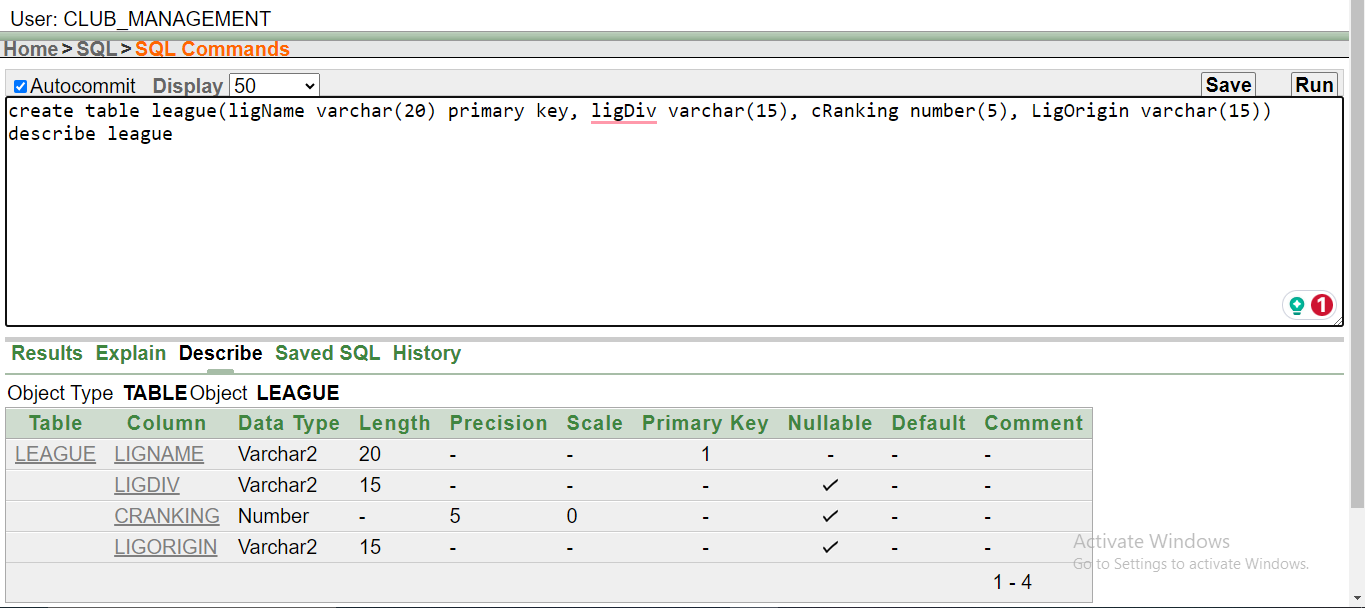


Image 6.5.1: commnand to create league table and description of the created table

* 1. **Participates**

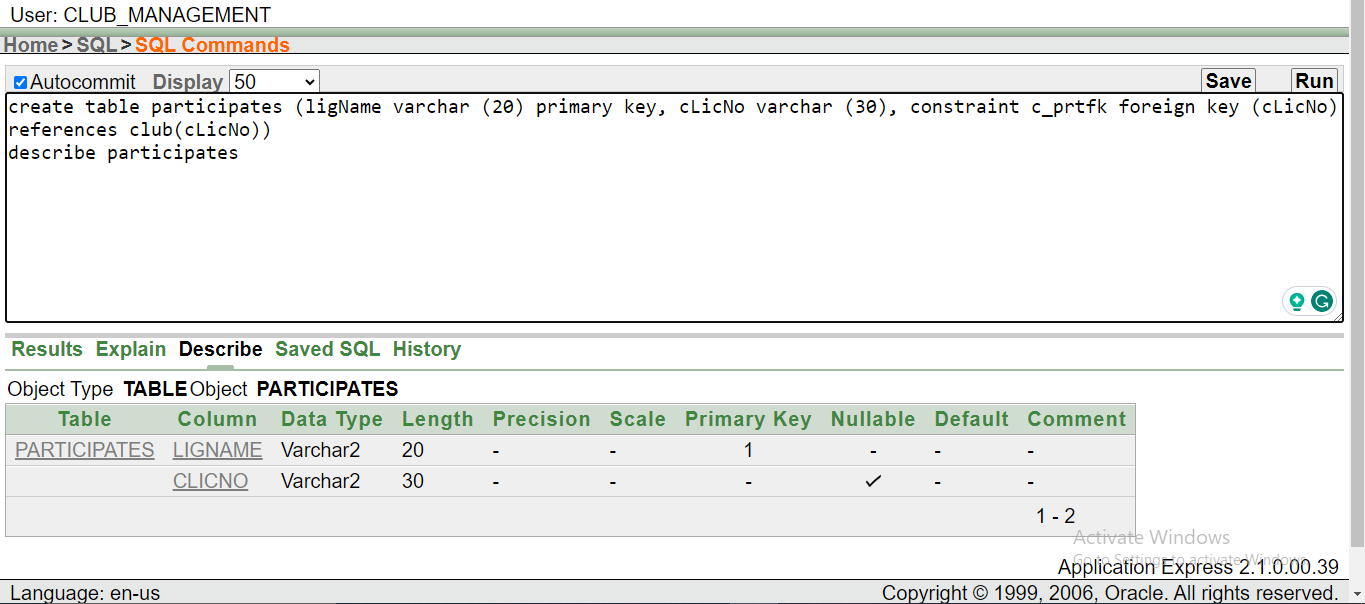


Image 6.6.1: commnand to create participates table and description of the created table

* 1. **Asset**

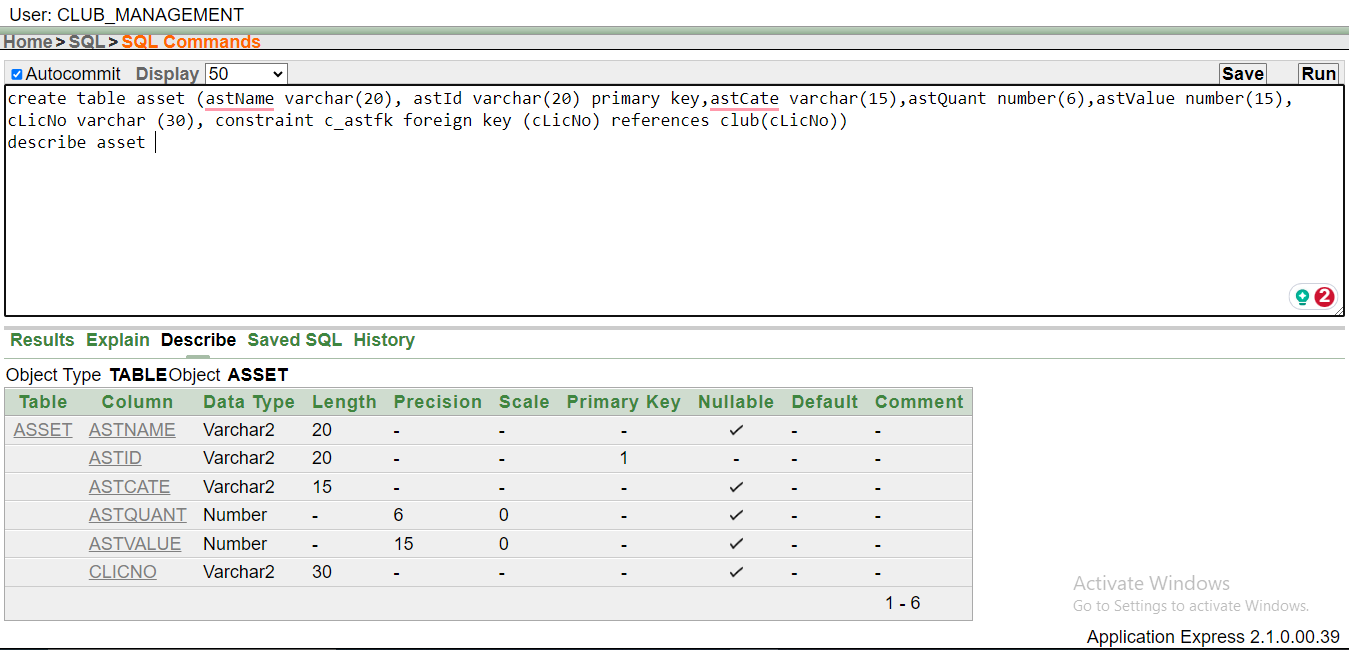


Image 6.7.1: commnand to create asset table and description of the created table

* 1. **Achievement**

A screenshot of a computer

Description automatically generated

Image 6.8.1: commnand to create achievement table and description of the created table

* 1. **Employee**

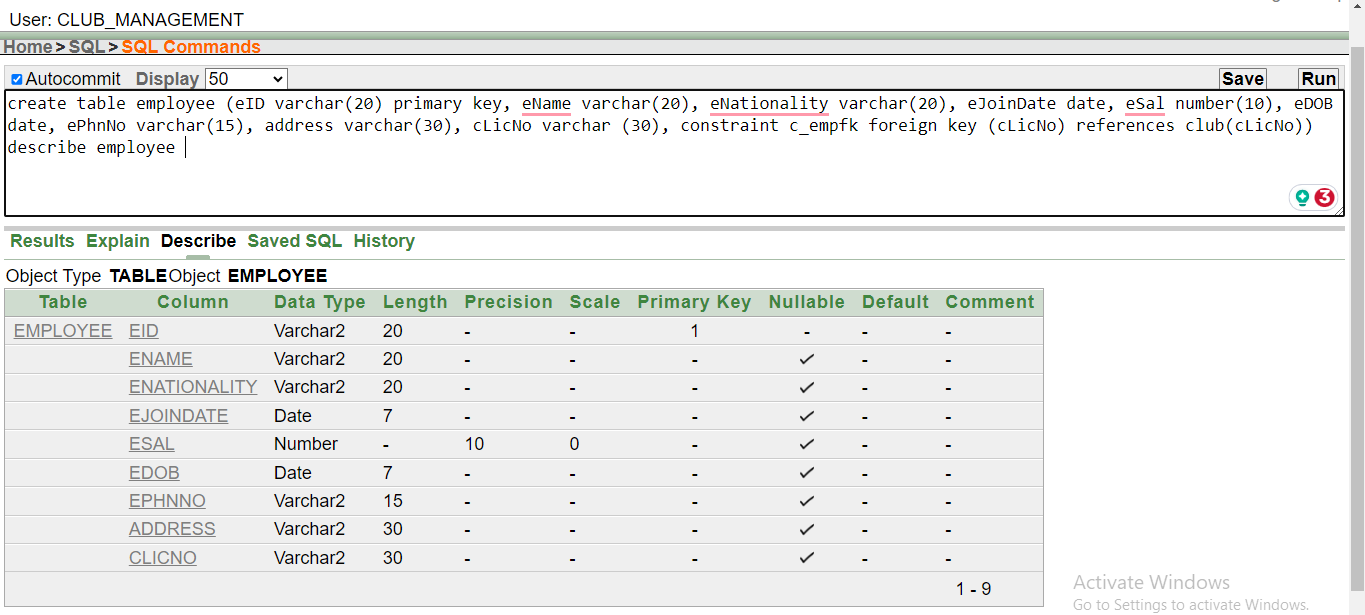


Image 6.9.1: commnand to create employee table and description of the created table

* 1. **Player**

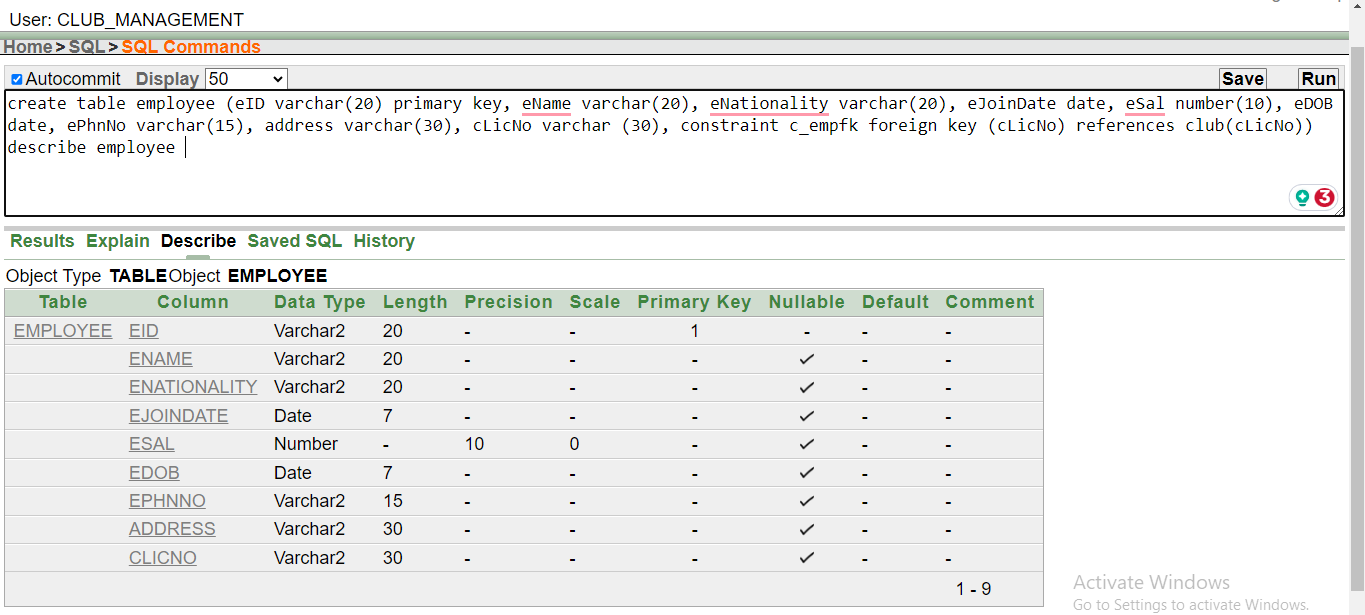


Image 6.10.1: commnand to create player table and description of the created table

* 1. **Staff**

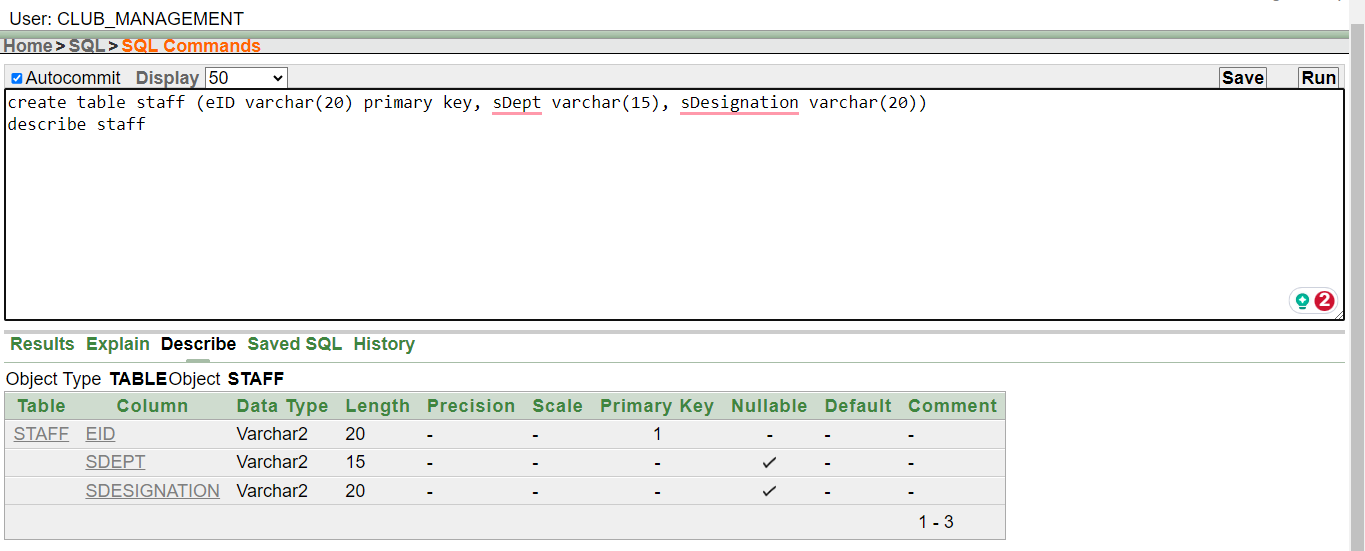


Image 6.11.1: commnand to create staff table and description of the created table

* 1. **Admin**

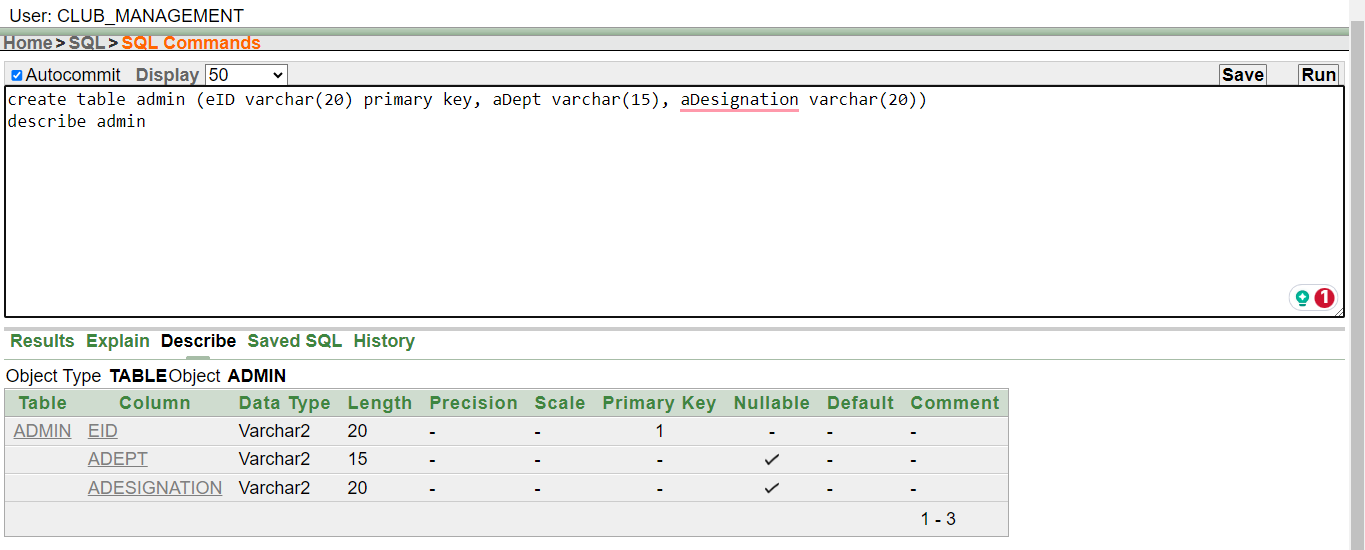


Image 6.12.1: commnand to create own table and description of the created table

1. **Value Insertion**
   1. **Owner** A screenshot of a computer

      Description automatically generated

Image 7.1.1: owner table values.

* 1. **Club**

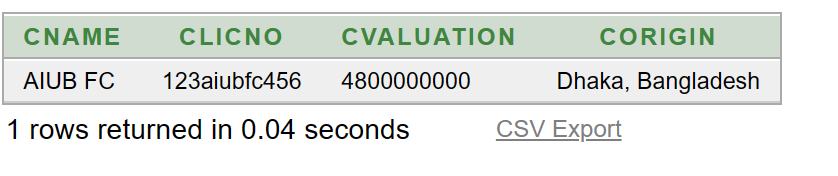


Image 7.2.1: club table values.

* 1. **Owns**

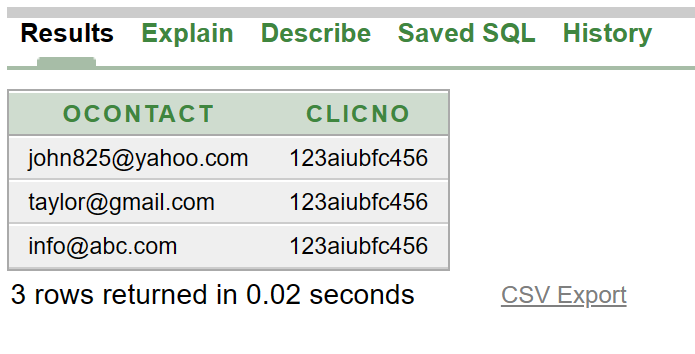


Image 7.3.1: Owns table values.

* 1. **Organization**

****

Image 7.4.1: Organization table values..

* 1. **League**

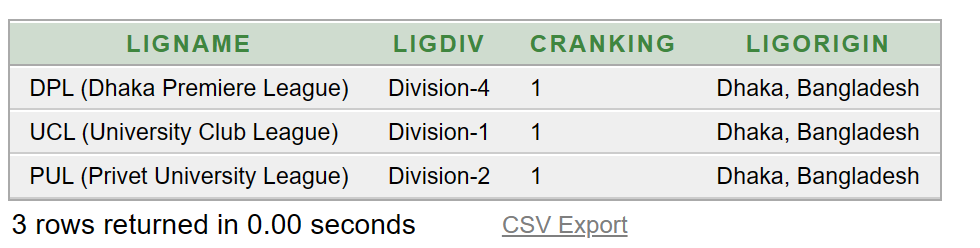


Image 2: League table values.

* 1. **Participates**

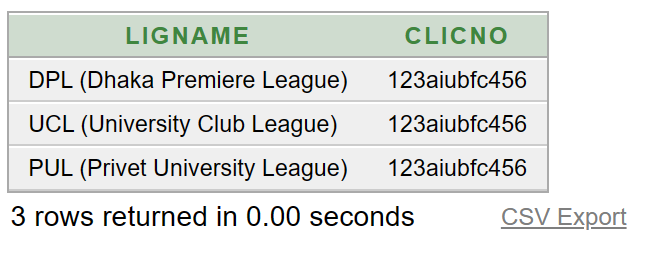


Image 7.6.1: Participates table values.

* 1. **Asset**

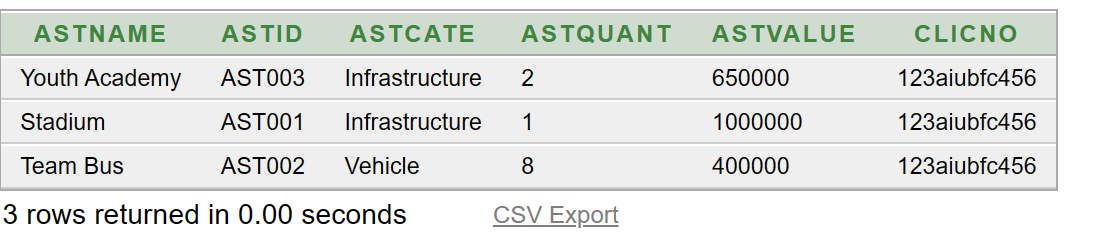


Image 7.7.1: Asset table values.

* 1. **Achievement**



Image 7.8.1: Achievement table values.

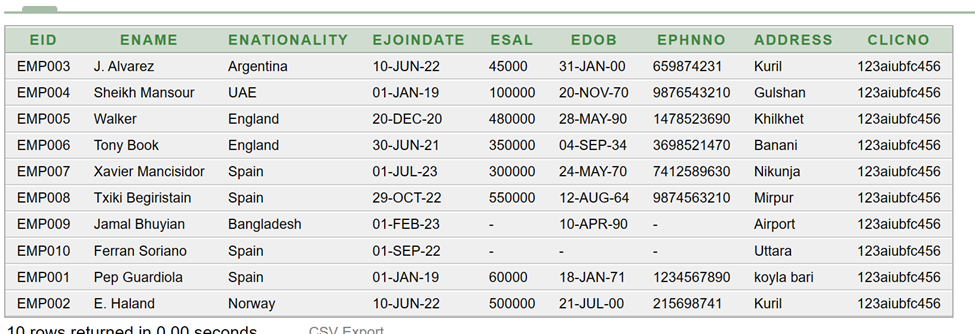
* 1. **Employee** 

Image 7.9.1: Employee table values.

* 1. **Player**

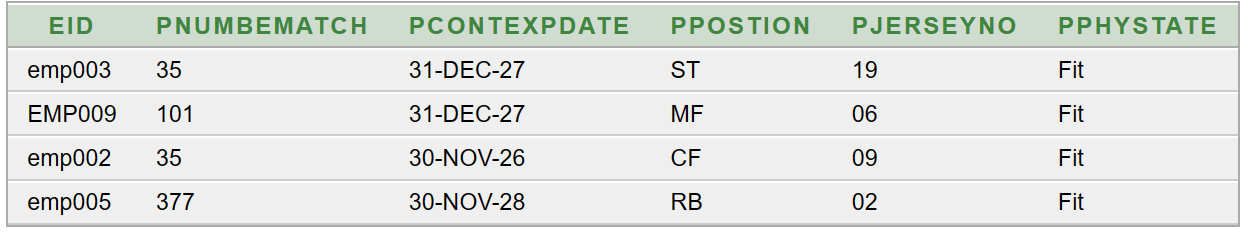


Image 7.10.1: Player table values.

* 1. **Staff**

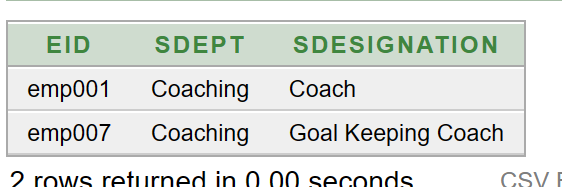


Image 7.11.1: Staff table values.

* 1. **Admin**

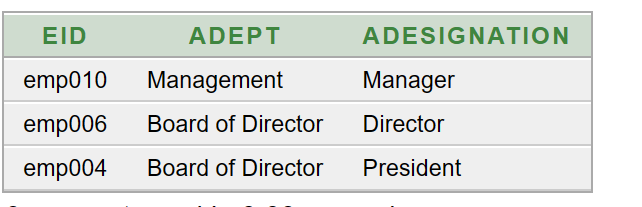


Image 7.12.1: Admin table values.

1. **Query Test**
   1. **Simple Query**

Question:

Retrieve the employee id, names, date of birth and nationality from employee table.

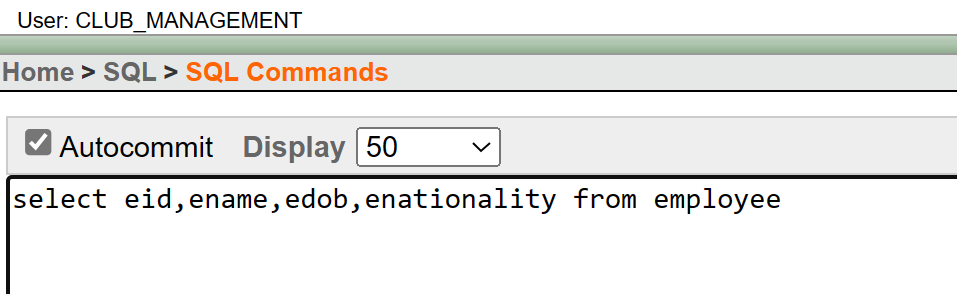


Image 8.1.1: Command to make simple query.

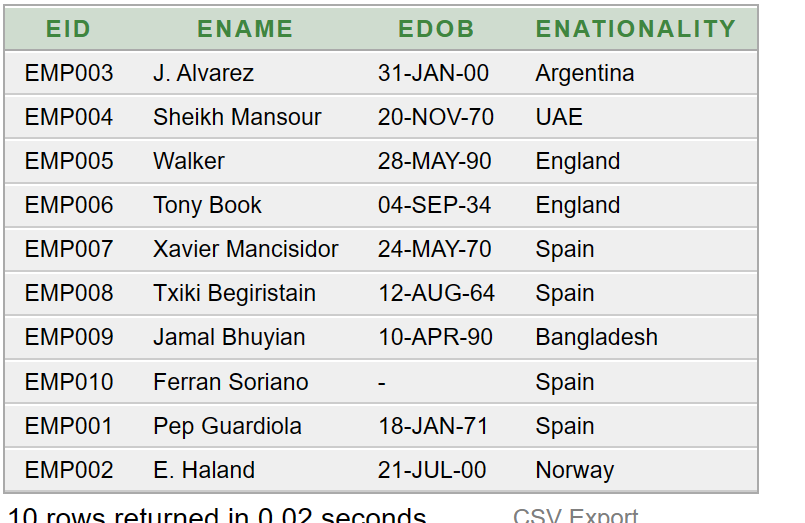


Image 8.1.2: Simple Query result

* 1. **Single Row function**

Question:

Round the employee’s salary of the employee with 5% bonus.

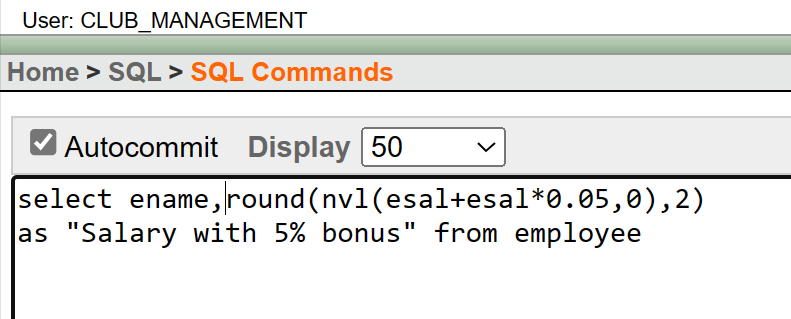


Image 8.2.1: command for single row function

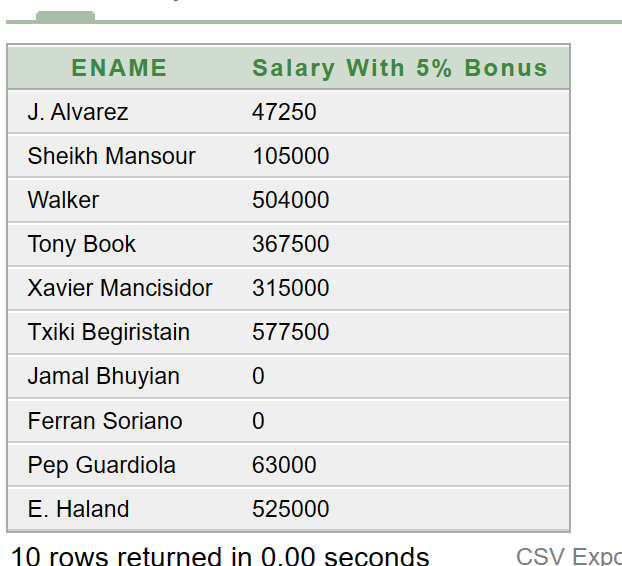


Image 8.2.1: Single Row Function result

* 1. **Aggregation Query**

Query:

Show the max valuation of the asset table.

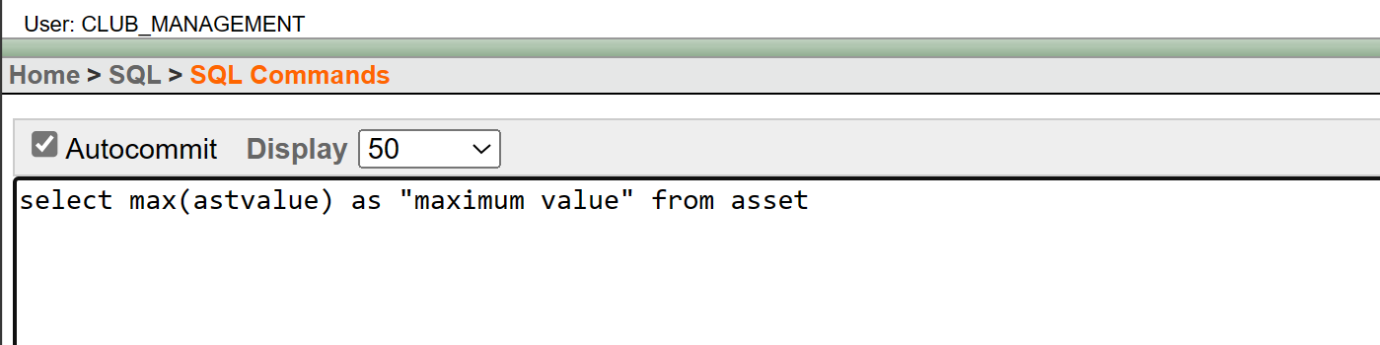


Image 8.3.1: command for aggregation query.

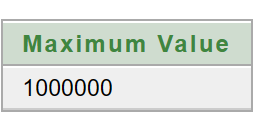


Image 8.3.2: Aggregation Query result

* 1. **Single Row Subquery**

Query:

Show the players with their ID no, name and date of birth who have same nationality as EMP007.

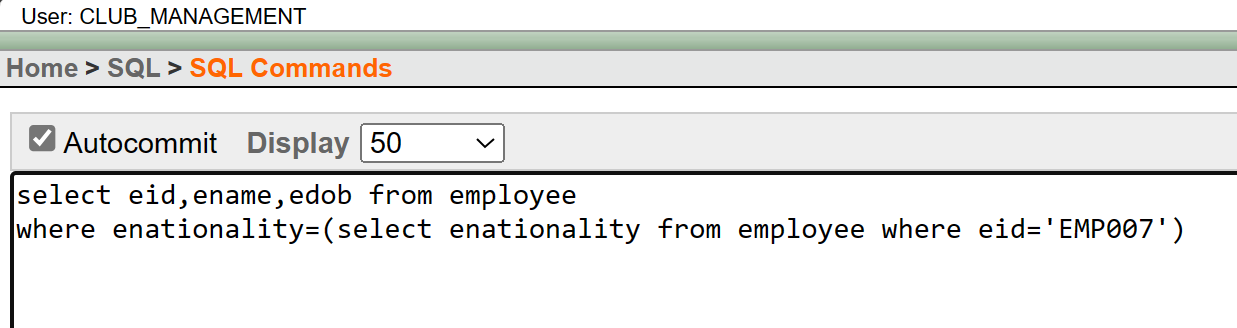


Image 8.4.1: command for single row subquery

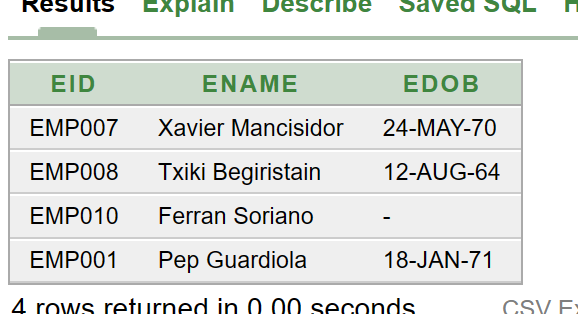


Image 8.4.2: Single Row Subquery result

* 1. **Multiple Row Subquery**

Query:

Show the prtnrname, category who have less fund than average asset value.

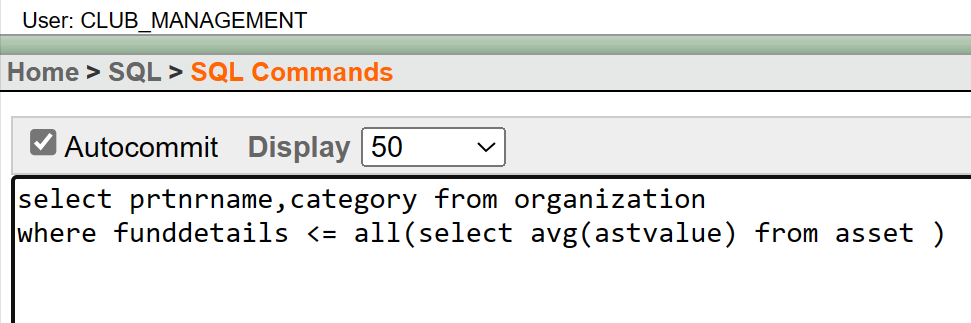


Image 8.5.1: command for multiple row subquery

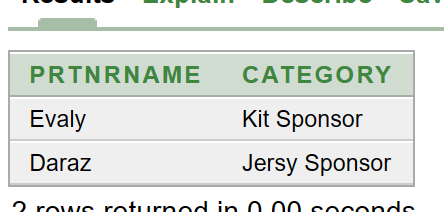


Image 8.5.2: multiple row subquery result

* 1. **Joining**
     1. **Non-Equijoin**

Query:

Show the players with their name, physical state and position.

A screenshot of a computer

Description automatically generated

Image 8.6.1.1: command for non-equijoin

A screenshot of a computer

Description automatically generated

Image 8.6.1.2: Non-Equijoin result

* + 1. **Outer join**

Query:

Show all employee’s name, nationality, and their designation in administration.

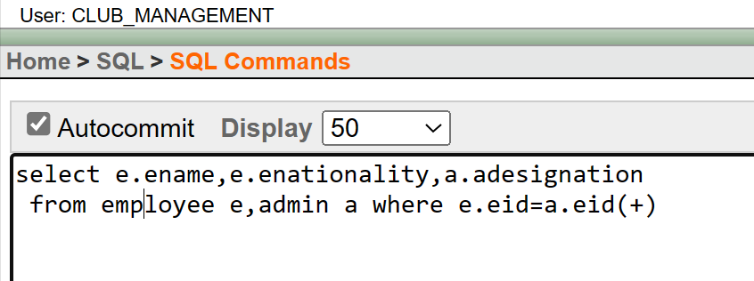


Image 8.6.2.1: command for outer join



Image 8.6.2.2: outer join result

* 1. **View**

Query 1:

Create a simple view of employee name, salary, and their age.

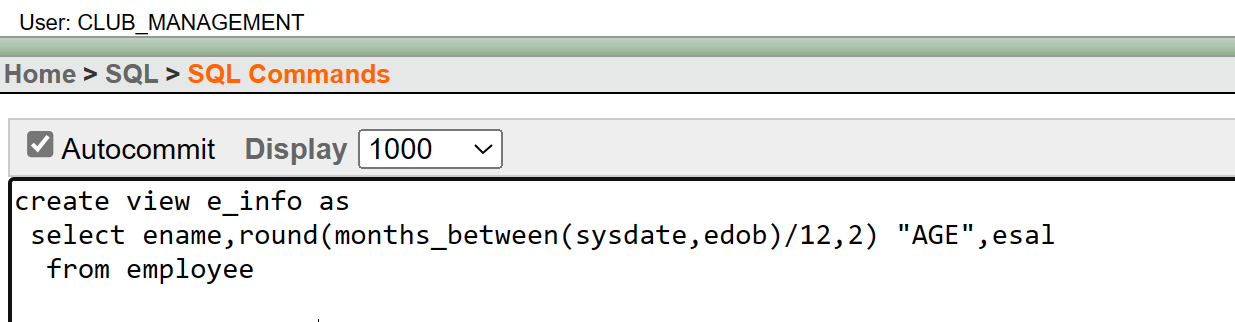


Image 8.7.1: command for simple view

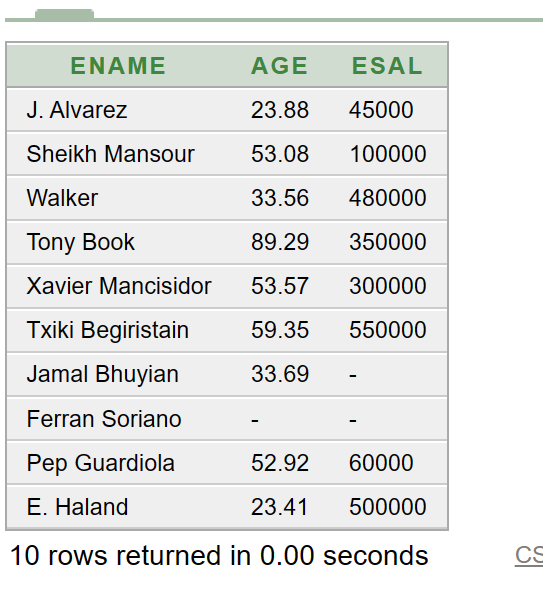


Image 8.7.2: simple view result

Query 2:

Create a complex view as player info of player name, join date, position, and physical state.

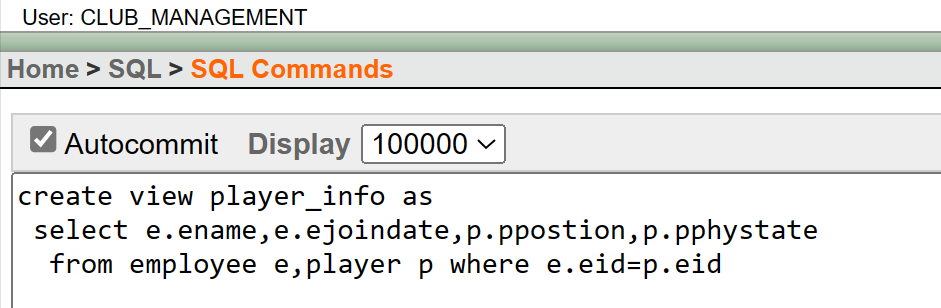


Image 8.7.3: command for complex view

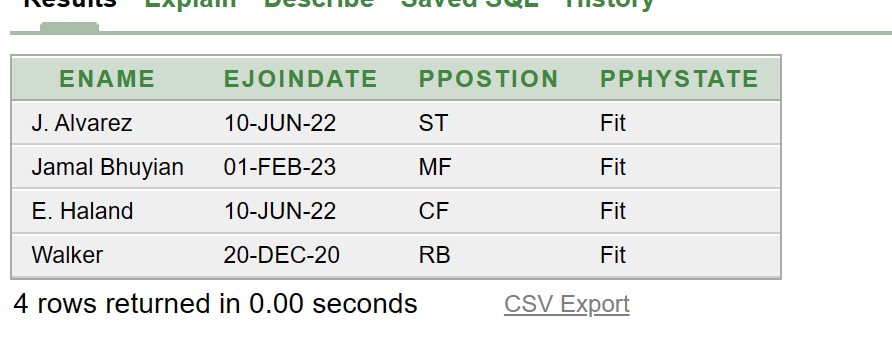


Image 8.7.4: Complex view result

1. **Database Connection**
   1. **MD.LUTFUL KABIR (22-49135-3)**
2. **Needed tools.**

**MySQL Java Connector**: Downloaded the MySQL Java Connector JAR file from the official MySQL website. This connector enables Java programs to interact with MySQL databases.

**XAMPP**: XAMPP is a development environment that includes Apache, MariaDB (MySQL replacement), Perl, and PHP. Install XAMPP to create a local server for testing the database operations.

1. **Setting up the environment**

* Installed XAMPP and started both the Apache and MySQL services through the XAMPP control panel. Accessed the MySQL admin panel to manage my databases.
* Launched the MySQL admin panel.
* Created a new database named club\_management that will be used in this project.
* Inside the newly created database, one table named employee with relevant columns to store data, also Defined data types.
* Populated the tables with relevant data to perform operations on.

A screenshot of a computer

Description automatically generated

Image 9.1.1: created table in SQL PhpMyAdmin

1. **Writing the java code**

* Chosen an Integrated Development Environment (IDE) of my choice. I used visual studio code for Java development.
* In my Java code, loaded the MySQL Java Connector driver by importing the JAR file into my project and using the **Class.forName()** method to register it.
* Used the **DriverManager.getConnection()** method to establish a connection to my MySQL database by providing the appropriate URL, username, and password. My username was root and there was no password set.
* After establishing the connection, created a **Statement** or **CreatedStatement** object to execute SQL queries.
* Used the created statement to execute SQL queries like SELECT, INSERT, UPDATE, or DELETE. Captured the necessary results.
* Used the **ResultSet** object to retrieve and process the data.
* Performed necessary operations on the data retrieved from the database.

A screen shot of a computer program

Description automatically generated

Image 9.1.2: java connection with SQL

A screenshot of a computer program

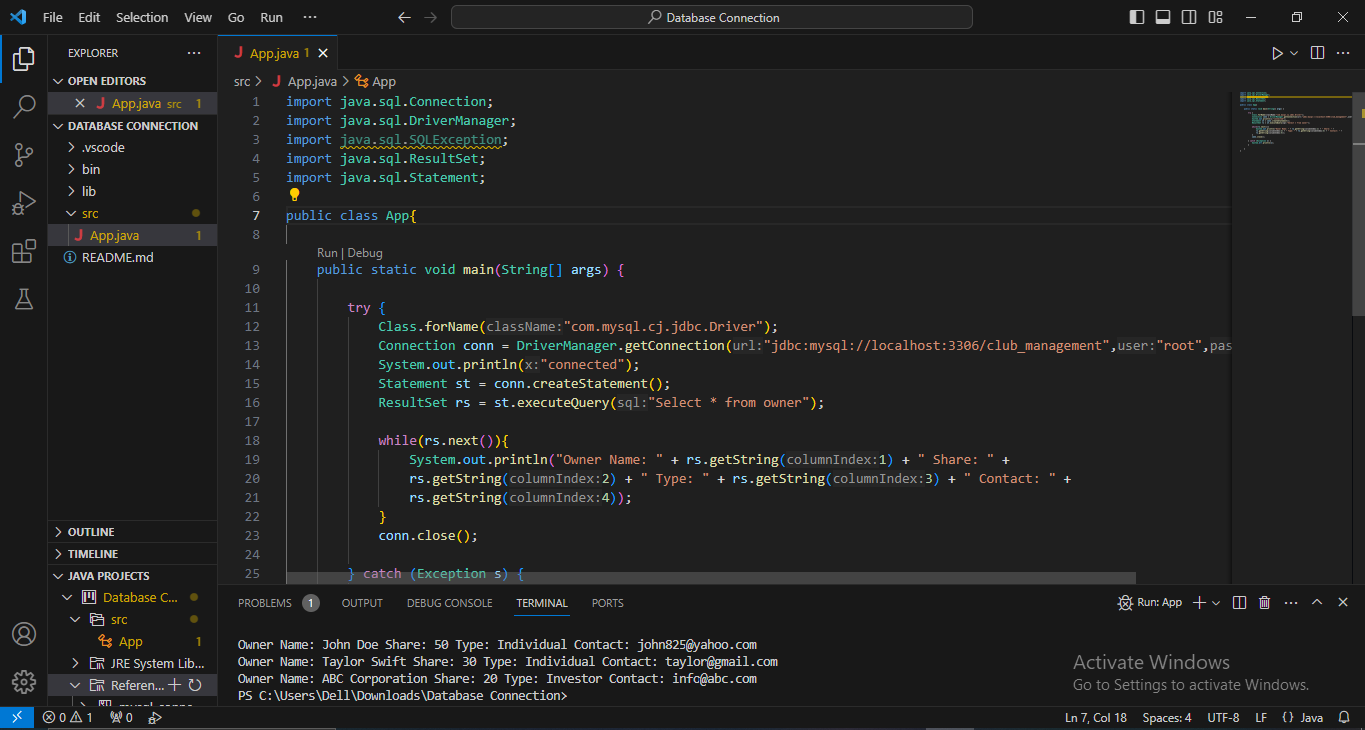
Description automatically generated

Image 9.1.3: Result of connected SQL with java

1. **Close resources**

After executing the operations, ensured to close the ResultSet, Statement, and Connection objects using the **.close()** method to release resources properly.

* 1. **MD. JUBAER AMIN JISHAN (22-49265-3)**
  2. Tools:
     1. Connector: Download mysql java connector JAR file from mysql website for creating interactions of java programs with the database.
     2. XAMPP: For creating a local server & database testing purpose need to install XAMPP software.
  3. Environmental Setup: Install xampp & launch mysql and apache and go to mysql admin panel for creating the database.
  4. Connecting Database: Write the java code in any IDE (I used Visual Studio Code). Attach the connector JAR file a reference library. Then launch the programme it will connect with the database & show the data in the terminal.
  5. Close: FInally close the connection



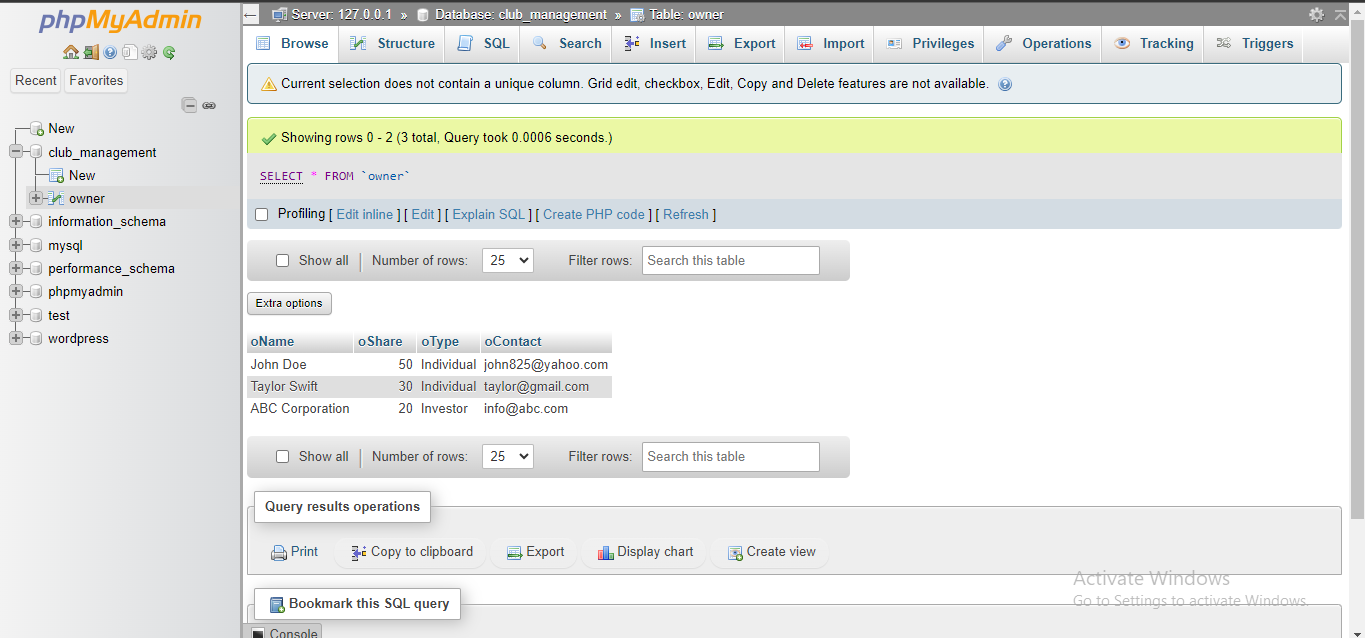


Image 9.2.1: Java connection

**Conclusion**

In essence, the Club Management Database System marks an important role for managing a club information. Its unparalleled adaptability is finely tuned to optimize the data structuring according to the club requirement. As a result, DBMS becomes a catalyst for unburdening leaders from administrative complexities, allowing them to channel their energies towards strategic initiatives that propel the club forward.

Furthermore, DBMS operates within a secure environment, safeguarding the confidentiality of critical data. The provision of real-time access and insightful reporting amplifies the decision-making capabilities of club leaders, empowering them to navigate challenges with clarity and precision.

In the broader context, DBML transcends its role as a mere operational tool, it emerges as a transformative force that not only streamlines day-to-day club activities but also cultivates an environment conducive to growth. As clubs navigate the ever-evolving landscape, DBMS positions them strategically for sustained success, providing a robust foundation for enhanced organization, communication, and overall prosperity.