SOFTWARE REQUIREMENTS SPECIFICATION

for

Game Zone

Prepared by,

S5_T1

Amit Thakkar (201901038)
Jigar Makwana (201901428)
Parth Prajapati (201901429)
Arpit Gohil (201901471)

Mentor : Shreyas

DBMS (2021)

DA-IICT

Contents

Section 1

Final Version of SRS

1 Introduction

1.1 Purpose

This Game zone management system's SRS main objective is to provide a base for the foundation of the project. It gives us a view of "how the system works?" And "what are the users expectations?". It gives the detailed description of functional and non-functional requirements proposed by the customer.

This Game Zone Management System is designed with the hope of making all the day to day work of Game Zone automated such that one can easily manage all tasks of the game zone. The purpose of the Game Zone management system is to give the best environment for Game Developers, Organizers and online gamers as well as offline by maximizing the profit, minimizing the workload and getting improvement in performance of management.

1.2 Document Convention

The document is prepared using Overleaf-Latex and has used the font type 'Time New Roman'. The fixed font size that has been used to type this document is 10pt with 1.5 line spacing. It has used the bold property to set the headings of the document. All pages except the cover page are numbered, the numbers appear on the lower right hand corner of the page. Every image and data table are numbered and referred to in the main text.

1.3 Intended Audience and Reading Suggestions

Engineers who can review project capabilities and easily understand where their efforts need to be identified to improve or add other features to it (create and write an application - set future development guidelines). Project evaluators can use this document as their basis testing strategy as some bugs are easy to find using requirements documents. In this way the testing becomes more well organized.

The document is intended for all the players, game developers Designers, Maintainers and for any person who wants to develop their game zone.

The audience going to play games online is assumed to have basic knowledge of accessing a website, surfing and knowing about its controls. Please prefer a rule book before playing a game, also read carefully the terms and conditions and take care of money transactions. This involves an element of financial risk and may be addictive. Please play responsibly at your own risk.

1.4 Product Scope

Today, Games are one of the most used modes for Entertainment same like TV, movies, Music. The culture of gaming never existed in India. It is now that we see that a potential market for games has emerged. This is because of the new generation which is not only acquainted but also equipped with digital media. Once the gaming Industry develops in India, it will grow to be one of the best markets in the world. It is also said that the gaming industry has generated great opportunities for game developers. It is growing in various platforms like mobile, PC, Game Zone etc. Our application is also based on the management of Game Cafe and Online gaming platforms.

This Game Zone Management System is designed with the hope of making all the day to day work of Game Zone automated such that one can easily manage all tasks of the game zone. This application helps the owner of the game zone to store details of all its games like name, type, Information, price of game, game duration, modes etc, the details of players like ID, game records, payment records, achievements etc and owner can also track all these details in real time with game history. It can also have multiplayer features, so that you can also enjoy various online multiplayer games in a team.

By adding some more features one can develop his own automated offline Game Zone in which Most of the work will be done by the application, so it reduces the workload and also satisfies the players and owner by its excellent performance.

One can also use our application to develop a Game store, which only takes the records of games and customers. We can also modify applications to focus on Single Game like, It also can be used in future to develop online gaming platforms like Battleground Mobile India(BGMI), Counter Strike, Dream11 using Artificial Intelligence, Machine learning, web development with application.

This application also can be upgraded in future by adding AI, ML to use in the International or National gaming tournaments like Indian Premier League (IPL), FIFA, Olympic, Kaun Banega crorepati (KBC) etc. One can also modify the features of an application to use it to easily manage small or big gaming tournaments in college, school or society in less time and less cost.

It can also help to decide the winner of a game by storing real time information of game history. It also takes care of eligibility of the player, timing duration, payment details etc. We can also upgrade this application to reduce player's traffic in the game to improve system performance. In online gaming platforms, it can also provide data security, more storage capacity, easy development, data abstraction like features.

1.5 Description

The project will be an SQL database which will be created in a modern database management system. This will allow it to use the most modern features in DBMS. This will allow us to access the database through the provided GUI, pgadmin for postgres, or the CLI (command line interface), psql for postgres. So, the data can be conveniently modified or inserted into using these interfaces.

PostgreSQL, also known as Postgres, is a free and open-source relational database management system emphasizing extensibility and SQL compliance. PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance. PostgreSQL includes multiple features that are designed to help the developers in developing the applications, manage our data in the datasets, and managers can keep the data integrity, and create the Risk-tolerant environments.

The essential features of PostgreSQL are as follows:

- Compatible with Data Integrity :- It supports data integrity which includes the Primary Keys, UNIQUE, NOT NULL, Foreign Keys, Explicit Locks, Advisory Locks, Exclusion Constraints.
- Support multiple features of SQL :- PostgreSQL supports various features of SQL which include the MVCC (Multi-Version Concurrency Control), It supports multiple Indexing such as Multicolumn, Partial, B-tree, and expressions, Complex SQL queries, It supports transactions, Nested Transactions through Savepoints.
- Compatible with multiple data types :- Structured, Primitives, Geometry, Document (XML, JSON/JSONB)
- Highly extensible :- PostgreSQL is highly extensible because It supports procedural Languages such as Perl, PL/PGSQL, and Python, etc. and also has Stored procedures and functions.
- Secure :- It is safe because it provides a robust access control system and it supports Column and row-level security.
- Highly Reliable :- It is highly reliable and also provides disaster recovery as Active standbys, PITR (Point in time recovery), It supports WAL (Write-ahead Logging).

So using the PgAdmin which is one of the most famous and open-source management and development platforms for PostgreSQL , developers can develop and design many applications easily. Now, let's talk about the features of the project.

The project will have the following features for the GameZone management system.

- Person :-

  In our Gamezone database application, we found there are mainly three types of users, whose information we need to store in the database application: Player, Manager and Worker. All three Entity have some

common properties like name, age, contact no., Email address etc. For the purpose of authentication we will ask users to provide username and password. We will store all this information in the database. In the ER diagram we have used specialization to represent the User class and its three sub-classes. Below we have given detailed information about all three types of users.

- Player :-

  We are creating the game zone database management system to provide the best environment for players to play games. So, players are the first requirement for any game zone, without players there is no meaning of Game zone. Players are the main source of earning money for managers, employees and owners of Game Zone. To experience the players, the features of this application like multiplayer games, Payment records to provide more consistency of transactions, rewards etc we need to store the personal details of players.

  The Database will store the information of players like player id, player name, player age, height, weight, mobile number, email id, team id. Here we use player id as primary key in "player" relation. In the game zone, there are games which have some criteria like age, height and weight so the database stores the age, height, and weight of the player. In future if there are any updates or discounts or events in the game zone, they can contact their customer via their Email Id and mobile number. For the multiplayer games database stores the Team Id which is also unique for each team, so two or more Teams can't take the same Team Id. Team ID will be referred from Team entity.

  The application will take care that the personal details of players do not leak by providing some restrictions like players can see their own profiles and also modify some allowed details. Players can see other player's details like name, score, rank but Players are not allowed to see other player's personal details like password, address, email id, contact number etc. Manager can also see the details of the player apart from the password field. Here the password can be reset by only the player.

- Manager :-

  Every organization or system has some management for their growth. Management is needed in order to coordinate the activities of a business and make sure all employees are working together toward the accomplishment of the organization's goals. For that we need managers to manage these things. In our game zone , to manage the whole system and to monitor employees work the system must need a manager.

  The manager database will have the information like ID, name, mobile no., email address, details about profit and loss of the game zone etc. ID is used to uniquely identify the manager. Profits and Losses have information about a game zone's sales and expenses over a specified period of time. Other information will be taken to contact the manager like using email ID, mobile no., address etc.

  Manager has to manage everything and for that he/she needs the whole control of the game zone. Means he/she must have the privilege of modifying the data, reading the data etc. Manager can read all information related to the game zone like employees data, players data, games data etc. Managers can modify the employees salary, delete any record of game history, remove the account of any player in case of cheating or damaging the property of the game zone and there are many other information managers can modify.

- Worker :-

  Gamezone can't be managed by only the manager. Manager needs some workers who can work under his guidance. Workers will have work like registration of players, supervising, managing payment details etc. So to manage and everything will work smoothly under the guidance of the manager the employees are needed.

  Worker is also a specialized subclass of Person class which will have information like ID, name, age, contact no., email address, role, salary etc. ID is used to uniquely identify each worker. Salary represents the amount paid by the owner per month to the worker. Role has the information of the role of each worker. Other

information is used to contact the employees like using mobile no, email ID etc.

Workers can read and update his/her personal information except its salary (it can only be modified by the manager). They don't have any privilege to modify or read the details of the manager. They can read and insert the players' details. Some of the Workers who are granted by the manager can modify the details of the game zone like price of game, rewards, etc.

- Game :-

  Games are also the most important entity in our project. In a Gamezone there are various types of games available. Players will see the details of the game and after that they make a decision about which game they should play. So, to attract the players and manage all the games we need to store details of the games in our database.

  The database application will store the details of games like name of game, type, mode, price of game, maximum amount of prize that the player can win by playing games etc. To determine all game details uniquely, the database application will give a unique Game ID to all Games. The name, type and mode will provide an overview of the game to players. Some games have age restrictions like this game is only available for players who will satisfy the age criteria decided by the owner or manager. The default value of required age will be 3 years. Game Zone can have both types of games, online games and offline games. Mode will specify whether the game is online or offline. The Game Zone can have different types of solo,duo or multiplayers games like racing games, adventure games, action games etc. Type of game will provide this information to players. In gamezone, generally all the games are premium, you need to pay money if you want to play a game and the player should know the price of the game. Players can win some rewards according to the rank in the leaderboard, the application will store the maximum amount of prize money that the player can win by playing the game. This will also encourage players to play games. Table_no attribute will show the game location if the game is online then it will show room id and if it's offline then it will show the location address.

  The details of the Game will be inserted or modified or deleted by the manager or employee (granted by the manager) . Players can retrieve the details of any game and enjoy the various types of game and also can win some exciting rewards. Players are not allowed to modify or delete any game details.

- Account :-

  When Player visits GameZone, he will be asked to open his account in the GameZone system. For that he has to choose a membership package and pay some money and upload some required documents and details as decided by the owner. After this process, one debit card will be provided to the player by gamezone. Using this debit card, players can make payments and get prize money. So all details related to money will be stored in the account class.

  Account relation contains the field, player_id, account_id, debit_number, current_balance, date_of_joining, username and password. Field player_id is referenced from the "player" relation, account_id is auto generated which is unique and assigned by application to each player, debit_number will be provided by game zone, current_balance shows the amount that in the account which will be updated when any transaction happen between player and game zone, date_of_joining is the date that first time player visited game zone, username is unique for all players which is selected by player and using this player can login to the website.

  Players are allowed to see their Account details, but they aren't able to see other player details. Players are not allowed to modify or delete any records. Manager can also see the account details of all players but can't modify it. Manager can delete the account of any player if the player is found to violate any term and condition, or his validity of the package is finished and the player has not renewed membership yet.

- Transaction :-

  As we have discussed above, we will give the debit card to all players when they open their account

in Game Zone and we are storing the current balance of the player in the Account entity. Players will pay the price for playing the game and also may win or lose some rewards in game, and it will affect the current balance of the player. So, we need to store the all-transaction records in the database application as a proof of transaction . For that we make the class of Transaction, which has the information of transactions like Player ID, Game ID, Amount paid/win/loss, Date and Time of payment, method of transaction etc. To uniquely determine all records, we need to add one more attribute Payment No. for each player. The Combination of Player ID and Payment no. will become the primary key of Transaction. We will take the reference of Player ID from Player Entity and Game ID from Game Entity. The information of the amount transferred helps the application to update the balance of the player efficiently.

If we talk about multiplayer games, then all the members of the team need to pay for playing the game. And the rewards and losses will be equally divided to all members of the team. So, the case of multiplayer is also handled as an individual player's payment.

Players are allowed to retrieve transactions from their own account. But they are not allowed to modify or delete any records. Players are not allowed to see other player's transaction details. Manager and employees (Granted by Manager) are allowed to retrieve the information of transactions. But they can't modify or delete it. The updating in the Account will be done by application as any record will be inserted in the transaction.

- Team :-

  In GameZone, there are some multiplayer games and to play multiplayer games, players are required to make a team and register a team in gamezone.

  The database application will store the information about the team like team name, game id, team rank and the number of games won. All the members of the team should be members of the game zone. To uniquely determine all teams, team_id will be provided by the application, team_name can be decided by the team members, team rank shows their position on the leaderboard of the game, and no. of game wins shows how many game teams won.

  Here players are allowed to view their own team's profile as well as the team member's profile. Some fields like rank, no. of games won will be automatically updated in the database.

- Game history :-

  Game History is one of the most important parts of our Game zone management system. The Game History database stores all the past information of players till now. The information which it stores is all the games details the individual player has played, game performance details like scores, how many levels has completed ?, achievements, no. of won and loss games, how much reward the player has got in different games ?, date and time of when the player has played that game ? etc.

  Now why do we need all this information ? This information helps the player to see their past gaming history about individual games. It has the game performance, scores and rank which helps the player to see their performance so that he/she can improve that. The owner/manager of the game zone can identify which player has the most amazing performance of all time? Which games are played most till now? Which player gets the highest rewards including all games he/she has played till now? etc. From all this information, owners and game developers get to know which games are underrated or have the least popularity among players so they can improve or can remove those games. So as a business perspective for the owner, game history is a very important entity of the game zone.

  For the Game history database, we need, Player ID, Game ID, Player's Score, Rank, the number of won and lost games, win rate, rewards, date and time etc. Player ID and Game ID to get the details of player and the game details he/she has played and For the game statistics and performance, player's score , rank, level all these details are needed. Date and time fields are needed to store when the player has played all those games.

This information will be taken from Player, Game and Leaderboard entities.

Every player can only see their own Game history details. He/she has no privilege to modify the details. Workers and managers can see all players' gaming history details but workers can't make any change but the owner has the privilege to remove or delete the record in case of banning the account or deleting the account.

- Lobby :-

The database application will store the details of players who are currently playing in Game Zone. Here, the application should have the information about which player is playing now? Which game is he playing? At which location is he playing? What is the current status of a player means the player is playing or has finished? This is all the information we need to maintain the players who are currently waiting to play a game.

We are going to store the information like Player ID, Game ID, Table no., number of players playing game (corresponding to game ID), date and time, and current status of player in Lobby. We need Player ID and Game ID to uniquely determine any tuple; we will take this information from the Player and Game entity. Table no., date and time will give extra information to manage the list. In current status there are two possible status: Playing and waiting.

Here, in Game details we have given the capacity of the game which denotes the maximum number of players that can play at a time. When a player will login into the game, we will compare the number of currently playing players with the capacity of the corresponding game, and if capacity has not been reached, we will add him into the currently playing players list and set its status playing, otherwise we will add the player to the waiting list. When a player logs out from the game, we will remove him from the lobby . After login or logout in game, we will update the number of currently playing players in the corresponding game. When any player logs out from the game, we will check in the lobby that any player is waiting for a corresponding game, if yes then we will add into this list and set his status playing and update the number of currently playing players. For multiplayer games, teams will be handled individually as above.

Any insertion, modification and deletion will be done by the application automatically according to the information stored in this list and game history. Players are not allowed to view or modify this list. Manager can give a role to any worker to manage this list. Manager can retrieve the information about currently playing players. But he can't modify it.

2 Fact-Finding Phase

2.1 Background Readings

2.1.1 Description of Readings

Play-Fab PlayFab is a Backend as a Service platform focused on games that is owned by Microsoft. It is one of the most amazing backend solutions which can help developers to create, manage, launch and grow effective cloud-connected gaming applications.

- It is a service platform which is developed for live games with player accounts, virtual goods and real-money ecommerce, leaderboards and in-game chatting, analytics and more.
- It has a good integration with Game developing software like Unity, Cocos etc.
- PlayFab is a user-friendly service that has good storage implementation for user data (Player, Owner, Employee), inventory and game characters. It stores the cloud data, leaderboards, has a ability to connect various services and has a good system for creating store and domestic currencies.
- PlayFab also has a system of triggers and events to automate operations on the backend part. PlayFab's backend services reduce the barriers to launch for game developers, offering both large and small studios cost-effective development solutions that scale with their games and help them engage, retain and monetize players.
- PlayFab enables developers to use the intelligent cloud to build and operate games, analyze gaming data and improve overall gaming experiences.
- Using PlayFab
  - you can remove the challenges of building, managing, and running servers at scale with a complete back-end solution.
  - you get instantly scale dedicated multiplayer servers that deliver low latency and high reliability for real-time game play.
  - You can get multiple forms of built-in authentication to track players across devices.
  - You can quickly create leaderboards to more deeply engage players,
  - You can accelerate growth with economy services that let you create and track virtual currencies, manage stores of items, and process payments.
- If we talk about the features provided by PlayFab,
  - First it provides the multiplayer services which includes Player authentication, Player data management (game state and information sharing across players), Matchmaking (Finding the opponent).
  - It also provides dedicated multiplayer servers which deliver low latency and real time game play for any platform.
  - It provides security by encryption,24/7 monitoring and protection from Distributed Denial of Service (DDoS) attacks.
  - It also provides cost management according to demand, Text-voice chat during game, Features required for tournaments and leaderboards, Scheduling - to reset leaderboards on a schedule so that players also can see past results, Reward management based on Decided reward and Player's rank.
  - It also has a game manager portal through which you can build, configure and operate your game.
  - It gives us daily reports about Owner's profit/loss and the player's profit/loss.
  - Customization
  - Powerful Achievement management System
  - Efficient communication between players
  - Real-time segmentation

- Content(Games) Management
- It also provides monetization, target player segments with personalized store offers and support payments with Xbox, Steam, Google, PayPal and more, Mint promotional coupons and virtual currencies with support for setting initial balances and optional auto-recharge. and also take care of Fraud prevention on payments.
- Testing of System- It has a feature that tests the system on a random group of players and analyzes the performance of the system.
- It has a feature of real time analytics and data management with good efficiency and consistency.

So, these are the features of PlayFab. Casino Management System We all are familiar with Casino. We have ever visited a casino or seen in any movie or web-series. The functionality and purpose of Casino and Gamezone (Project topic) are very similar. Let's see some requirements of the Casino management System, explained by Steve Karoul , a recognized casino consultant with 35 year of hand-on experience of the best Casino in the United States.

A Casino Management System (CMS) serves as a focal point for the day to day processing and recording of operational transactions throughout the facility. Realistically, It's very difficult to find a one single vendor who offers all of the software that any casino needs to operate and manage their facility. There is a need for a central management system which can interact and collect meaningful information from the many operational systems commonly found in today's modern casino facilities. Here there are some basic applications that should integrate with Casino Management Software like Slot Accounting/Monitoring, Slot Ticketing, Cashless Gaming, eGames Platforms, RFID Table Monitoring, Live Point of Sale Systems, Promotional Kiosks, Surveillance and Security, Responsible Gaming, Online Internet Gaming, Third Party Check Cashing, General Ledger, and Data Warehousing etc.

- Security - The sensitive data like player's personal information and data relevant to determine game outcomes, communication between players and employees must be protected from unauthorized access at all times.
- Critical software integrity- We should ensure that the Gaming management system does not execute compromised critical software, for that system must verify the integrity of its critical software prior. The system interface devices deployed at the gaming site to communicate directly between the gaming devices. and the integrity of critical software must be safeguarded during its execution.
- Authentication- the management system must required to having authentication system, which can authenticate their Players and employee. And also take care of that they can only see the details that are allowed to them.
- Access Control - The database should include authorization, so that the casino management system must limit access to only authorized personnel for various functions. All user accounts on the System must be handled by a single individual. The System should have feature of identification and authorization.
- Records and Reporting - The System must be able to record all activities related to the player details, game details, game history, payment transactions, updating leaderboard etc.
- Slot accounting and Slot monitoring - The Slot accounting system must accurately receive, maintain and store each connected gaming device's accounting meters. The Slot monitoring system must be capable of receiving, recording, and alerting the operator in real time of all events and error condition that may impact game integrity, security etc.
- Cashless Wagering- It must accurately issue, record, maintain, validate and redeem only valid cashless wagering instruments. It must accurately identify cashable or non-cashable transactions, the player's account such as deposit funds and promotional funds and also store details of transactions. The cashless Wagering system which includes wagering accounts, must record, maintain, communicate and process all transactions associated with player's account and balance securely and accurately.
- Testing of system - It required to test the system on a decided time period to maintain performance of the system.

Sports Management System This Sports Management System project used to manage the activities of Sports and Games. It stores many sports and games database in a single software. This software stores records of Football,

Cricket, Hockey, Volleyball, Badminton, Basketball. This project can be used for college and mainly this software stores student details and selection details. Even this stores tournament details and multiple college student details.

The goal of the Sports Management System (SMS) is to ensure that it handles the practices of multiple sports at a time. It also oversees student placement programs at the college and state levels. Users can waste less time as compared to manual paperwork on an automatic device. The device will take care of all the service operations in a fast way.

In the Sports Management System (SMS) , there are 2 users, i.e. the controller and the employee. Admin supplies the employee with the username and password. It also can add or change the employee's username and password. Using this username and password to log in to the device. The first procedure is to register a pupil. Here, the employee enters all the specifics of the players, including the sport they want to compete in. Both of this information will be stored in the folder.

Next is the tournament qualification. Depending on the sport, the date of the tournament and the venue are stored in the database for further clarification. There are five different sports out there. The method also carries out the screening process at the college level and the competition at the state level. At the college level, both players are eligible to participate. Whereas at the state level, the best players will be picked from the college level list, which will also be saved in the database. Here is the modules of sports management system are listed :

- Login Registration : Multiple users can login to this system. There are two types of users, i.e; Admin and Employee. Admin has full authority of the software and Employee has limited authority. Only administrators can add new employee details.
- Student Registration : Students are competitors in the Sports Management System. This module allows user to add student details by entering College name, Student name, Date of birth, Age, Gender and Competing sports and games. Using the given information it will see the games criteria like minimum age, gender preferences, disability of players etc.
- Tournament : This module stores every single detail related to the tournament like tournament schedule details with Tournament sports, scheduled date and Tournament Venue.
- Game : This will show Football, Cricket, Hockey, Volleyball, Badminton, Basketball Players details, Tournament details and Photo gallery. Several players will be highlighted and the upcoming competitions will also be listed according to the same sport. The system also allows a special authority to submit pictures to the system for a specific sport for nice memories and evidence.
- Selection Procedure : The system also carries out the selection procedure at the college level and the competition at the state level. At the college level, both players are eligible to participate. Whereas at the state level, the best players will be picked from the college level list, which will also be saved in the database.
- Logout : This module helps the user to check in to the program. Further operations cannot be carried out until the user exits.

Sports management system is not only rewarding but it also helps the programmer to quickly organize the sports events and lists in short interval of time. It will be able to check anything related to sports at any time. Paper work and manual work is reduced through this system. The system is user friendly and easy to use. Poki Playground Management Poki has the best free online games selection and offers the most fun experience to play alone or with friends. Poki offers instant play to all Poki's games without downloads, login, popups or other distractions. Their games are playable on desktop, tablet and mobile so anyone can enjoy them at school, at home or on the road.

All features on the Poki Site are available without the need for registration by Users. No personal data (such as email addresses and phone numbers) are requested from users. Poki may collect certain electronic data (such as the User's IP address, cookies or device ID).

Poki has the best features I think because you can play free games without downloading them. Also, you don't have to register yourself just play when you want to play and It has multiplayer games. Where two players or more

players can play a game.

There is also a different subsection of games :

- Action games
- Adventure games
- Racing games
- Funny games
- Puzzle games

They have 30 people who manage these platform games as they are not accessing player data but they use an IP address to market their platforms. They also have a privacy policy that children who are more than 13 years only can play on the Poki platform.

Battleground Mobile India Battlegrounds Mobile India is a player versus player shooter game in which up to one hundred players fight in a battle royale, a type of large-scale last man standing deathmatch where players fight to remain the last alive. Players can choose to enter the match solo, duo, or with a small team of up to four people. The last person or team alive wins the match. It is designed to get the adrenaline rushing- BGMI is an intense game, which is so fast paced and suspenseful that it takes the player to another level of excitement, it keeps you under suspense till the last enemy is shot down. Hence it is difficult to lose your concentration even for a second. BGMI is a team game, so you can play it with your friends and discuss strategies, plan other things during the game to take the enemy down.

There are so many features in the game. Some of the best features are listed below :

- Amazing graphics and Sound Effects : The players get to experience realistic gameplay effects with either weapons,or vehicles. And, the high-quality audio and 3D sound effects, make the gameplay more immersive and intense.
- Different modes : There are so many modes in game like Classic, Arcade, Arena, VS AI, Evo ground etc. In classic mode there are many maps like Erangel, Miramar, Sanhok, Vikendi, Karakin, Livik etc. In the arcade there are three types: Quick match (same as classic but less player and small map) , Sniper training, and War. In Arena, There is team Deathmatch, Arena Training, Gun game, Assault, Domination.
- Promotes fair play : BGMI has anti-cheat mechanisms which quickly catch any players trying to win from cheating or illegal techniques. This promotes a sense of unity and fair play among the players.
- Team work : The players can play with their friends by connecting through facebook or by making friends in game. It works on the principle of Team Work and Team Building. As a part of the team, you are aware of the team tactics, strategy and enemy movement. It becomes easier to survive and kill if surrounded by the enemies and can help you get ammo, med kits, etc. You can talk to your teammates and you make them aware of the next move.
- Voice and text messaging : BGMI is a multi player game, apart from sending chats to friends, you can even send voice chats to other players.
- Always come up with a new idea : The game developers always modernize when things start to get boring. For example, they have added new events like flora menace, mission ignition, anniversary etc, new weaponry like MG3-machine gun, MK12 shot gun, P90 submachine gun etc, and new maps like livik, karakin, vikendi , hanger with diverse weather and day/night options, new characters like Andy – the magician, carlo – bounty hunter etc,
- Age restriction : To be able to play Battlegrounds Mobile India, someone who is under 18 by law, will need to get express consent from parents or guardians. According to Krafton, Battlegrounds Mobile will need you to provide one of your parents' or guardians' mobile number "to confirm that you are legally eligible to play the game." The player whose age is less than 18 can play at most 3 hrs in a day.
- Realistic : BGMI has feature to play in FPP (First person perspective) and TPP (Third person perspective). It supports the UHD – Extreme graphics which makes the game so realistic.

There is an in-game currency called UC through which players can buy royal pass, purchase outfits, gun skins, crate opening, get new characters, design the guns etc.It will be harmful for people who are so addicted to this game. But the overall gaming experience is so immersive and smooth and the reason for its management system behind the game.

The game takes the information from users like nickname, email id, phone number, gender, age, country of residency, etc. to manage the whole system behind the game. The data used for making characters in game, register for E-sports events, for surveys etc. It takes some data automatically like device information (IP address, device id, device model, network type, OS version, Screen resolution, time zone setting etc.) This information is used to enhance the gaming experience. From this information the game will know that its device is good enough to play the game. It takes data of our gameplay to check whether any user is breaking any rules and regulations while playing games. Generally it is used to improve the NPCs (non – player character) game play such that it looks like real players. It takes crash reports and system activity details to solve any error or problem while playing a game. Some information is also taken from the third parties app, like when a user registers him/her self then he / she gives their gmail ID, facebook ID, twitter ID . It also takes the information about the users' purchases and transaction history.

The game stores the information about the player's statistics like how many matches the player has won or losses, what is the highest damage he/she has ever made, the number of headshots, finish/ death ratio(F/D ration) which defines the total number of players he/she has finished divided by the total match he/she has played(doesn't matter the won matches or loss matches). It also takes the information of achievements of the players and also stores the controls setting, sensitivity to the cloud.

They use the information to solve technical problems and improve the quality of service, to identify the individual players to provide game service. Technical protection against usage of unauthorized programs, to prevent inappropriate game play that could harm other players, to search player statistics and provision of leaderboard, to provide customer service by collecting and responding to inquiries etc.

Shott Game Zone Shott is a gamezone which is located in two cities in India: Ahmedabad and Surat.In gamezone, there are games for everybody like for children, it has games that can help them enhance their gross and fine motor skills , for youngsters and families it has games that can be played with friends, cousins, siblings, parents, uncles or aunts in a group. It also has a Restaurant and Café. Everybody can indulge in the gaming atmosphere at Shott game zone and can spend a great time together to create more memories.

When we register for an Account, they ask for our contact information, including items such as name, company name, address, email address, and telephone number. They Store our data in their database. They use our information to improve and expand their website, develop new services, features and functionality based on user requirements, communicate with customers to provide updates and other information regarding their offers and all.

They follow a standard procedure of utilizing log files. These files log visitors when they visit websites. The information accumulated by log files include internet protocol (IP) addresses, browser type, Internet Accommodation Provider (ISP), date and time stamp, and referring/exit pages. These are not linked to any information that is personally identifiable. They utilize this information for analyzing trends, administering the site, tracking users' kineticism on the website, and amassing demographic information.

In Shott gamezone there are games like Go-Karting, Bowling, 360 Cricket, Laser wars, Kids City etc. There we can experience indoor and outdoor adventure fun games, virtual reality games and arcade games. They also give some cashback offers and combo of two games.They also give some prizes and rewards for completing some tasks of games.

2.1.2 References

- https://docs.microsoft.com/en-us/gaming/playfab/what-is-playfab
- https://www.euroasiacasino.com/understanding-the-basics-about-casino-management-systems-cms/
- https://www.agco.ca/lottery-and-gaming/igaming-operators-and-gaming-related-suppliers
- https://www.freestudentprojects.com/studentprojectreport/projectreport/sports-management-system/
- https://poki.com/
- https://www.battlegroundsmobileindia.com/privacy
- https://www.quora.com/Why-is-PlayerUnknown%E2%80%99s-Battlegrounds-so-popular
- https://www.shott.in/ahmedabadsbr/

2.2 List of Requiremnets Gathered from Backgroud Readings

- To help the developers to create, manage, launch and grow effective cloud-connected gaming applications.
- There are real money games so they strictly need to maintain data so players don't lose their money.
- To easily manage Player, Owner, Employee data and also the game leaderboards.
- To help the developers to improve overall gaming experience.
- GMS can help to instantly scale dedicated multiplayer servers that deliver low latency and high reliability for real-time gameplay.
- With the use of a gamezone management system you can quickly create leaderboards.
- GMS can help to accelerate growth with economy services that let Developers create and track virtual currencies, manage stores of items, and process payments.
- Design a system such that anyone from anywhere can play games and also if a group of friends are not at the same place still all can play together just they need an internet.
- Reward management based on Decided reward and Player's rank.
- To easily manage Multiple games at the same time.
- To manage when people booking their slots for games like slots are not free then it will show that slots are full or avoid two people to book the same slot.
- If a player wants to update his/her information then it will not affect anyone else's data.
- Different views like one view for the player and another view for the admin who is managing the game.
- To manage tournaments who won, their past matches, points etc.
- GMS can help to easily check eligibility criteria.

2.3 Interviews

2.3.1 Interview 1

DBMS Project: Role Play Interview Plan

System : Game Zone Management System

Project Reference : DBMS/S5/T1

Interviewee : Smit Savan (Role Play)
Designation: Game Developer

Interviewer :

1. Amit Thakkar (Actual)
   Designation: Student at DAIICT
2. Jigar Makwana (Actual)
   Designation: Student at DAIICT
3. Parth Prajapati (Actual)
   Designation: Student at DAIICT
4. Arpit Gohil (Actual)
   Designation: Student at DAIICT

Date: 7 October 2021 **Time:** 15:00

Duration: 45min **Place:** Google Meet

Purpose of the interview

Preliminary meeting to identify problems and requirements regarding Game Zone Management System

Agenda

- Problems with existing Game Zone System
- Strategy to Attract Players
- Gaming Payment Solutions
- Managing both Gaming modes online and offline simultaneously
- More about Multiplayer Mode
- Problems with security and any other concerns
- Follow-up actions

Documents to be brought to the interview

- Rough plan of Game Management System
- Documents relating to Gaming payment solutions

2.3.2 Interview 1 Summary

DBMS Project: Role Play Interview Plan

System : Game Zone Management System

Project Reference : DBMS/S5/T1

Interviewee : Smit Savan (Role Play)
Designation: Game Developer

Interviewer :

1. Amit Thakkar (Actual)
   Designation: Student at DAIICT
2. Jigar Makwana (Actual)
   Designation: Student at DAIICT
3. Parth Prajapati (Actual)
   Designation: Student at DAIICT
4. Arpit Gohil (Actual)
   Designation: Student at DAIICT

Date: 7 October 2021 **Time:** 15:00

Duration: 45min **Place:** Google Meet

Summary of Interview

- Lack of awareness in people about Game Zone
- Poor implementation of payment security and privacy policy
- Lack of multiplayer support in games because of it is hard to implement
- Need an improvement in the performance of existing system
- Handling the power failures and system crashes to recover the data
- Poor designing of websites and lack of optimization
- Need of synchronized and secure system

2.3.3 Interview 2

DBMS Project: Role Play Interview Plan

System : Game Zone Management System

Project Reference : DBMS/S5/T1

Interviewee : Charitya Dedhi (Role Play)
Designation: Owner of Casino

Interviewer :

1. Amit Thakkar (Actual)
   Designation: Student at DAIICT
2. Jigar Makwana (Actual)
   Designation: Student at DAIICT
3. Parth Prajapati (Actual)
   Designation: Student at DAIICT
4. Arpit Gohil (Actual)
   Designation: Student at DAIICT

Date: 7 October 2021 **Time:** 16:00

Duration: 45min **Place:** Google Meet

Purpose of the interview

Preliminary meeting to identify problems and requirements regarding Business Constraints

Agenda

- Problems with existing Game Zone System
- Need of Centralized Management System
- Initial ideas for Business
- Software and Hardware requirements
- Business Constraints
- Follow-up actions

Documents to be brought to the interview

- Rough Building plan of Game Zone
- Current System Manual

2.3.4 Interview 2 Summary

DBMS Project: Role Play Interview Plan

System : Game Zone Management System

Project Reference : DBMS/S5/T1

Interviewee : Charitya Dedhi (Role Play)
Designation: Owner of Casino

Interviewer :

1. Amit Thakkar (Actual)
   Designation: Student at DAIICT
2. Jigar Makwana (Actual)
   Designation: Student at DAIICT
3. Parth Prajapati (Actual)
   Designation: Student at DAIICT
4. Arpit Gohil (Actual)
   Designation: Student at DAIICT

Date: 7 October 2021 **Time:** 16:00

Duration: 45min **Place:** Google Meet

Summary of Interview

- Preliminary meeting to identify problems and requirements regarding the gamezone.
- According to the situation, how we can enhance the profit.
- Financial position of owner.
- How much he will be able to invest for the gamezone.
- How should we marketize our gamezone? What way should we use like youtube ads, TV ads or posters etc.

2.3.5 Interview 3

DBMS Project: Role Play Interview Plan

System : Game Zone Management System

Project Reference : DBMS/S5/T1

Interviewee : Ronak Jethava (Role Play)
Designation: Member of Game Zone

Interviewer :

1. Amit Thakkar (Actual)
   Designation: Student at DAIICT
2. Jigar Makwana (Actual)
   Designation: Student at DAIICT
3. Parth Prajapati (Actual)
   Designation: Student at DAIICT
4. Arpit Gohil (Actual)
   Designation: Student at DAIICT

Date: 7 October 2021 **Time:** 17:00

Duration: 45min **Place:** Google Meet

Purpose of the interview

Preliminary meeting to identify problem regarding existing Game Zone system and Demands of Players

Agenda

- Problems with existing Game Zone System
- Ideas to Improve the performance of existing system
- Follow-up actions

2.3.6 Interview 3 Summary

DBMS Project: Role Play Interview Plan

System : Game Zone Management System

Project Reference : DBMS/S5/T1

Interviewee : Ronak Jethava (Role Play)
Designation: Member of Game Zone

Interviewer :

1. Amit Thakkar (Actual)
   Designation: Student at DAIICT
2. Jigar Makwana (Actual)
   Designation: Student at DAIICT
3. Parth Prajapati (Actual)
   Designation: Student at DAIICT
4. Arpit Gohil (Actual)
   Designation: Student at DAIICT

Date: 7 October 2021 **Time:** 17:00

Duration: 45min **Place:** Google Meet

Summary of Interview

- Preliminary meeting to identify problems regarding the existing Game management system and to know about the player's requirements from the Game management system.
- His experience and how we can enhance their fun through games.
- An overview of the gamezone from a member perspective.

2.3.7 List of Requirements from Interviews

- Solutions of existing problems in Game Zone system
- Use social media platforms for marketing of product.
- Accept all type of payment methods like G-pay, PhonePay, Paytm etc.
- Need to increase the membership price for more profit.
- Offer great customer service.
- Give exciting offers to increase the member
- Try to add/Update new games for more fun
- Remove bugs and glitches from games for better experience.
- Replace the under rated games from the system.

2.4 Questionnaire

2.4.1 Questions

1. Select your Age - group
   (a) < 13
   (b) 13 - 18
   (c) 18 - 25
   (d) > 25
2. Height
   _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
3. Weight
   _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
4. Have you ever visited a Game Zone ?
   (a) Yes
   (b) No
5. Which game mode do you like ?
   (a) Online
   (b) Offline
6. What are your favourite game genres ? (You can select multiple answers)
   (a) Adventure
   (b) Racing
   (c) Puzzle
   (d) Action
   (e) Arcade
7. How frequent do you visit the Game Zone ?
   (a) Once in a week
   (b) Once in a month
   (c) Once in 3 months
   (d) Once in 6 months
   (e) Once in a year
8. How much are you willing to pay for a game ?
   (a) < 100
   (b) < 200
   (c) < 500
   (d) < 1000
   (e) Any price if game is worth it
9. What are your favourite gaming platforms ? (You can select multiple answers)
   (a) Mobile
   (b) PS5

(c) Xbox

(d) PC

10. Which type of game do you like to play ? (You can select multiple answers)

(a) Solo

(b) Multiplayer

11. Why do you play games ?

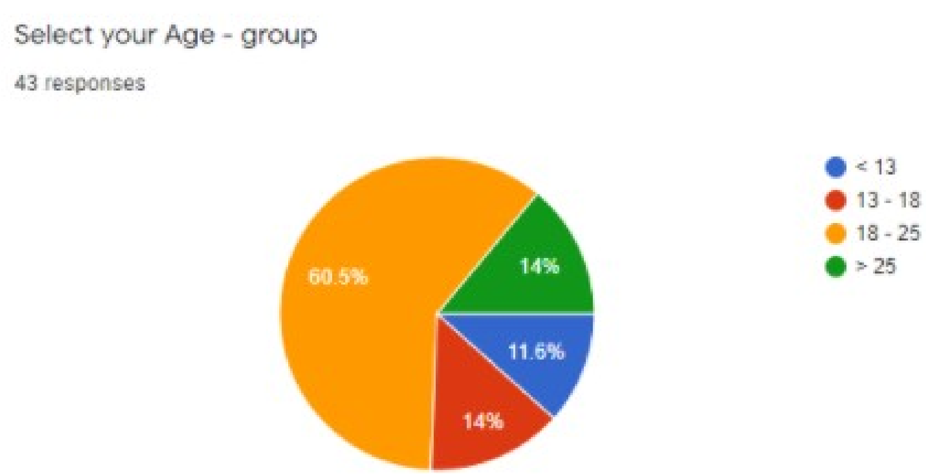
(a) Have Fun

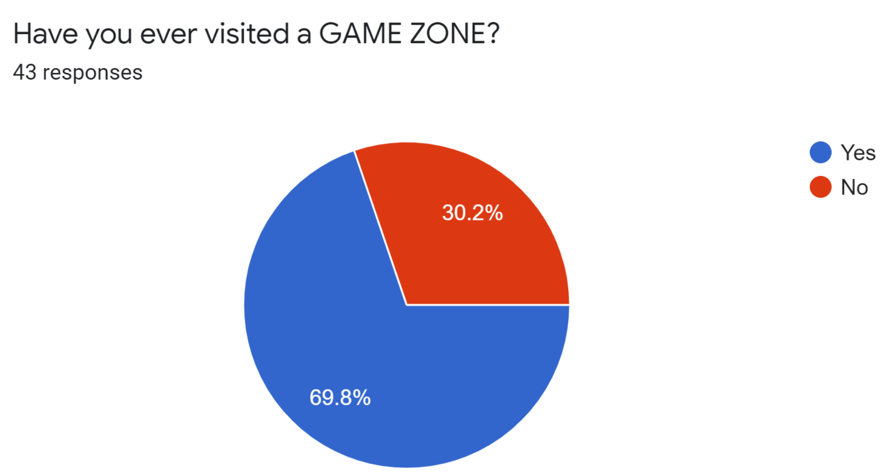
(b) Stress relief

(c) Win Prize

12. Anything you want to say about the Game Zone
_ _
_ _
_ _

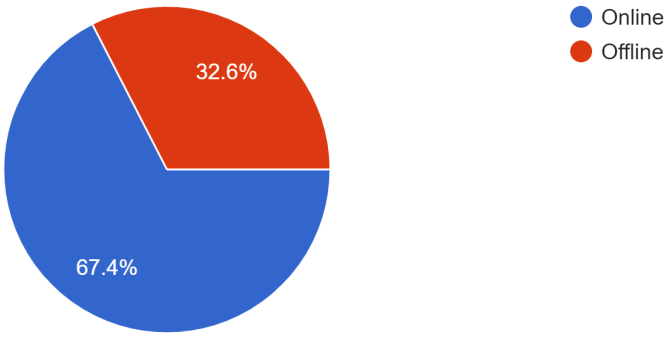
2.4.2 Graphs of Received Responses





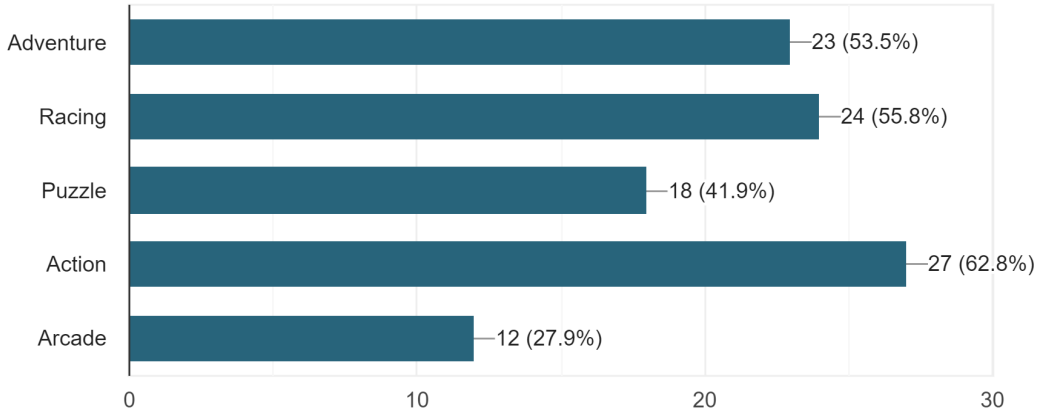
Which game mode do you like?

43 responses



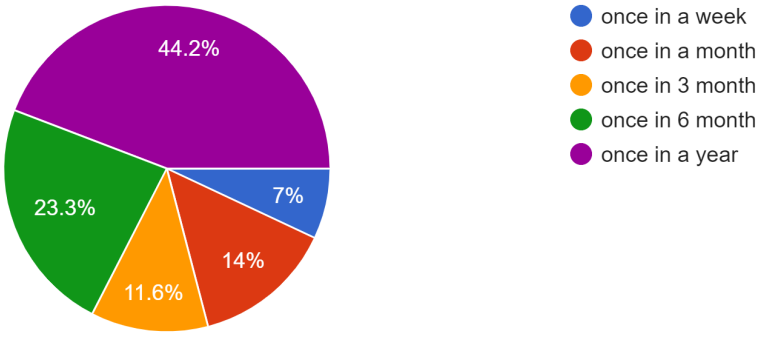
What are your favourite game genres?

43 responses



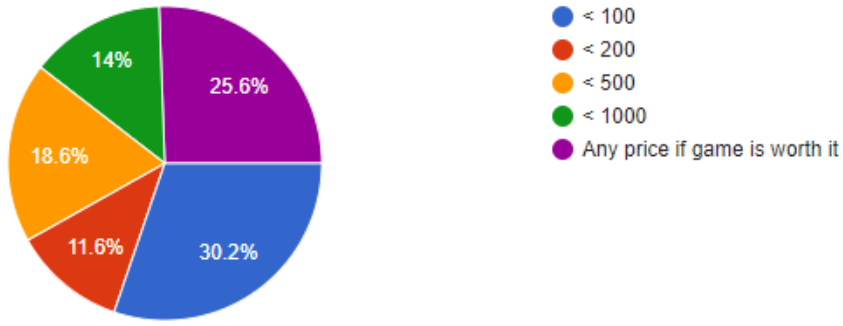
How frequent do you visit the GAMEZONE?

43 responses



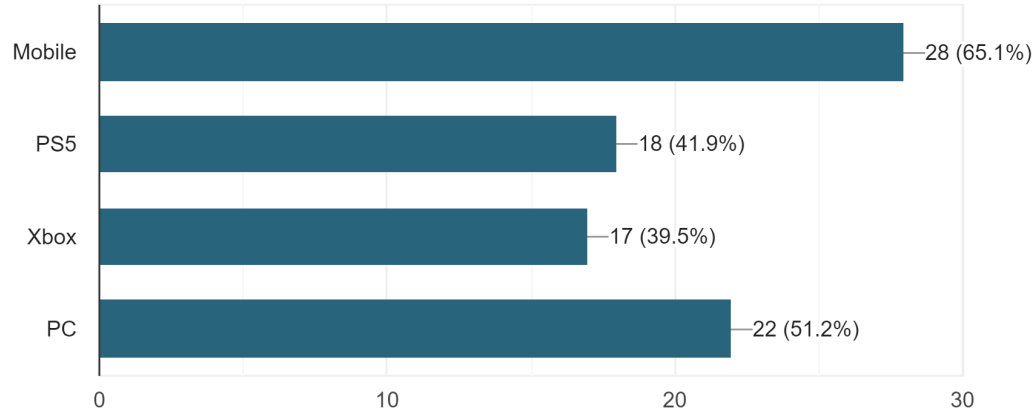
How much are you willing to pay for game?

43 responses



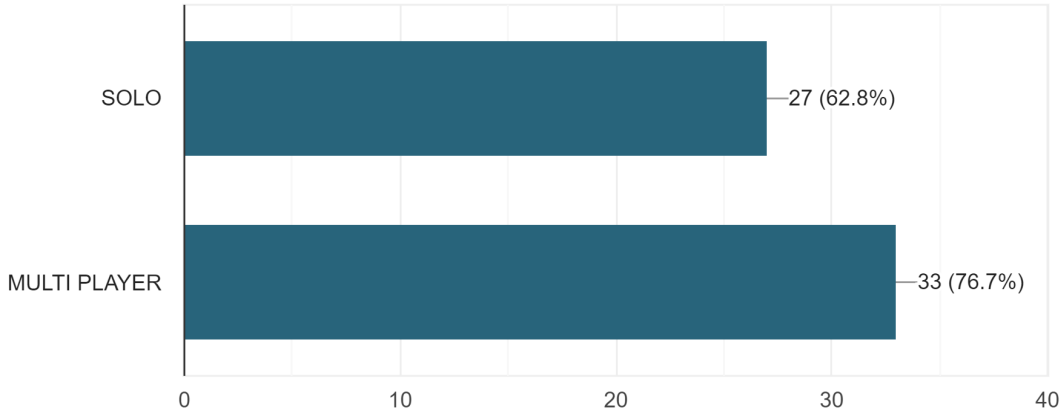
What are your favourite gaming platforms?

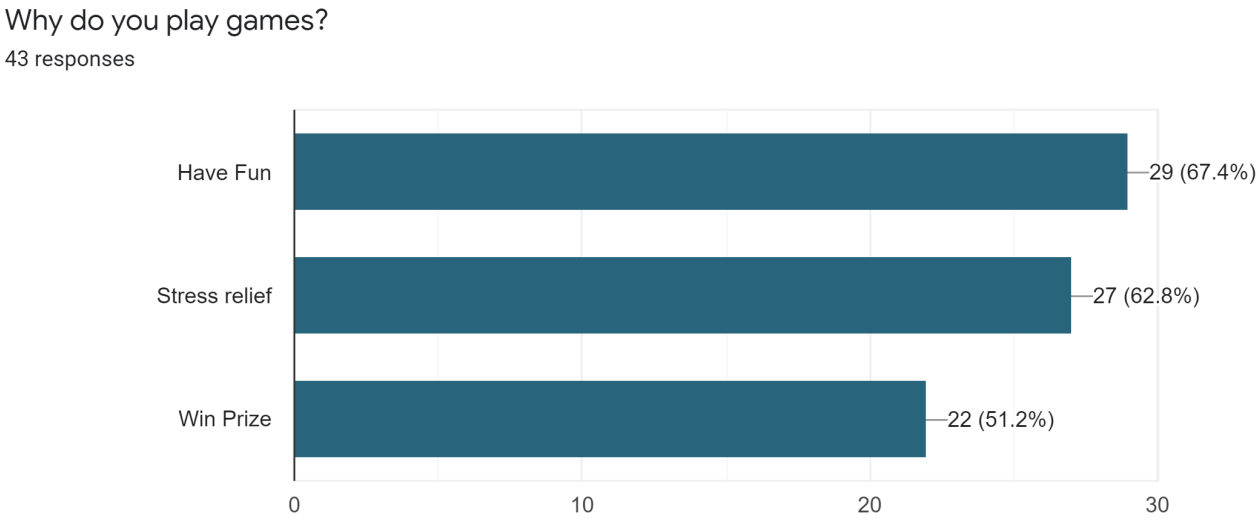
43 responses



Which type of game you like to play?

43 responses





2.4.3 Results

From the Game Zone Management System Survey we have observed following :

- The majority of people whose age is between 18 to 25 are playing games very frequently. The age below 13 and above 25, there are a few people who have interest in playing games.
- Majority of the population has visited the Game Zone. That means most people are aware of the Game zone.
- Most people are playing online games. Because it can play from anywhere, you can connect with other players and communicate with them, unlimited players are playing so there are a lot of competitions and more challenges whereas in offline gaming there are fewer challenges.
- Nowadays the action has become entertainment for everyone. In movies, TV series, sports, games etc. Most people like to play action games. After the action games, people like adventures and racing games.
- Most people visit the Game Zone once a year. Because everyone has gaming platforms like Mobile, PCs. So people avoid going to Game Zone and play games with their mobile and PCs. That is why Mobile and PC have become the most used gaming platforms nowadays.
- Majority of people are playing multiplayer games because these games can be played with friends with voice and text messaging.
- Most people play games for fun and to remove the stress and tensions.

2.5 Observations

DBMS Project: Role Play Interview Plan

System : Game Zone Management System

Project Reference : DBMS/S5/T1

Observed by :

1. Amit Thakkar (Actual)
   Designation: Student at DAIICT
2. Jigar Makwana (Actual)
   Designation: Student at DAIICT
3. Parth Prajapati (Actual)
   Designation: Student at DAIICT
4. Arpit Gohil (Actual)
   Designation: Student at DAIICT

Date: 7 October 2021 **Time:** 15:00

Duration: 45min **Place:** Google Meet

Observations :

- Lack of awareness in people about Game Zone
- Improper planning and data management
- Need an improvement in the performance of existing system
- Need of investors and Developers to enhance the security of system
- Need to meet the user's requirements
- Need to optimize the outdated websites and its designing
- Need to add new features in offline mode of gaming because of least number of offline players.
- Proper managing of waiting users.

3 Fact-Finding Chart

Objective	Technique	Subject	Time Commitment
To understand the working of Game Zone Management system	Background Reading	Articles, Websites, Videos	2 days
To establish business objectives and agree on likely scope of new system	Interview	Business development executive	1 hour
To identify the problems in the current system and to establish the additional requirements	Interview	Game developer	1 hour
To understand the perspective of players about system concerns and requirements	Questionnaire	College Students	2 days
To determine the features of the system	Questionnaire	College Students	2 days
To follow up development of database and website	Observation	Game Developers	1 hour

4 List of Requirements

- High storage capability to store the all data.
- Player Authentication techniques to track the players across devices
- Requirement of Economical services so that owner can create and track virtual currencies, manage stores of items an process payments.
- data security, encrption, 24/7 monitoring and protection from cyber attacks.
- Powerful Content Management System, Achievement management System, Reward Management.
- Secure Payment System, that also provide monetization with personalized offers and support payments on many payment applications. Mint promotional coupons and virtual currencies with support for setting initial balance and optional auto-recharge.
- Good access control and authorization that limit access to only authorized personnel for various functions.
- Slot accounting and Slot monitoring
- Testing of system to know needed improvements in system and ensure about working of system
- Remove bugs and glitches from games for better experience.
- Optimize the game graphics such that low end device can support.
- Give exciting updates on given periods so people can enjoy to play games.
- The games must have Anti cheat mechanism so the game play will be fair for everyone.
- Eligibility criteria should be followed in each games.
- Data recovery management in case of system crashes or power failures.

5 User classes and characteristics

There are mainly 3 type of Users that can use our application:

5.1 Manager

Manager can retrieve, modify and manage the information of their workers, players, games. He decides the salary of workers, price of games, availability for playing games, capacity, duration of playing etc. He can give workers the permission of retrieval, modification and deletion in the database. This application provides many features to help him manage all tasks efficiently.

5.2 Player

Players can open their account in our gamezone, can enjoy various types of solo, duo or multiplayer games and win exciting rewards. Players can see their own profile, their team's profile and team member's profile and also modify some allowed details. Players are allowed to view other player's details on some conditions. Players also can retrieve the details of the game and also can play the game on some conditions. Players are allowed to do cash payments, payment records will be saved in the database and players are allowed to retrieve information of the payment records. Players can view the leaderboard. All games have age, height, weight constraints, So all players who satisfy these constraints can play games. We have distributed the players based on their characteristics.

- Play for Stress Relaxing
- Play for fun
- Play to win money

5.3 Worker

Manager of Gamezone can hire the persons to manage some important tasks. Workers have rights to retrieve information of players, games, game history and also can perform modification on allowed game details. They are employed to manage achievements, real-time game history, Player's traffic management, Game performance etc. Managers can give rights to workers to perform certain tasks, and workers make the work of managers easier.

6 Operating Environment

6.1 Hardware Requirements

Intel i5 processor or Android 5.0 or higher
Hardware equipments for games

6.2 Software Requirements

Windows10 or Android OS
Windows/Unix based server that supports PostgreSQL

6.3 Connectivity Requirements

Wi-Fi or Mobile Internet connectivity

6.4 External Interfaces Requirements

6.4.1 Hardware Interfaces

The hardware interface required for the user to retrieve the data from the server in which the web portal is hosted mainly involves a server and a personal pc connected through a network interface.

6.4.2 Software Interfaces

The complete user data is stored in the database and the information is accessed by the user through a web browser enabled device. It may also use game interfaces to directly take real time game-history details from games.

7 Product Functions

- Basic functionalities

  Players, workers and managers can perform Insertion, data retrieval, Modification and deletion according to privileges given to them. Some functionalities are automatically managed by postgres database system like Transaction managements, Integrity constraints, concurrent access, Data consistency and security, Authorization, Scalability, Data backup and restore, fraud prevention on payments etc.

- Data abstraction

  The application will provide the data abstraction using view through which users can not view the details that are not relevant to them. Like Players can't see other player's payment records or personal details etc.

- Eligibility Constraints

  Every Game has some age/height/weight constraint for players. Players are required to fulfill all eligibility criterias to play the game. These eligibility constraints are automatically handled by our database system.

- Leaderboard

  A leaderboard is the most commonly used game feature in all types of games nowadays. Leader-boards are used for displaying the ranking of the players in competitive games. Such as BGMI, COD, Free Fire etc. for example in BGMI games, The players are participating in tournaments and competing with other players. The result of the tournament is displayed in the leaderboard with their ranks, name and rating points, win rate etc. It also has the leaderboard in the other modes of the game.

  The purpose of a leaderboard is to show players where they rank in a gamified system. Those at the top enjoy the notoriety it brings; as for everyone else, the leaderboard shows them where they stand relative to their peers. Often, the very presence of a leaderboard can elicit the desire to play. The simple goal of rising up the rankings serves as a powerful motivator to continue. People like to keep score. Understanding this and providing easy ways to do it is a great way to foster engagement in gameplay. For some, the mere sight of their rank on the leaderboard is all the reward they seek. When it comes to cost-effective gamifying tools, nothing beats a leaderboard. The trick to leaderboards is designing them in such a way that they encourage players to stay in the game. Leaderboards should always be encouraging, never discouraging.

  Leaderboard database will have the information like Player Id, Game ID, Scores, Ranks, Win rate etc. Player ID and Game ID are used to uniquely identify the player and game respectively. Score is calculated on many bases, for example using survival time, using the number of kills, etc. Win rate shows the consistency of the player in that game.

  All players, workers and Manager have the privilege of reading the leaderboard but not have the privilege of modifying any details of the leaderboard.

- Waiting list Management

  The database application will store the details of players in the waiting list who are waiting to play a specific game. For that we need the information like, which player is waiting? For which Game? That's why we will store Player ID, Game ID and login time (From when the player is waiting to play the corresponding game) in the Waiting list. Combination of Player ID and Game ID will determine all tuples uniquely. We have taken the reference of Player ID from Player Entity and Game ID from Game Entity.

  As we have discussed above, When the player will login into any game, we will check whether the number of currently playing players reached the capacity of the corresponding game or not. If yes then we will add the player's details and the session time into the waiting list. As any player finish the game, we will remove it from the currently playing player's list (lobby). We will find the earliest player that is waiting for that game, if found then we will remove him from the waiting list and add into the currently playing player's list and change the status of him to "Playing". So, to find the earliest player, we need game session time that denotes the time from when the player is waiting for the corresponding game. For the multiplayer games we will handle the waiting list individually same as above.

  All tasks will be done by the database application automatically using the Currently playing player's list. Manager can retrieve the details of the waiting list, but can't modify it. Manager can grant permission to any employee to manage the waiting list. Players are not allowed to view or modify any details of the waiting list.

- Reward management

  Manager has rights to decide rewards for games. The system will check the rank of the player in the corresponding game from the leaderboard and add the reward money into the player's current account balance.

- Multiplayer Mode

  The application has multiplayer games on the system. Players can form a team with their friends and enjoy multiplayer games. The application will store the team details in the "Team" table.

8 Privileges

- Manager

  Manager can retrieve, modify and manage the information of their workers, players, and games. He has a privilege to decide the salary of workers, price of games, eligibility criteria for playing games, Rewards for Game, capacity, duration of playing etc. Manager can grant any privilege to worker and also can revoke the privileges from Players and Workers. He can retrieve the information of all payment records, but can't modify it.

- Worker

  Workers have rights to retrieve, modify the information of players, games, game history, leaderboard etc according to the privileges given to them by Manager.

- Players

  Players can see their own profile, their team's profile and team member's profile and also modify some allowed details. Players are allowed to view other player's details allowed by Manager or Worker. Players also can retrieve the details of the game and also can play the game if they satisfy eligibility criteria. Players are allowed to do cash payments, payment records will be saved in the database and players are allowed to retrieve information of the payment records. Players can view the leaderboard. Players are not allowed to perform modification or deletion in the system.

- Waiting list Management

  This task is done automatically by the database application. But manager can increase the capacity of a particular game. Only Manager has permission to retrieve the information of the waiting list. No one can modify or delete records from the waiting list and queue.

- Updation of Leaderboard

  All Players and Teams are allowed to view the leaderboard. The update on the leaderboard will be done by the application. No one is allowed to perform modification or deletion on it.

- Team

  Players are allowed to make a Team and enjoy various multiplayer games. Team members can retrieve or modify the details of their own team.

9 Assumptions

- It is assumed that the database will work efficiently.
- It is assumed that the database will be updated every time.
- All players gave permission to store their data.
- The Details provided by Players are right.
- All persons have an active data plan and access to the internet.

10 Business Constraints

- The database has detailed information about players, winners, employees and hence must be secured properly.
- The database should be regularly updated and a backup of updated data should be available to the concerned authorities.
- Data should not get corrupted in case of system crash or power failures.
- Creation of digital infrastructure, good internet connection and fast accessible database server that can handle large amounts of traffic at a time.

Section 2

Noun Analysis

11 Noun Analysis

11.1 All Extracted Nouns & Verbs from Problem Description

Nouns	Verbs
Above	Access
Account	Add
Account id	affect
Achievements	allow
Action	Alter
Address	ask
Administrator	Auto generate
Adventure	Automate
Age	Ban
Application	become
attribute	Change
Awareness	Check
balance	choose
Capacity	compare
class	Compete
Company	Connect
Competition	contact
Consistency	Contain
Constraints	Create
Contact number	Decide
criteria	Delete
Current balance	denote
customer	Design
Database	determine
Date	Develop
Date of joining	Differ
Debit Card	discuss
Debit card no.	Display
Desire	divide
detail	Editing
Developer	Entertain
discount	Find
documents	finish
Employee	Follow
Engagement	Foster
Engineers	get
Entity	Give
events	Grant
feature	Handle
field	Help
Foreign key	Identify
Game Interfaces	Improve
Game mode	Insert
Game Name	leave
Game session time	Log out
Game Type	login
Gameplay	Loss
GameZone	Maintain
Geometry	Make
Goal	ManageImprove

Nouns	Verbs
Height	modifyUnderrate
History	Pay
information	Play
Integrity	process
Interface	Provide
Keys	reach
Leaderboard	Read
Level	referenced
list	remove
Lobby	reset
location	Restrict
Locks	retrieve
Manager	Revoke
Member	Scroll
Modern	Secure
Modes	see
money	Seek
Motivator	Select
multiplayer	Serves
Notoriety	set
Owner	Show
password	Specify
Past	Spend
Payment	Store
Payment No.	take
Peer	Task
People	Travel
Performance	Type
permission	Underrate
PgAdmin	Update
Player	Upload
Player Age	use
Player ID	view
Player Name	visit
Player team	Wait
PostgreSQL	Win
Primary key	Work
Primitive	
Privilege	
Prize	
profile	
Queue	
Racing	
Rank	
Rating	
Record	
Recovery	
reference	
relation	
Rewards	
role	
Rules	
Salary	

Nouns	Verbs
Score	
Security	
Stand	
Statistics	
Status	
Storage	
Survival	
System	
Table	
Team	
Team ID	
Team Name	
Team rank	
Time	
Tournament	
Transaction	
tuple	
username	
vacancy	
Waiting list	
Website	
Weight	
Win rate	
Membership	
Type	
Validity	
Membership No.	

11.2 All Accepted Nouns & Verbs list

Candidate Entity Set	Candidate attribute set	Candidate relationship set
Game	Game id	Play
Player	Game name	History
Game History	Type	Team history
Team	Mode	Member of
User	Price	Open
Manager	Reward	Manages
Employee	Age	Playing in
Account manager	Height	Transaction
Transaction	Weight	Deals
Account	Record number	
Lobby	Date and time	
Worker	Rank	
	Score	
	Level	
	Wins	
	Losses	
	Win rate	
	Team id	
	Team name	
	Number of players	
	User id	
	Name	

Candidate Entity Set	Candidate attribute set	Candidate relationship set
	First name	
	Second name	
	Last name	
	Contact number	
	Email Address	
	Username	
	Password	
	Salary	
	Role	
	Speciality	
	Account number	
	Debit card number	
	Date of joining	
	Balance	
	Transaction id	
	Amount	
	Type	
	Method	
	Status	
	Location	
	Login time	

11.3 All Rejected Nouns list with reason of rejection

NOUN	REJECT REASON	VERB	REJECT REASON
Above	General	Access	Vague
Achievements	Irrelevant	Add	Duplicates
Action	Attributes	Affect	General
Administrator	Irrelevant	Allow	General
Adventure	Attributes	Ask	Irrelevant
Application	General	Automate	Associations
Attribute	Duplicates	Ban	Attributes
Awareness	General	Become	Vague
Balance	Duplicates	Change	Associations
Capacity	Attributes	Check	General
Class	Duplicates	Choose	Vague
Company	General	Compare	Duplicates
Competition	Irrelevant	Compete	Attributes
Consistency	Vague	Connect	Irrelevant
Constraints	Associations	Contact	Attributes
Criteria	Duplicates	Contain	Irrelevant
Contact Number	Duplicates	Decide	General
Customer	Duplicates	Denote	Vague
Database	General	Design	General
Debit Card	Duplicates	Determine	Irrelevant
Desire	Irrelevant	Develop	General
Detail	General	Differ	Vague
Developer	Vague	Discuss	General
Discount	Irrelevant	Display	Duplicates
Documents	Vague	Divide	Duplicates
Engagements	Associations	Entertain	Irrelevant
Engineers	Irrelevant	Find	Vague
Entity	Duplicates	Finish	General

NOUN	REJECT REASON	VERB	REJECT REASON
Events	Vague	Follow	Attributes
Feature	Irrelevant	Foster	Vague
Field	General	Get	Associations
Foreign Key	Associations	Grant	Associations
Game Interfaces	Duplicates	Handle	General
Game play	Irrelevant	Help	General
Game Zone	General	Identify	Associations
Geometry	Irrelevant	Improve	Associations
Goal	Attributes	Leave	General
History	General	Log out	General
Information	General	Log in	General
Integrity	Associations	Make	Duplicates
Interface	Irrelevant	ManageImprove	Associations
Keys	Vague	modifyUnderrate	Duplicates
Level	Attributes	Process	Vague
List	Irrelevant	Provide	Vague
Locks	Vague	Reach	Attributes
Modern	Irrelevant	Read	General
Money	Duplicates	Referenced	Associations
Motivator	Irrelevant	Remove	Duplicates
Multiplayer	Attributes	reset	Irrelevant
Notoriety	General	Restrict	Attributes
Past	General	retrieve	Vague
Payment	Duplicates	Revoke	Vague
Peer	Duplicates	Scroll	Irrelevant
People	General	Secure	General
Performance	Duplicates	see	Vague
Permission	Associations	Seek	Attributes
pgAdmin	General	Serves	Associations
PostgreSQL	General	Show	Duplicates
Primary Key	Associations	Specify	Irrelevant
Primitive	Associations	Spend	Duplicates
Privilege	General	Store	General
Price	Attributes	Take	Duplicates
Profile	Irrelevant	Task	Duplicates
Queue	Irrelevant	Travel	Irrelevant
Racing	Attributes	Type	General
Rating	Duplicate	Underrate	Attributes
Record	General	Use	General
Recovery	General	View	General
Reference	Associations	Visit	Vague
Relation	Associations	Work	Duplicates
Reward	General		
Role	Irrelevant		
Rules	General		
Score	Attributes		
Security	General		
Stand	Associations		
Statistics	General		
Status	Attributes		
Storage	General		
Survival	Irrelevant		
System	General		

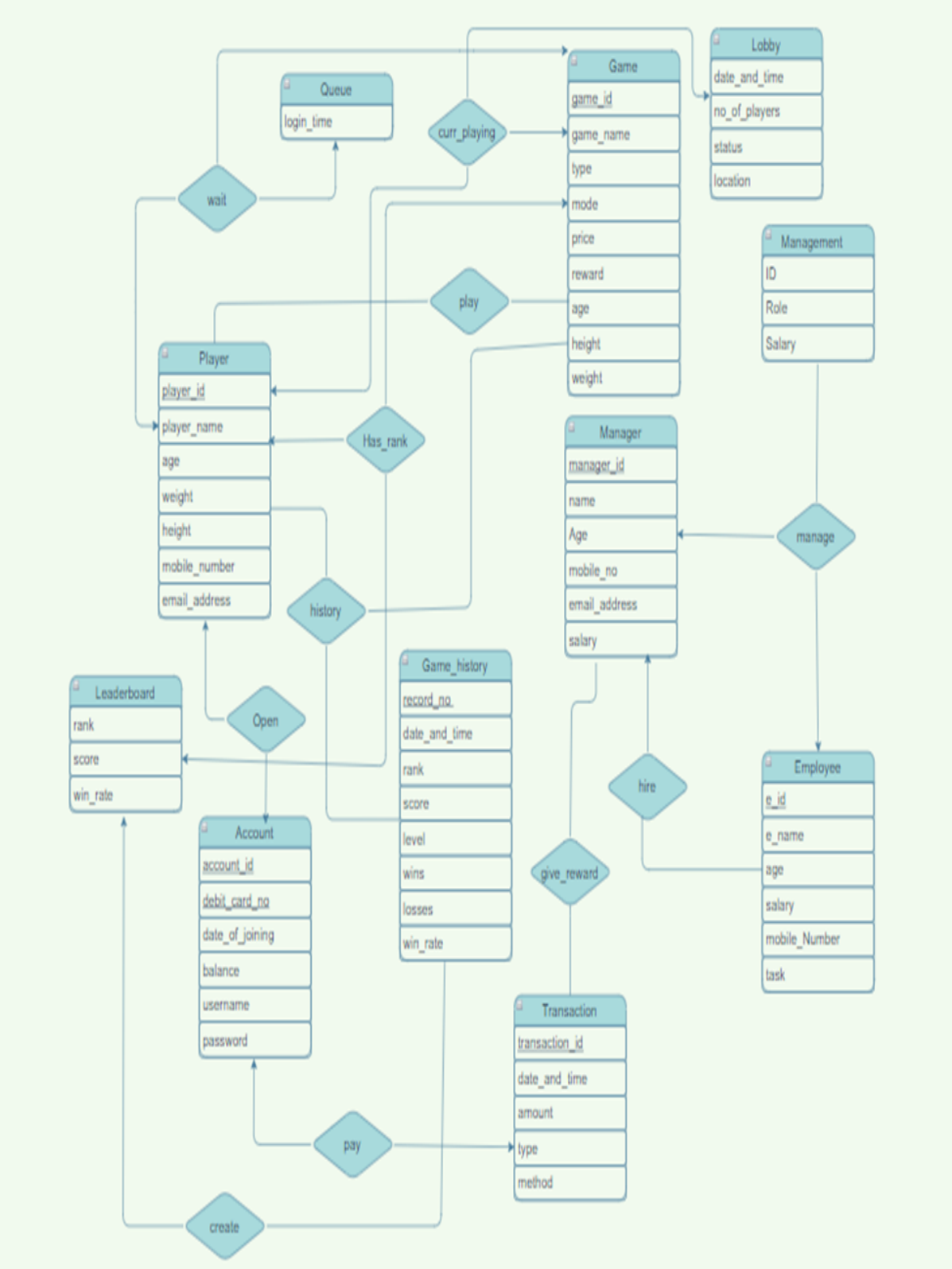
NOUN	REJECT REASON	VERB	REJECT REASON
Table	Duplicates		
Tournaments	General		
Tuple	General		
Vacancy	Attributes		
Website	General		

Section 3

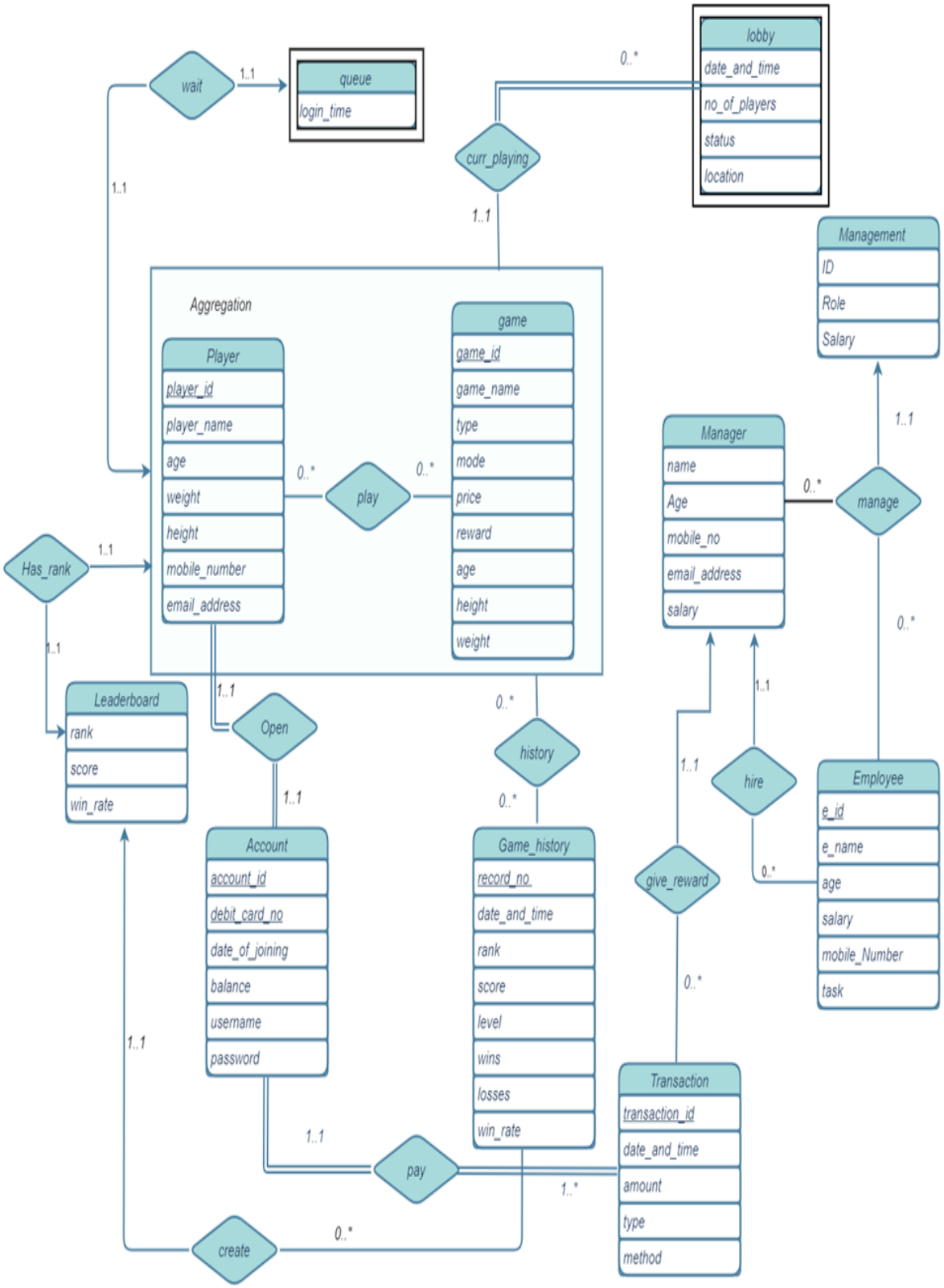
ER-Diagrams all versions

12 ER-Diagrams

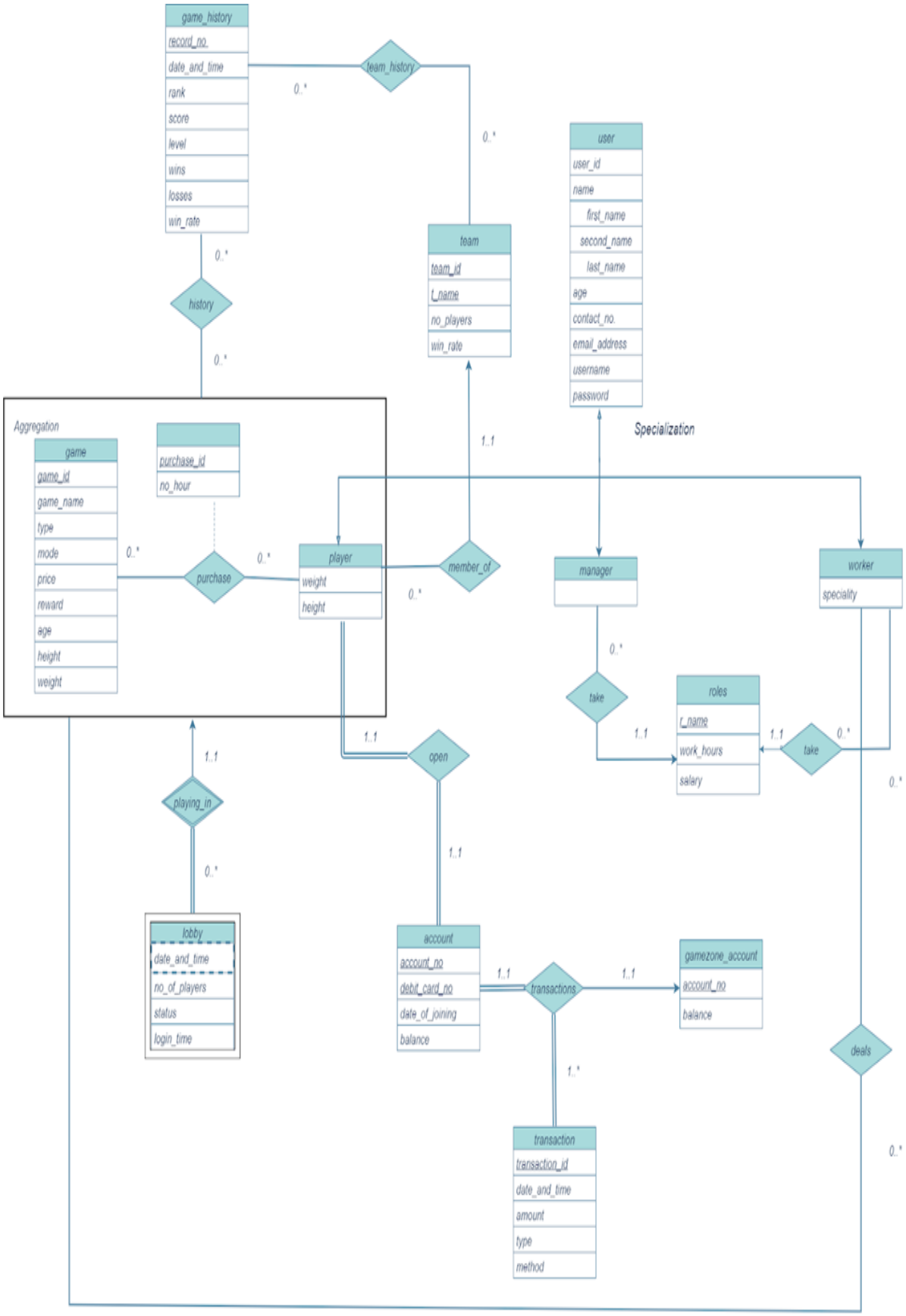
1. ER Diagram V1



2. ER Diagram V2



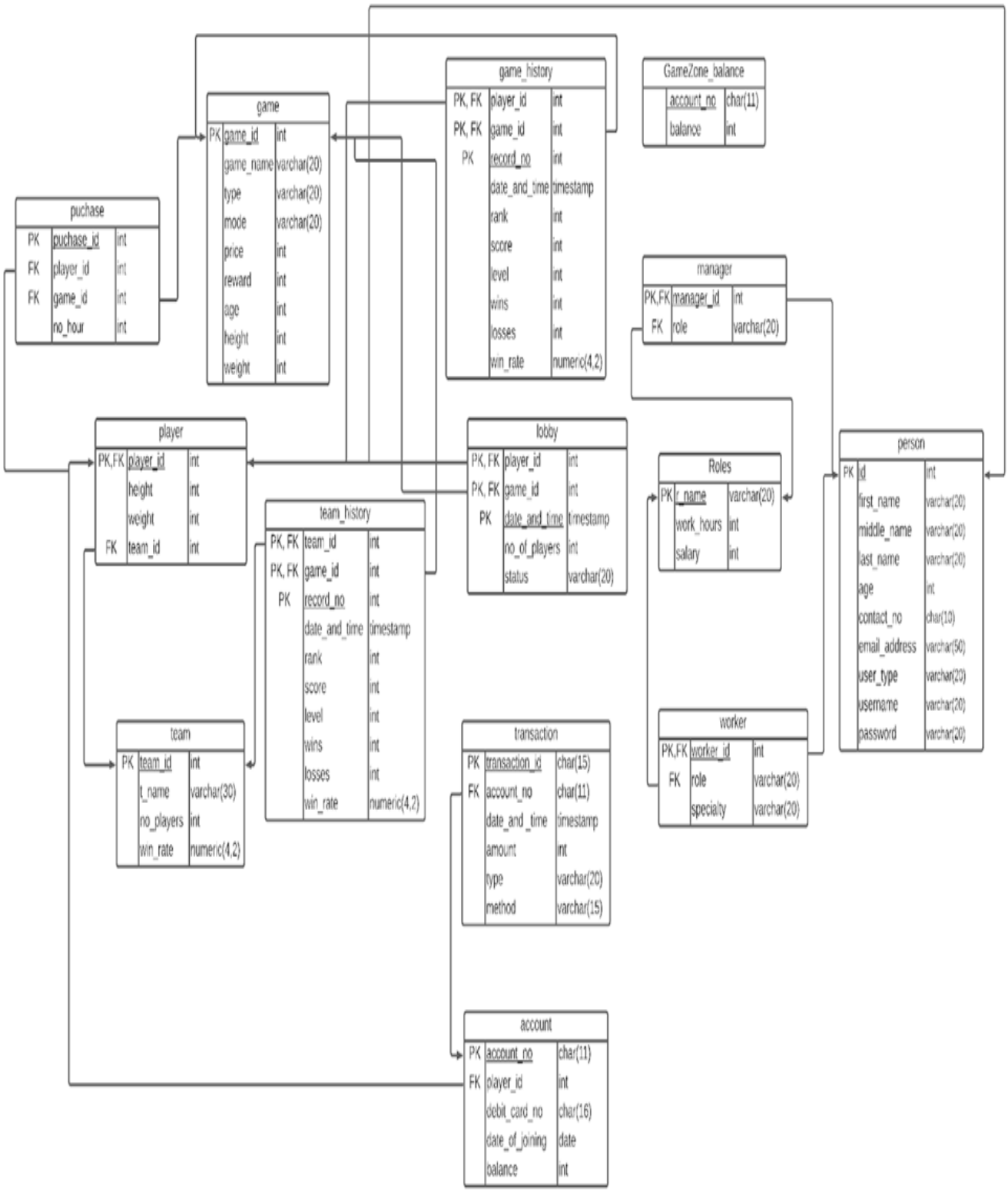
3. Final ER Diagram



Section 4

Conversion of Final ER-Diagram to Relational Model

13 ER-Diagram to Relation Model



Section 5

Normalization and Schema Refinement

14 Normalization and Schema Refinement

- person(**id**, first_name, middle_name, last_name, contact_no, age, email_address, user_type, username, password)
  - Primary key : id
- roles(**r_name**, work_hours, salary)
  - Primary key : r_name
- player(**player_id** , height ,weight ,team_id)
  - Primary key : player_id
  - Foreign key : player_id(player), team_id(team)
- account(**account_no**, player_id, debit_card_no, date_of_joining, balance)
  - Primary key : account_no
  - Foreign key : player_id(player)
- game(**game_id**, game_name, type, mode, price, reward, age, height, weight)
  - Primary key : game_id
- game_table(**table_no**, game_id, capacity)
  - Primary key : table_no
  - Foreign key : game_id (game)
- GameZone_balance(**account_no**, balance)
- purchase(**purchase_id**, player_id, game_id, no_hour)
  - Primary key : purchase_id
  - Foreign key : game_id(game), player_id(player)
- transactions(**transaction_id**, account_no, date_and_time, amount, type, method)
  - Primary key : transaction_id

- Foreign key : account_no(account)

- team(**team_id**, t_name, no_players)
  - Primary key : team_id

- manager(**manager_id**, role)
  - Primary key : manager_id
  - Foreign key : manager_id(person(id)), role(roles(r_name))

- worker(**worker_id**, role)
  - Primary key : worker_id
  - Foreign key : worker_id(person(id)), role(roles(r_name))

- game_history(**record_no**, game_id, player_id, date_and_time, rank, score, level, wins, losses, win_rate)
  - Primary key : record_no
  - Foreign key : game_id(game), player_id(player)
  - Partial Dependencies:
    - game_id, player_id → rank
    - game_id, player_id → wins
    - game_id, player_id → losses
  - Transitive Dependencies:
    - record_no → wins,losses
      wins,losses → win_rate
      record_no → win_rate
    - record_no, player_id,game_id → score
      score → rank
      record_no, player_id,game_id → rank
  - Insert Anomaly: For multiplayer games, if we insert any tuple in game_history then rank will be inconsistent as rank depends on score and number of players playing the game at that time, but at a time we can insert only one tuple. And without rank we can't calculate wins and losses and so winrate.
  - Update Anomaly: If we change the score of any player then it reflects the change in rank of player in a particular game. And also if we change wins and losses, it also reflects a change in win_rate.
  - Delete Anomaly: If we delete any table in game_history then we lose the information about the rank of player for a specific game and how many times the player wins or loses the game.

- 1NF : There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
- 2NF : This relation is not in 2NF form. There is some partial dependency in this relation as mentioned above.
- 3NF : This relation is not in 3NF form. There is some transitive dependency in this relation as mentioned above.
- Updated schema:
  * player_history(**player_id**,**table_no**, **date_and_time**, score, win)
  * player_records(**player_id**,**game_id**, total_games, wins)
- BCNF : This relation is already in BCNF form.

- team_history(**record_no**, team_id, game_id, rank, date_and_time, score, level, wins, losses, win_rate)
  - Primary key : record_no
  - Foreign key : game_id(game), team_id(team)
  - Partial Dependencies:
    * game_id, team_id(team) → rank
    * game_id, team_id(team) → wins
    * game_id, team_id(team) → losses
  - Transitive Dependencies:
    * record_no → wins,losses
      wins,losses → win_rate
      record_no → win_rate
    * record_no, team_id(team),game_id → score
      score → rank
      record_no, team_id(team),game_id → rank
  - Insert Anomaly: For multiplayer games, if we insert any tuple in team_history then rank will be inconsistent as rank depends on score and number of teams playing the game at that time, but at a time we can insert only one tuple. And without rank we can't calculate wins and losses and so win_rate.
  - Update Anomaly: If we change the score of any team then it reflects the change in rank of team in a particular game. And also if we change wins and losses, it also reflects a change in win_rate.
  - Delete Anomaly: If we delete any tuple in team_history then we lose the information about the rank of team for a specific game and how many times the team wins or loses the game.
  - 1NF : There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
  - 2NF : This relation is not in 2NF form. There is some partial dependency in this relation as mentioned above.
  - 3NF : This relation is not in 3NF form. There is some transitive dependency in this relation as mentioned above.
  - Updated schema:
    * team_history(**team_id**,**table_no**, **date_and_time**, score, win)

    - team_records(**team_id**,**game_id**, total_games, wins)
  - BCNF : This relation is already in BCNF form.

- lobby(**date_and_time, player_id, game_id**, no_of_players, status, login_time)
  - Primary key : date_and_time, player_id, game_id
  - Foreign key : game_id(game), player_id(player)
  - Partial Dependencies:
    - data_and_time → login_time
  - Transitive Dependencies:
    - record_no → wins,losses
      wins,losses → win_rate
      record_no → win_rate
    - record_no, player_id,game_id → score
      score → rank
      record_no, player_id,game_id → rank
  - Insert Anomaly: In a chase when two players are playing on the same table, then there are two tuples which have different values of no_of_players, which is inconsistent. So, this is the insert anomaly in this relation.
  - Update Anomaly: if we update no_of_players, then it leads to inconsistency in status of players.
  - Delete Anomaly: If we delete any tuple from this relation then it leads to the same problem in no_of_players and status values as we discussed above.
  - 1NF : There is no multi-valued attribute in this relation. So this relation is already in 1NF form.
  - 2NF : There is a partial dependency in this relation. So this relation is not in 2NF form. We removed the login_time attribute from this relation. It removes partial dependency.
  - 3NF : There is no transitive dependency in this relation. So this relation is already in 3NF form.
  - Redundancy : Here we can measure no_of_player and login time from the value of other attributes.
  - Updated schema:
    - lobby(**date_and_time, player_id, table_no**, status)
  - BCNF : This relation is already in BCNF form.

Section 6

SQL: Final DDL Scripts, Insert statements, 40 SQL Queries with Snapshots of output of each query

15 DDL scripts and Insert Statements

1. Person

```
CREATE TABLE IF NOT EXISTS person (
  id int,
  first_name varchar(20) COLLATE pg_catalog."default" not null,
  middle_name varchar(20) COLLATE pg_catalog."default" not null,
  last_name varchar(20) COLLATE pg_catalog."default" not null,
  age int not null,
  contact_no char(10),
  email_address varchar(20) COLLATE pg_catalog."default" not NULL UNIQUE,
  user_type varchar(20) COLLATE pg_catalog."default" not null,
  username varchar(20) COLLATE pg_catalog."default" not null UNIQUE,
  password varchar(20) COLLATE pg_catalog."default" NOT NULL,
  PRIMARY KEY (id),
  CHECK(user_type in ('manager','player','worker'))
  );

  COPY
  person(id, first_name, middle_name, last_name, age, contact_no, email_address,
      user_type, username, password)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\person.csv'
  DELIMITER ','
  CSV HEADER;
```

2. team

```
 CREATE TABLE IF NOT EXISTS team (
  team_id int,
  t_name varchar(30) not null unique,
  no_players int not null CHECK(no_players > 1),
  PRIMARY KEY (team_id)
 );

  COPY
  team(team_id, t_name, no_players)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\team.csv'
  DELIMITER ','
  CSV HEADER;
```

3. Roles

```
    CREATE TABLE IF NOT EXISTS Roles (
      r_name varchar(20) COLLATE pg_catalog."default",
      workhours int not null check(workhours > 0),
      salary int not null check(salary > 0),
      PRIMARY KEY (r_name)
    );

  COPY
  roles(r_name, work_hours, salary)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\roles.csv'
  DELIMITER ','
  CSV HEADER;
```

4. player

```
    CREATE TABLE IF NOT EXISTS player (
      player_id int,
      height int not NULL,
      weight int not NULL,
      team_id int,
      PRIMARY KEY (player_id),
      FOREIGN KEY (player_id) REFERENCES person(id) MATCH SIMPLE ON UPDATE CASCADE ON
          DELETE CASCADE,
      FOREIGN KEY (team_id) REFERENCES team(team_id) MATCH SIMPLE ON UPDATE CASCADE ON
          DELETE CASCADE
    );

  COPY
  player(player_id , height ,weight ,team_id)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\player.csv'
  DELIMITER ','
  CSV HEADER;
```

5. manager

```
CREATE TABLE IF NOT EXISTS manager (
  manager_id int,
  role varchar(20) COLLATE pg_catalog."default" NOT NULL,
  PRIMARY KEY (manager_id),
  FOREIGN KEY (role) REFERENCES roles(r_name) MATCH SIMPLE ON UPDATE CASCADE ON DELETE
      CASCADE,
  FOREIGN KEY (manager_id) REFERENCES person(id) MATCH SIMPLE ON UPDATE CASCADE ON DELETE
      CASCADE
);

  COPY
  manager(manager_id, role)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\manager.csv'
  DELIMITER ','
  CSV HEADER;
```

6. worker

```
CREATE TABLE IF NOT EXISTS worker (
  worker_id int,
  role varchar(20) COLLATE pg_catalog."default" not NULL,
  PRIMARY KEY (worker_id),
  FOREIGN KEY (role) REFERENCES roles(r_name) MATCH SIMPLE ON UPDATE CASCADE ON DELETE
      CASCADE,
  FOREIGN KEY (worker_id) REFERENCES person(id) MATCH SIMPLE ON UPDATE CASCADE ON DELETE
      CASCADE
);

  COPY
  worker(worker_id, role)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\worker.csv'
  DELIMITER ','
  CSV HEADER;
```

7. account

```
CREATE TABLE IF NOT EXISTS account (
  account_no char(11),
  player_id int not NULL UNIQUE,
  debit_card_no char(16) not NULL ,
  date_of_joining date not NULL,
  balance int default 1000,
  PRIMARY KEY (account_no),
  FOREIGN KEY (player_id) REFERENCES player(player_id) MATCH SIMPLE ON UPDATE CASCADE ON
      DELETE CASCADE
);

  COPY
  account(account_no, player_id, debit_card_no, date_of_joining, balance)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\account.csv'
  DELIMITER ','
  CSV HEADER;
```

8. transactions

```
CREATE TABLE IF NOT EXISTS transactions (
  transaction_id char(15),
  account_no char(11) not NULL,
  date_and_time timestamp not NULL,
  amount int not NULL,
  type varchar(20) COLLATE pg_catalog."default" not NULL,
  method varchar(15),
  PRIMARY KEY (transaction_id),
  FOREIGN KEY (account_no) REFERENCES account(account_no) MATCH SIMPLE ON UPDATE RESTRICT
      ON DELETE RESTRICT
);

  COPY
  transactions(transaction_id, account_no, date_and_time, amount, type, method)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\transactions.csv'
  DELIMITER ','
  CSV HEADER;
```

9. game

```
CREATE TABLE IF NOT EXISTS game (
  game_id int,
  game_name varchar(20) COLLATE pg_catalog."default" not null unique,
  type varchar(20) COLLATE pg_catalog."default" ,
  mode varchar(20) COLLATE pg_catalog."default" not null,
  price int not null CHECK(price > 0),
  reward int default 0,
  age int default 5,
  height int,
  weight int,
  PRIMARY KEY (game_id),
  CHECK(mode in ('online','offline'))
);

  COPY
```

```
  game(game_id, game_name, type, mode, price, reward, age, height, weight)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\game.csv'
  DELIMITER ','
  CSV HEADER;
```

10. purchase

```
CREATE TABLE IF NOT EXISTS purchase (
  purchase_id int,
  player_id int,
  game_id int ,
  no_hours int not null CHECK(no_hours > 0),
  PRIMARY KEY (purchase_id),
  FOREIGN KEY (player_id) REFERENCES player(player_id) MATCH SIMPLE ON UPDATE CASCADE ON
      DELETE CASCADE,
  FOREIGN KEY (game_id) REFERENCES game(game_id) MATCH SIMPLE ON UPDATE CASCADE ON DELETE
      CASCADE
);

  COPY
  purchase(purchase_id, player_id, game_id, no_hour)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\purchase.csv'
  DELIMITER ','
  CSV HEADER;
```

11. game_table

```
CREATE TABLE IF NOT EXISTS game_table (
  table_no int,
  game_id int,
  capacity int not null check(capacity>0),
  PRIMARY KEY (table_no),
  FOREIGN KEY (game_id) REFERENCES game(game_id) MATCH SIMPLE ON UPDATE CASCADE ON DELETE
      CASCADE
);

  COPY
  game_table(table_no, game_id, capacity)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\game_table.csv'
  DELIMITER ','
  CSV HEADER;
```

12. player_history

```
CREATE TABLE IF NOT EXISTS player_history (
  player_id int,
  table_no int,
  date_and_time timestamp not NULL,
  score int not NULL CHECK(score >=0),
  win boolean default false,
  PRIMARY KEY (player_id,table_no,date_and_time),
  FOREIGN KEY (table_no) REFERENCES game_table(table_no) MATCH SIMPLE ON UPDATE CASCADE
      ON DELETE CASCADE,
  FOREIGN KEY (player_id) REFERENCES player(player_id) MATCH SIMPLE ON UPDATE CASCADE ON
      DELETE CASCADE
);
```

```
  COPY
  player_history(player_id, table_no, date_and_time, score, win)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\player_history.csv'
  DELIMITER ','
  CSV HEADER;
```

13. player_records

```
CREATE TABLE IF NOT EXISTS player_records (
  player_id int,
  game_id int,
  total_games int default 0,
  wins int default 0 check(wins<=total_games),
  PRIMARY KEY (player_id,game_id),
  FOREIGN KEY (game_id) REFERENCES game(game_id) MATCH SIMPLE ON UPDATE CASCADE ON DELETE
      CASCADE,
  FOREIGN KEY (player_id) REFERENCES player(player_id) MATCH SIMPLE ON UPDATE CASCADE ON
      DELETE CASCADE
);

  COPY
  player_records(player_id, game_id, total_games, wins)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\player_records.csv'
  DELIMITER ','
  CSV HEADER;
```

14. team_history

```
CREATE TABLE IF NOT EXISTS team_history (
  team_id int,
  table_no int,
  date_and_time timestamp not NULL,
  score int not NULL CHECK(score >=0),
  win boolean default false,
  PRIMARY KEY (team_id,table_no),
  FOREIGN KEY (table_no) REFERENCES game_table(table_no) MATCH SIMPLE ON UPDATE CASCADE
      ON DELETE CASCADE,
  FOREIGN KEY (team_id) REFERENCES team(team_id) MATCH SIMPLE ON UPDATE CASCADE ON DELETE
      CASCADE
);

  COPY
  team_history(team_id, table_no, date_and_time, score, win)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\team_history.csv'
  DELIMITER ','
  CSV HEADER;
```

15. team_records

```
CREATE TABLE IF NOT EXISTS team_records (
  team_id int,
  game_id int,
  total_games int default 0,
  wins int default 0 check(wins<=total_games),
  PRIMARY KEY (team_id,game_id),
```

```
  FOREIGN KEY (game_id) REFERENCES game(game_id) MATCH SIMPLE ON UPDATE CASCADE ON DELETE
      CASCADE,
  FOREIGN KEY (team_id) REFERENCES team(team_id) MATCH SIMPLE ON UPDATE CASCADE ON DELETE
      CASCADE
);

  COPY
  team_records(team_id, game_id, total_games, wins)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\team_records.csv'
  DELIMITER ','
  CSV HEADER;
```

16. lobby

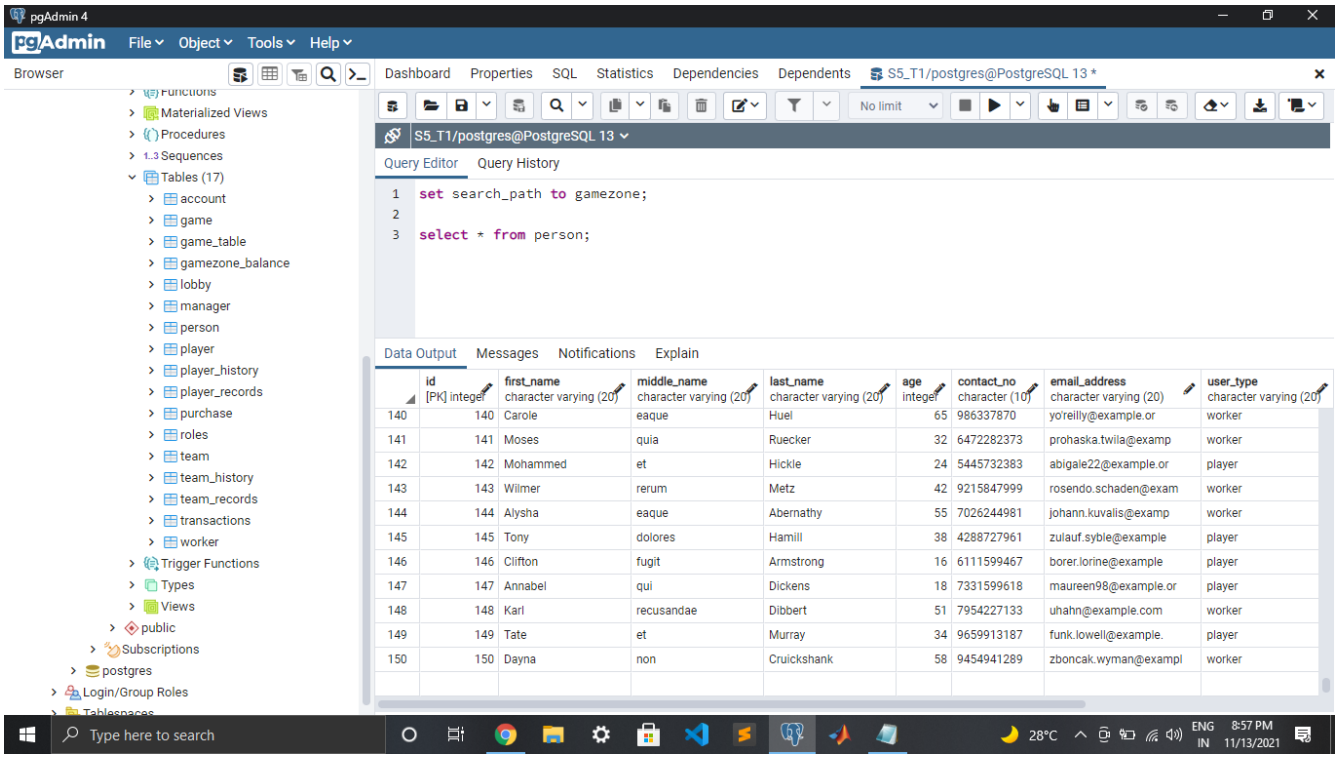
```
CREATE TABLE IF NOT EXISTS lobby (
  player_id int,
  table_no int,
  date_and_time timestamp,
  status varchar(20) COLLATE pg_catalog."default",
  PRIMARY KEY (player_id, table_no, date_and_time),
  FOREIGN KEY (player_id) REFERENCES player(player_id) MATCH SIMPLE ON UPDATE CASCADE ON
      DELETE CASCADE,
  FOREIGN KEY (table_no) REFERENCES game_table(table_no) MATCH SIMPLE ON UPDATE CASCADE
      ON DELETE CASCADE,
  CHECK (status in ('playing','waiting'))
);

  COPY
  lobby(date_and_time, player_id, table_no, status)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\lobby.csv'
  DELIMITER ','
  CSV HEADER;
```

17. GameZone_balance

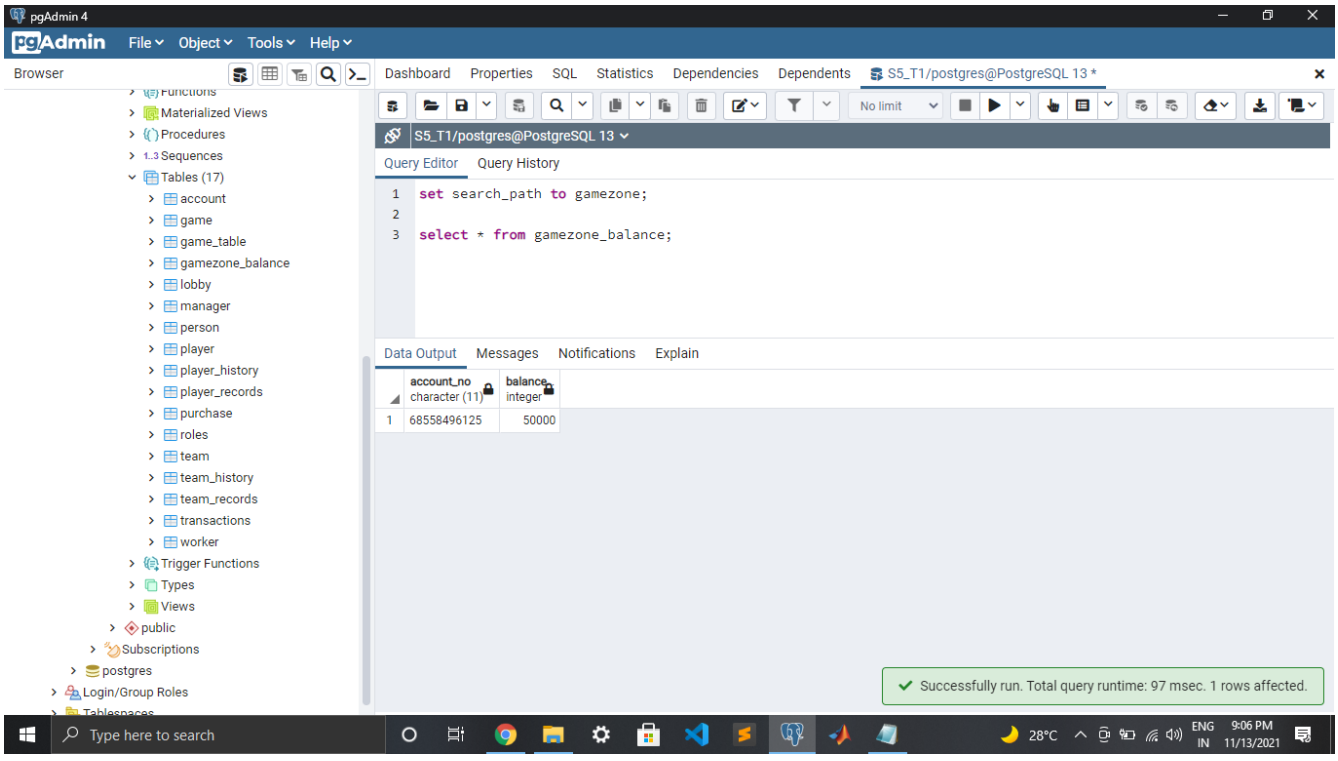
```
CREATE TABLE IF NOT EXISTS GameZone_balance (
   account_no char(11) NOT NULL,
   balance int default 50000
);

  COPY
  GameZone_balance(account_no, balance)
  FROM 'C:\Users\Parth Prajapati\Desktop\DBMS_PROJECT\Gamezone_data\team.csv'
  DELIMITER ','
  CSV HEADER;
```

Data Snapshots

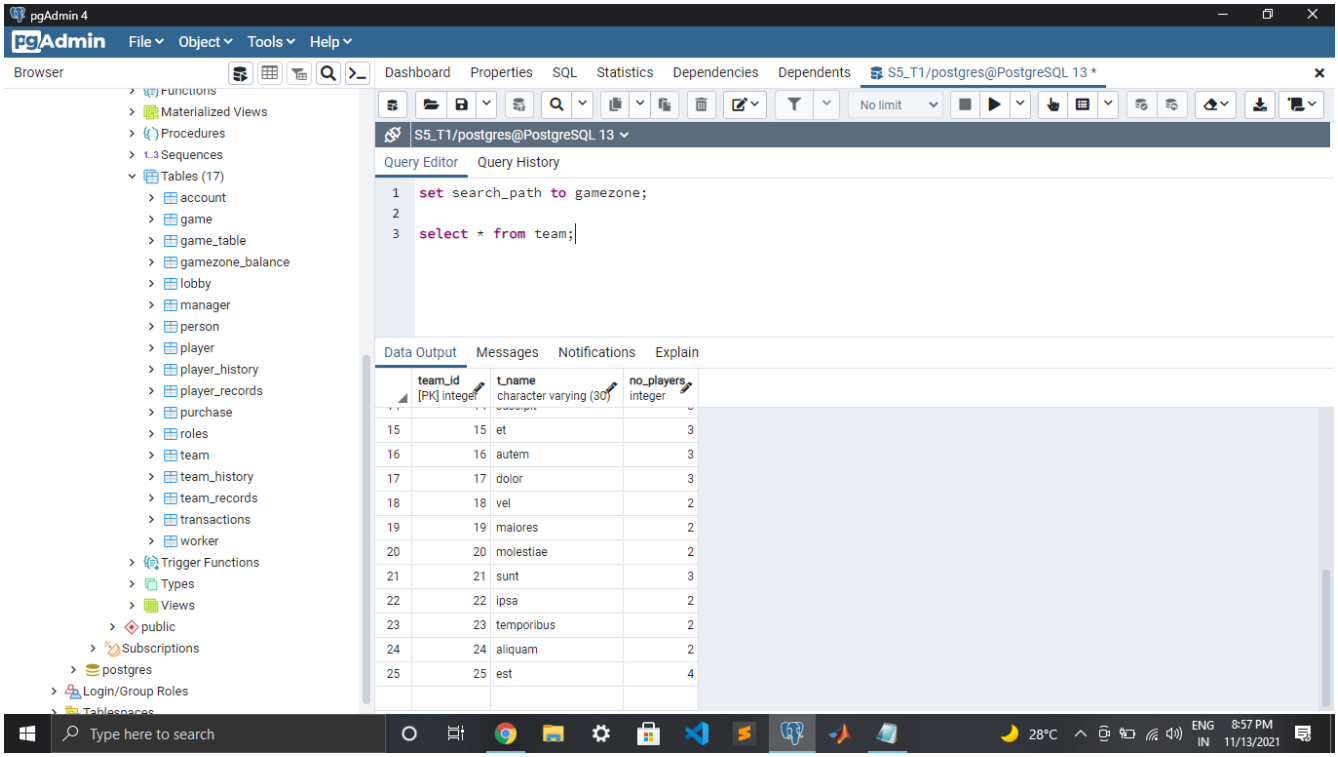
1. Person



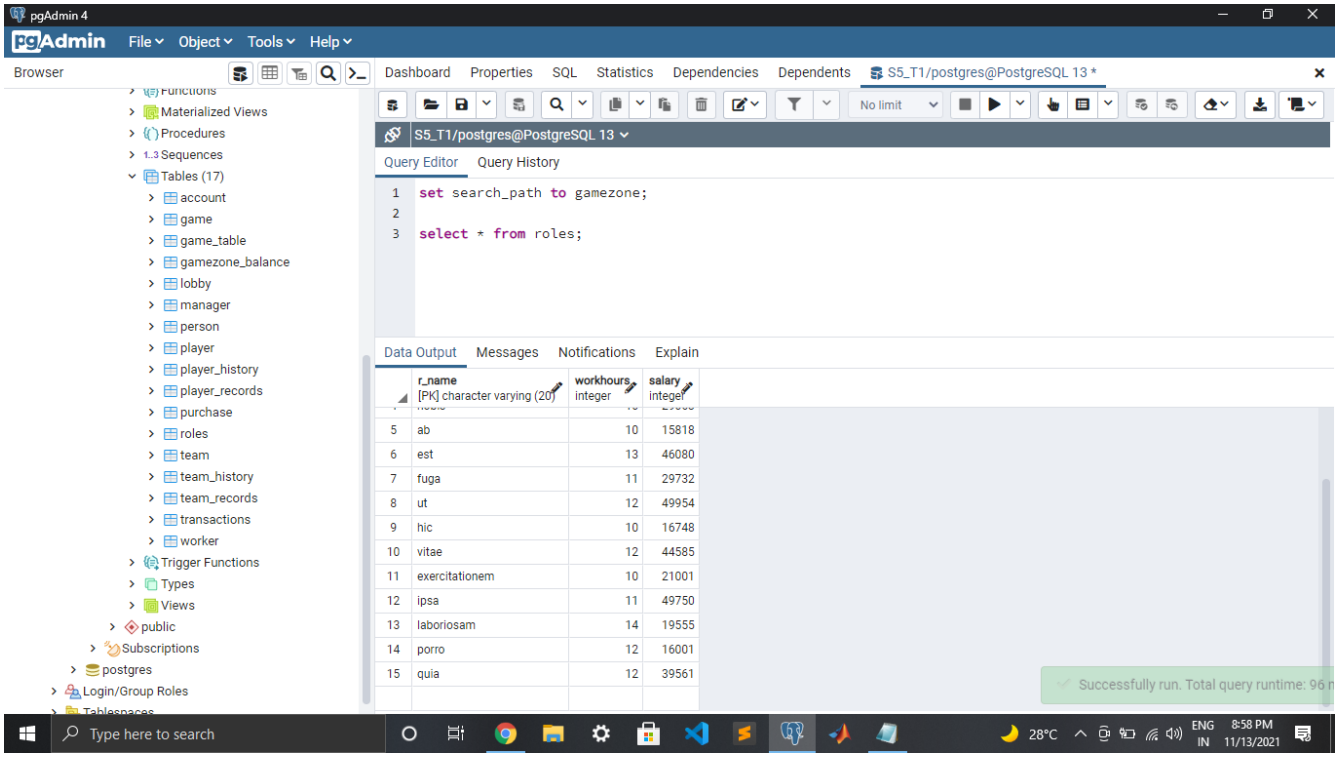
2. GameZone_balance



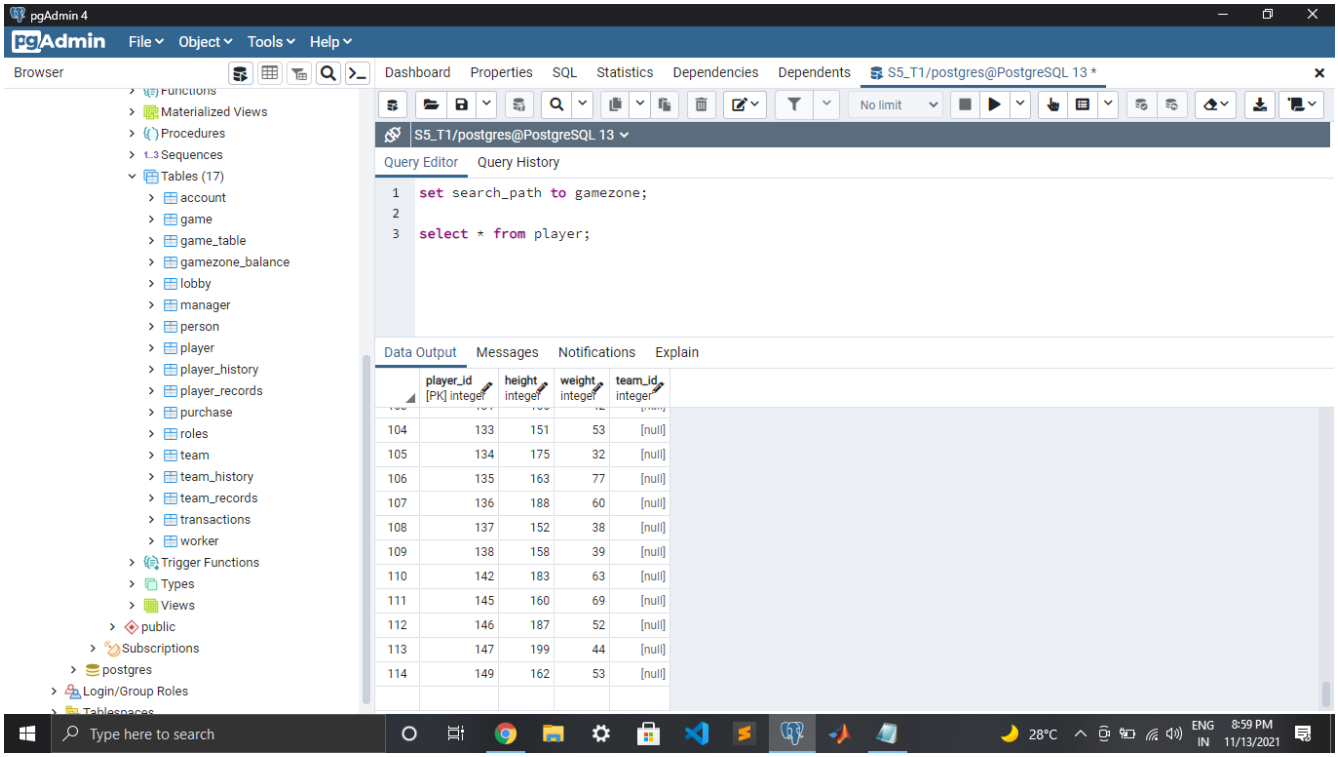
3. team



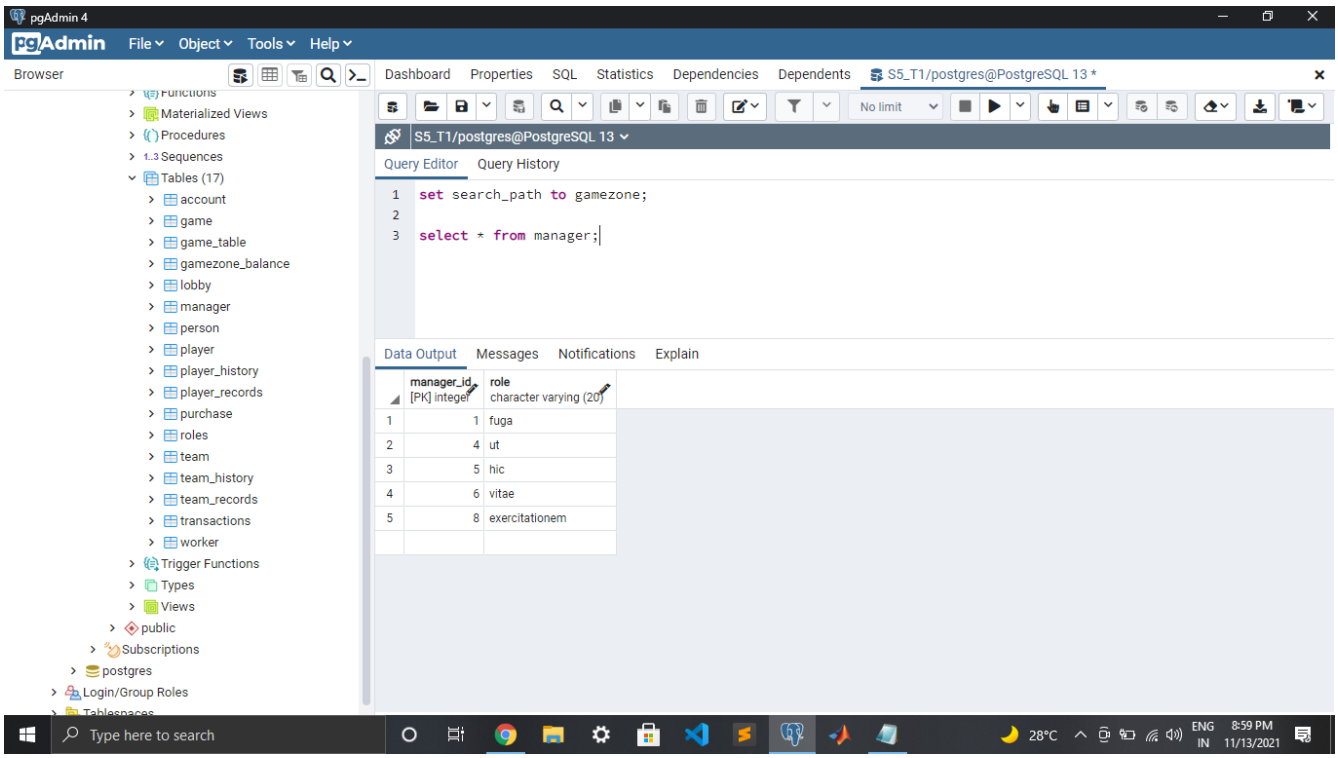
4. Roles



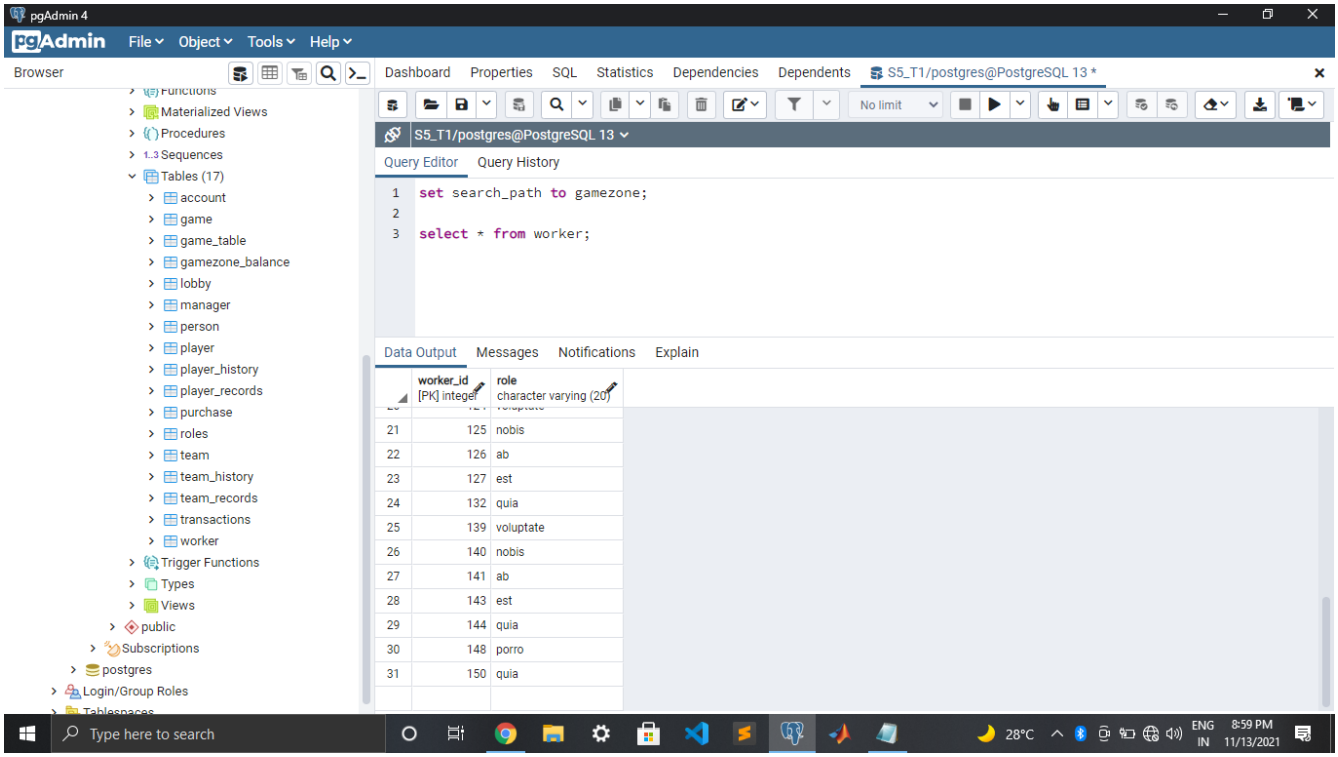
5. player



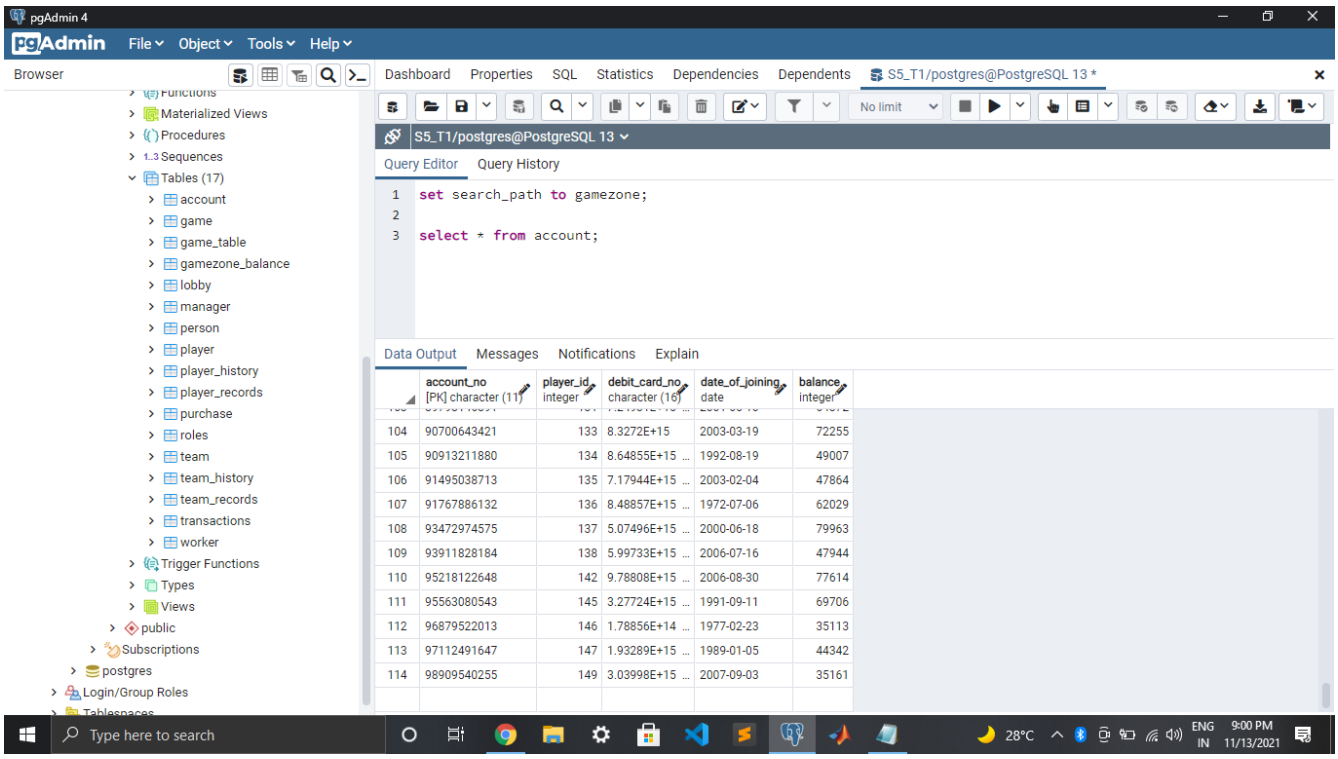
6. manager



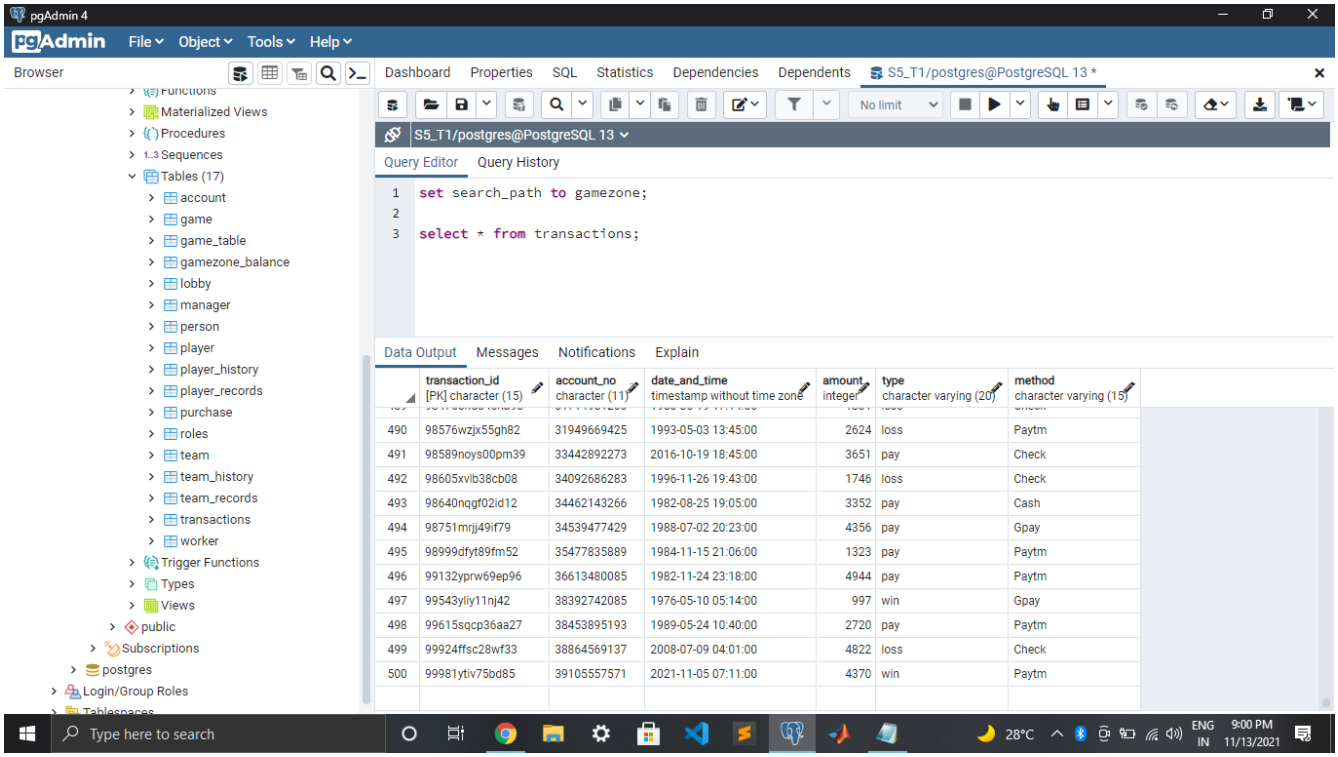
7. worker



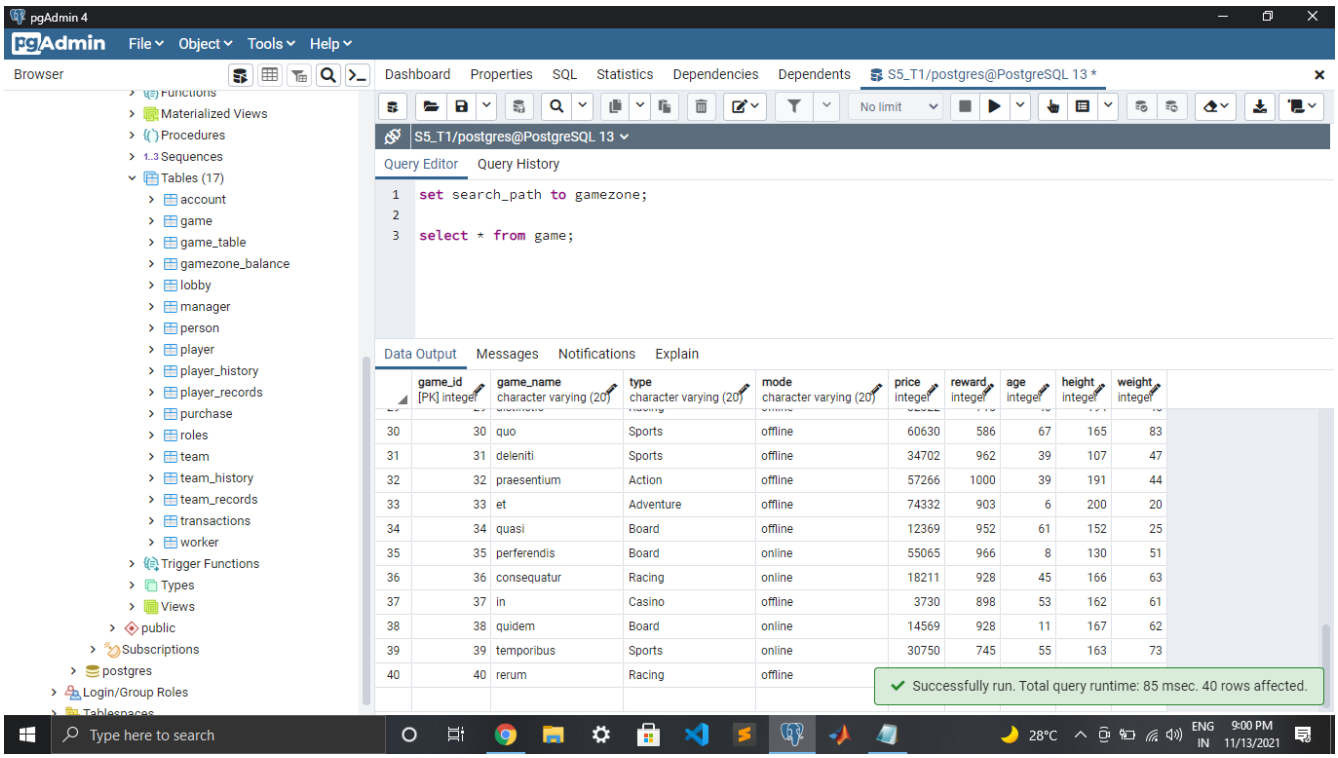
8. account



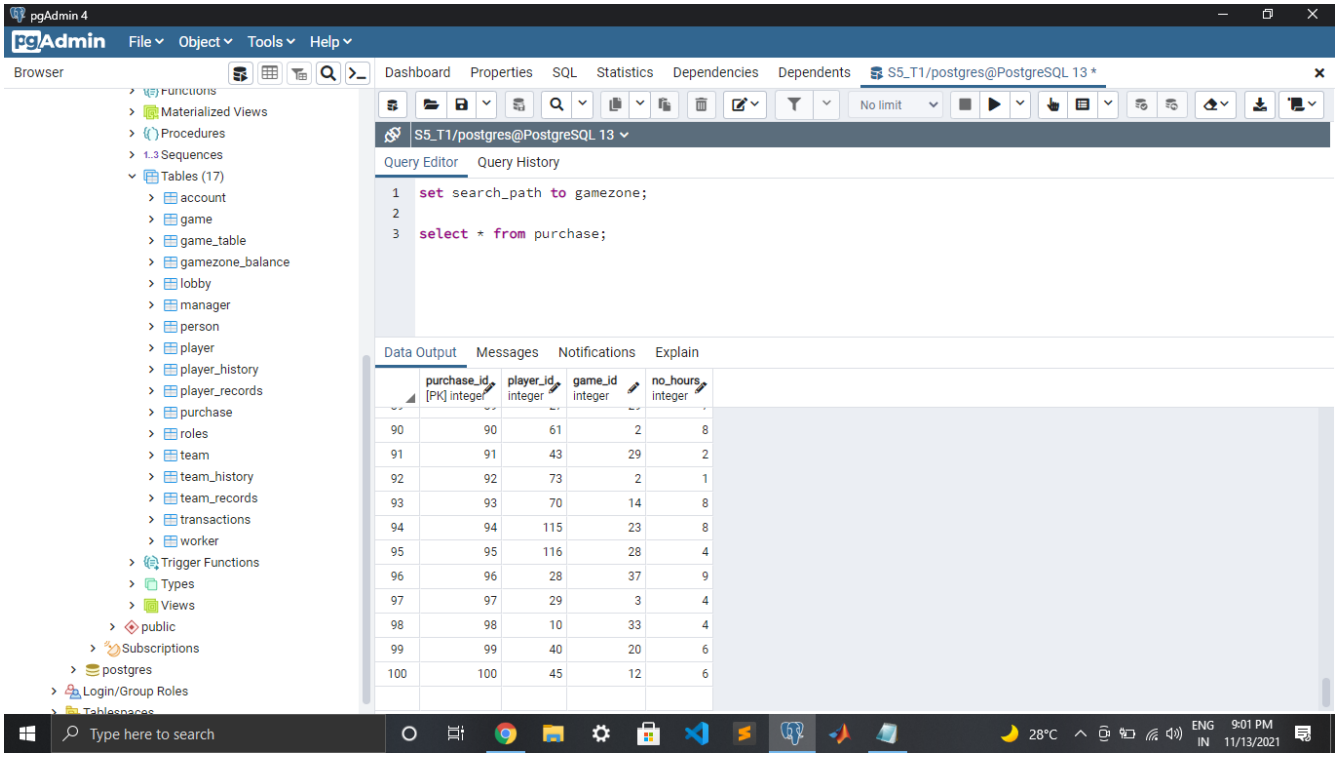
9. transactions



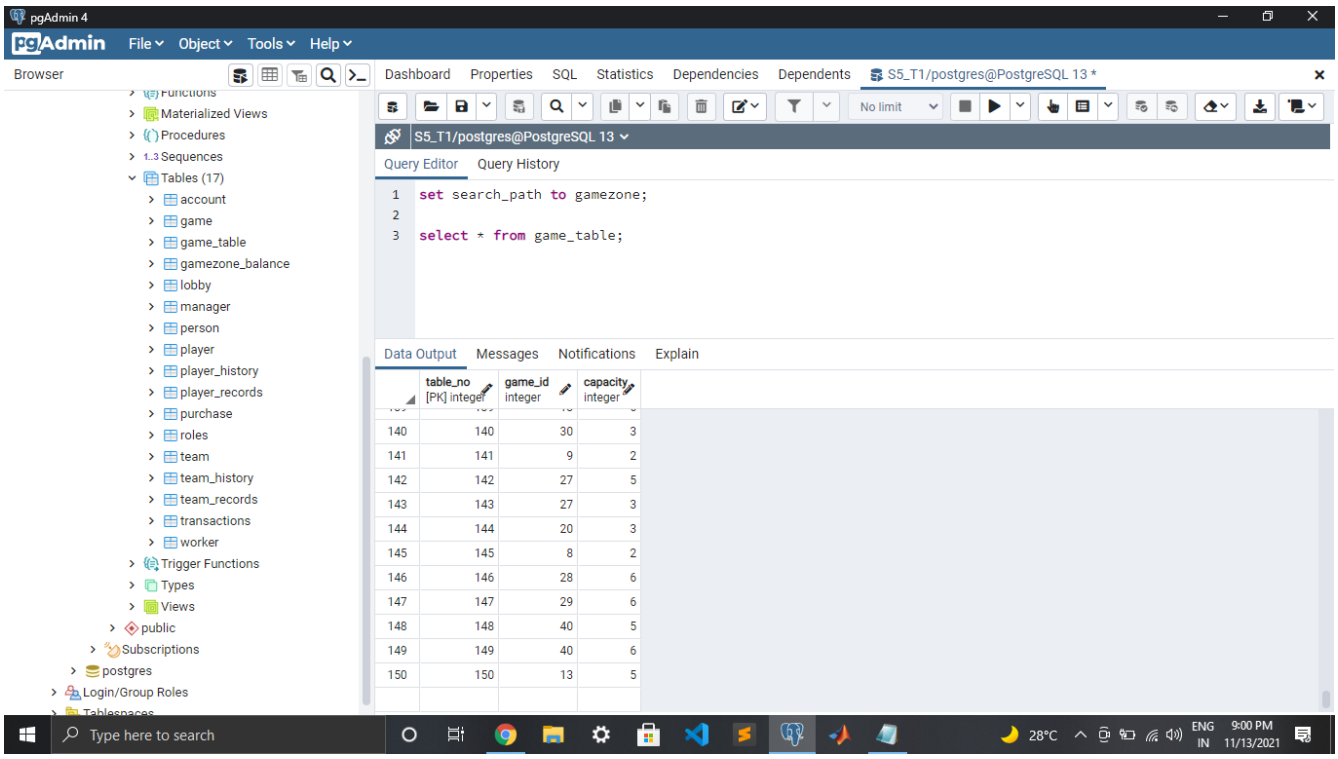
10. game



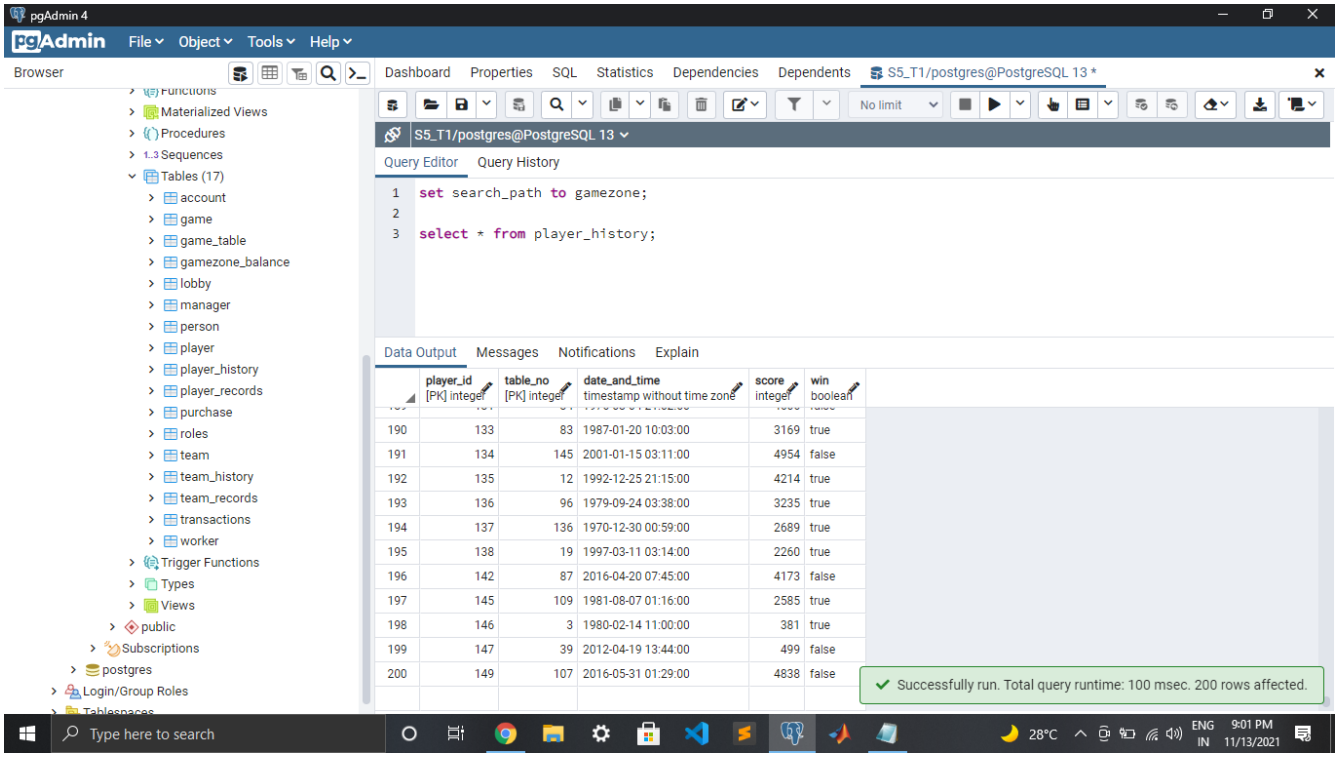
11. purchase



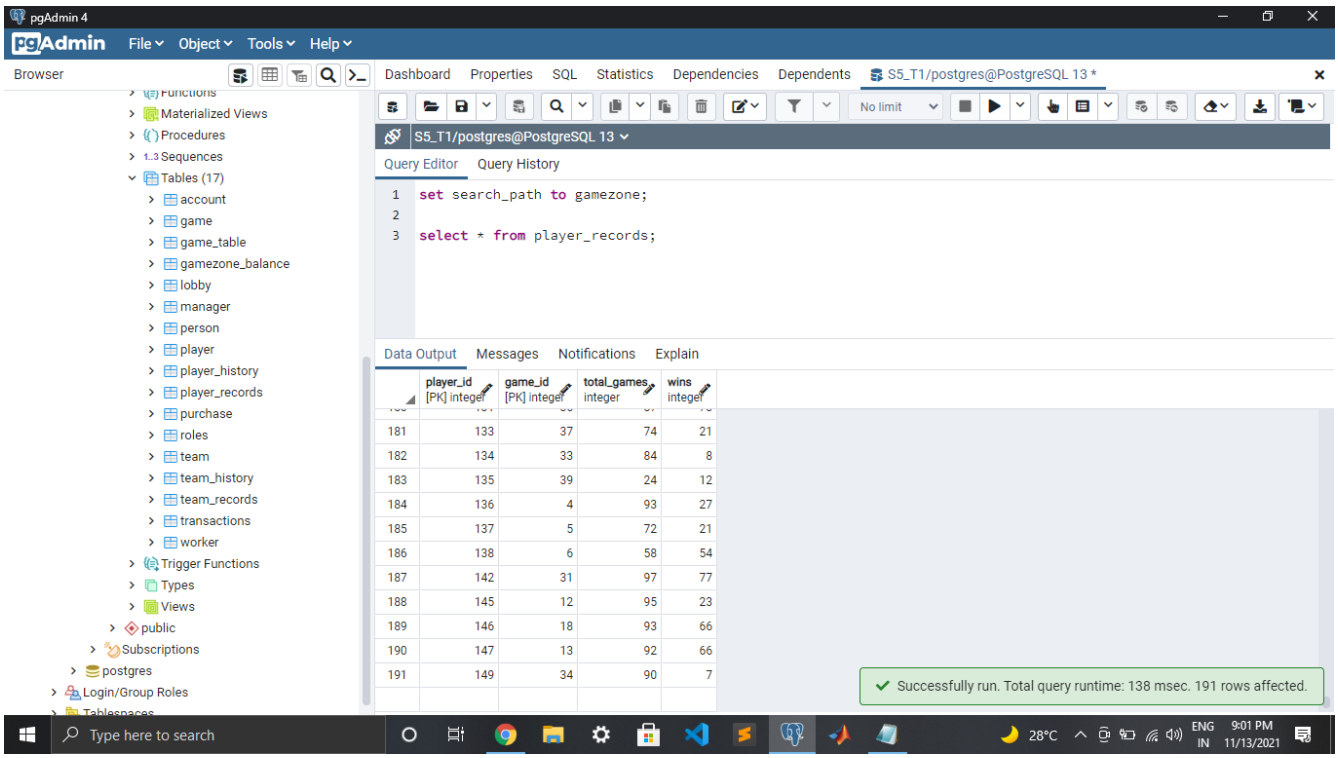
12. game_table



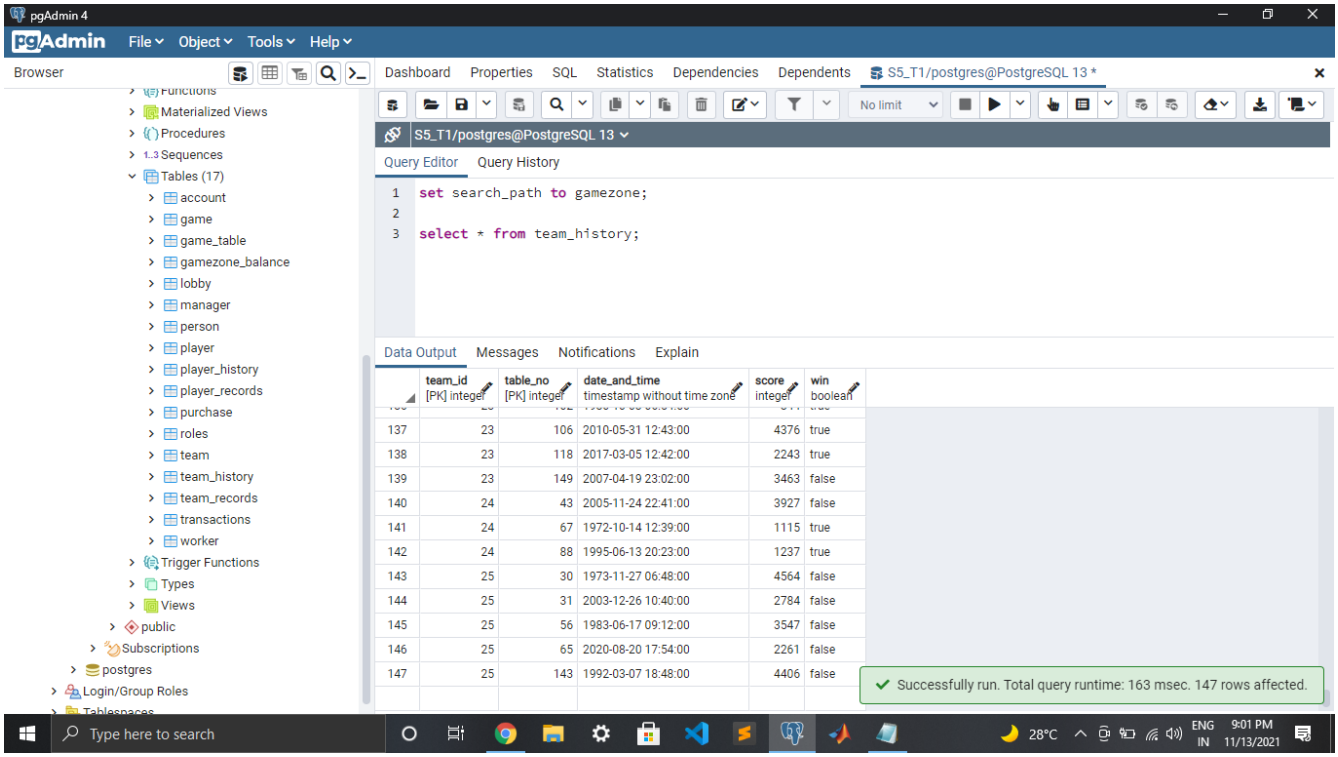
13. player_history



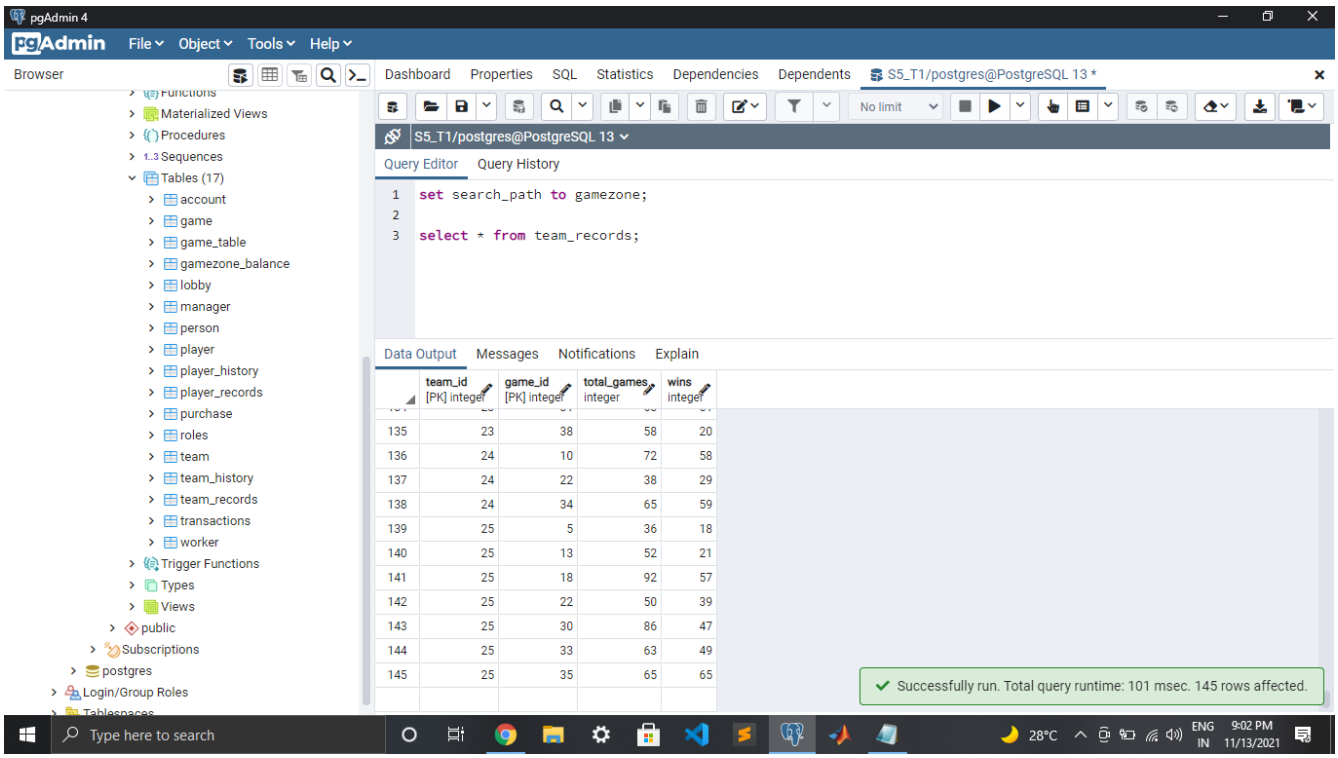
14. player_records



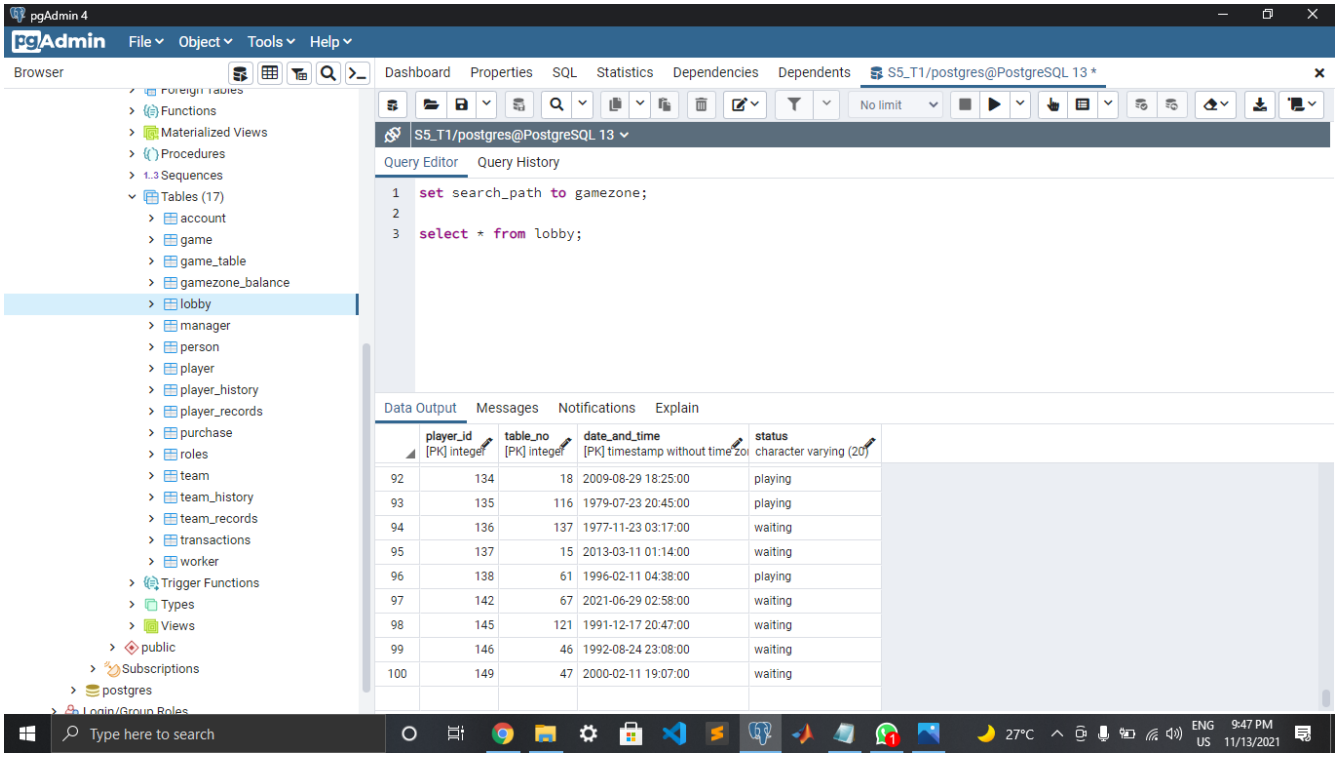
15. team_history



16. team_records



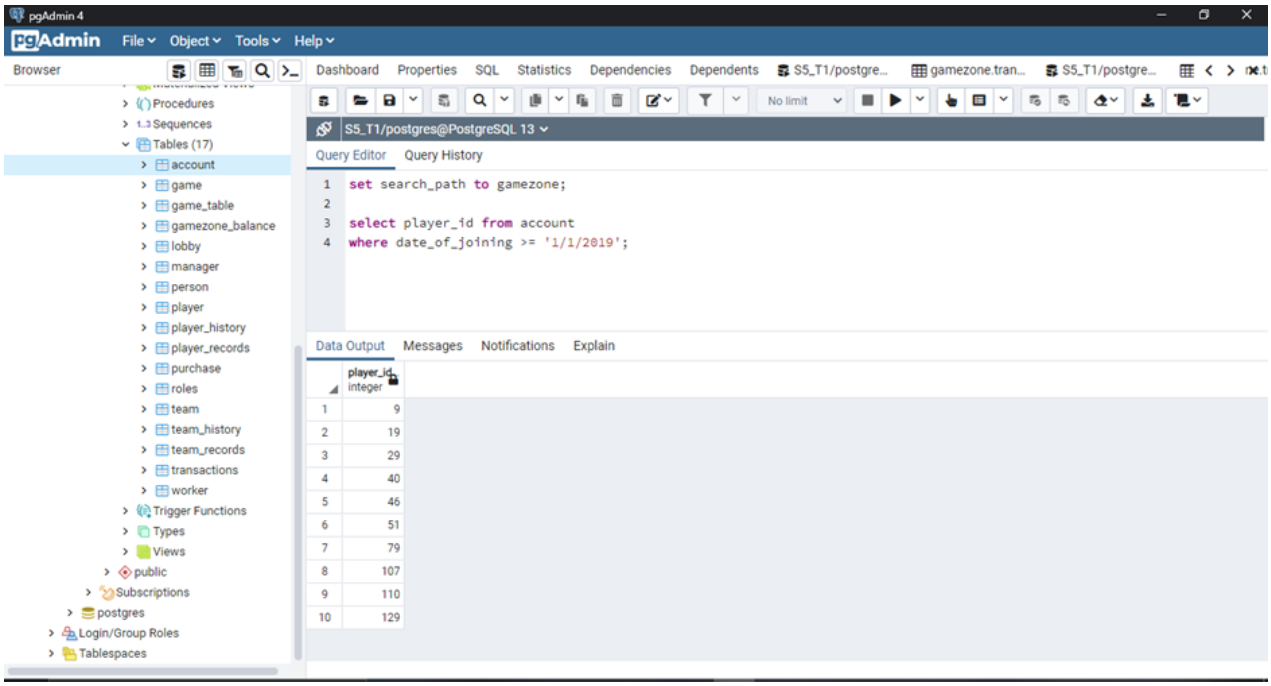
17. lobby


pgAdmin 4
pgAdmin
File
Object
Tools
Help
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
S5_T1/postgres@PostgreSQL 13 *
No limit
S5_T1/postgres@PostgreSQL 13
Query Editor
Query History
1 set search_path to gamezone;
2
3 select * from lobby;
Functions
Materialized Views
Procedures
Sequences
Tables (17)
account
game
game_table
gamezone_balance
lobby
manager
person
player
player_history
player_records
purchase
roles
team
team_history
team_records
transactions
worker
Trigger Functions
Types
Views
public
Subscriptions
postgres
Data Output
Messages
Notifications
Explain
player_id [PK] integer
table_no [PK] integer
date_and_time [PK] timestamp without time zo
status character varying (20)
92 134 18 2009-08-29 18:25:00 playing
93 135 116 1979-07-23 20:45:00 playing
94 136 137 1977-11-23 03:17:00 waiting
95 137 15 2013-03-11 01:14:00 waiting
96 138 61 1996-02-11 04:38:00 playing
97 142 67 2021-06-29 02:58:00 waiting
98 145 121 1991-12-17 20:47:00 waiting
99 146 46 1992-08-24 23:08:00 waiting
100 149 47 2000-02-11 19:07:00 waiting
Type here to search
27°C
ENG
US
9:47 PM
11/13/2021

16 SQL queries

1. List all the players who have joined the gamezone on or after 1/1/2019.

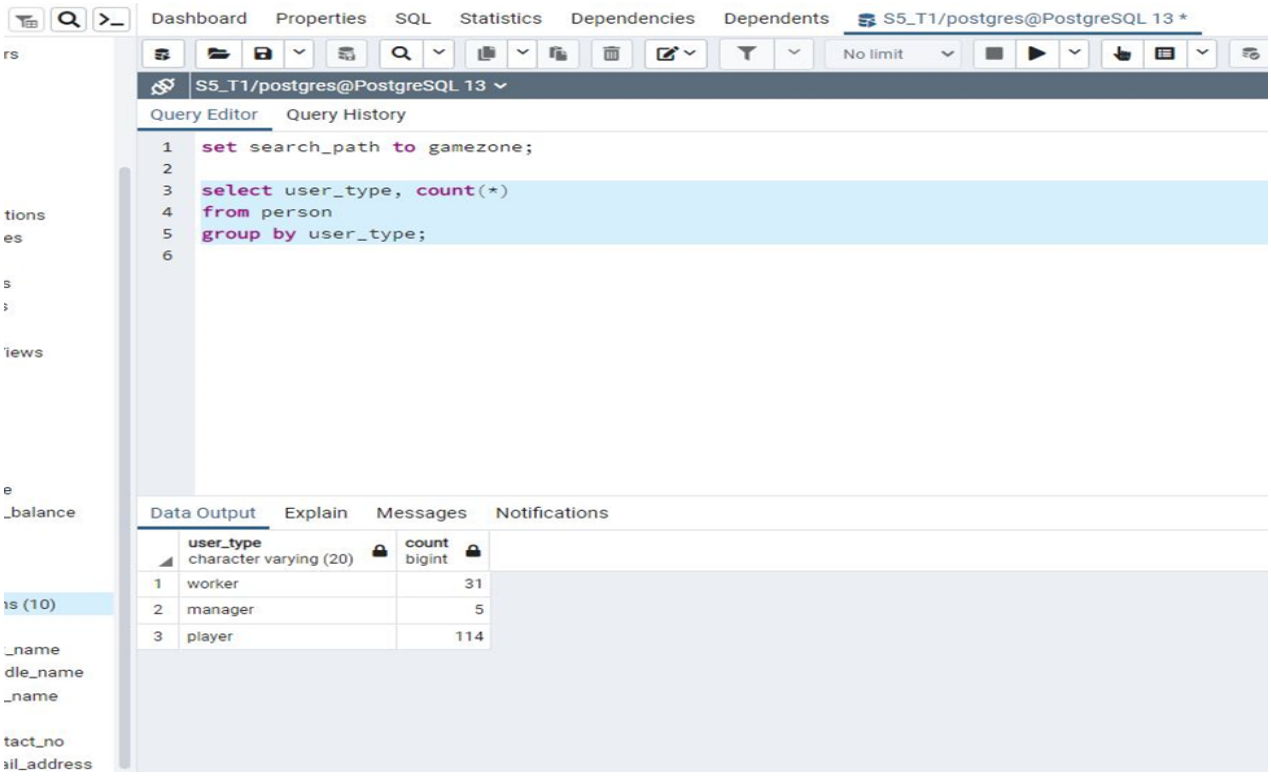
```
select player_id
from account
where date_of_joining >= '1/1/2019';
```



- Number of Tuple = 10

2. Count the number of players, managers and employees present in the person table.

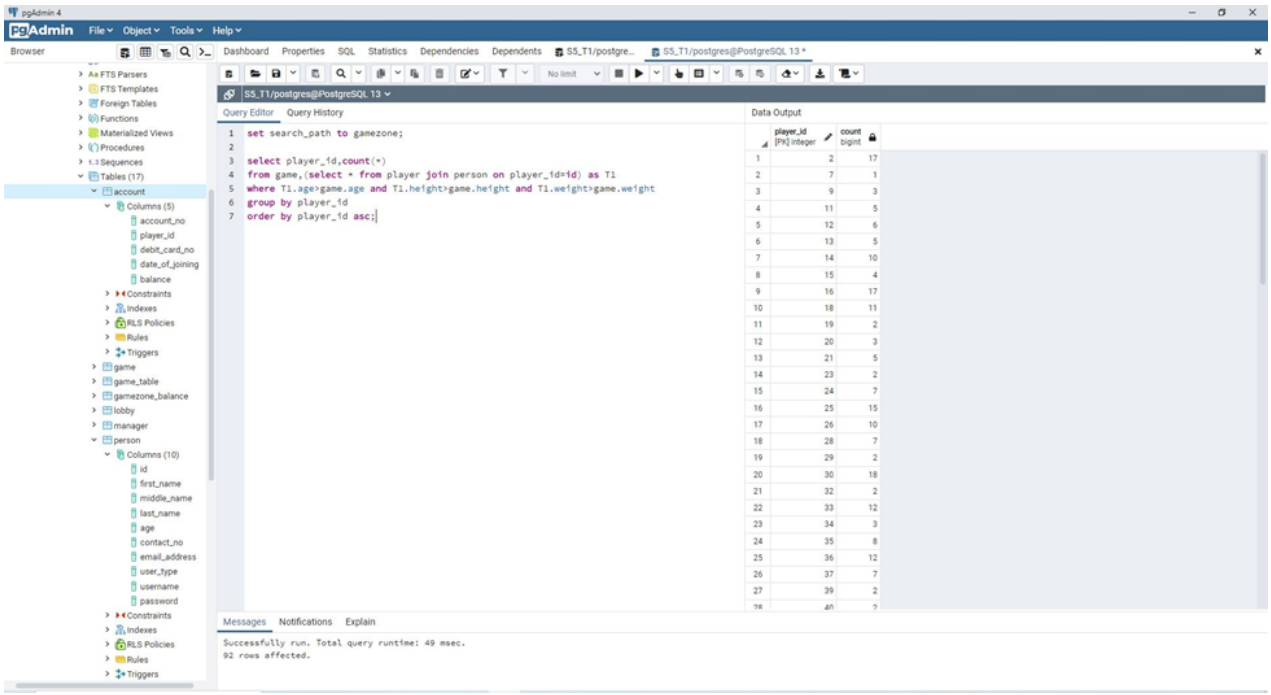
```
select user_type, count(*)
from person
group by user_type;
```



- Number of Tuple = 3

3. Count the number of games that player can play.

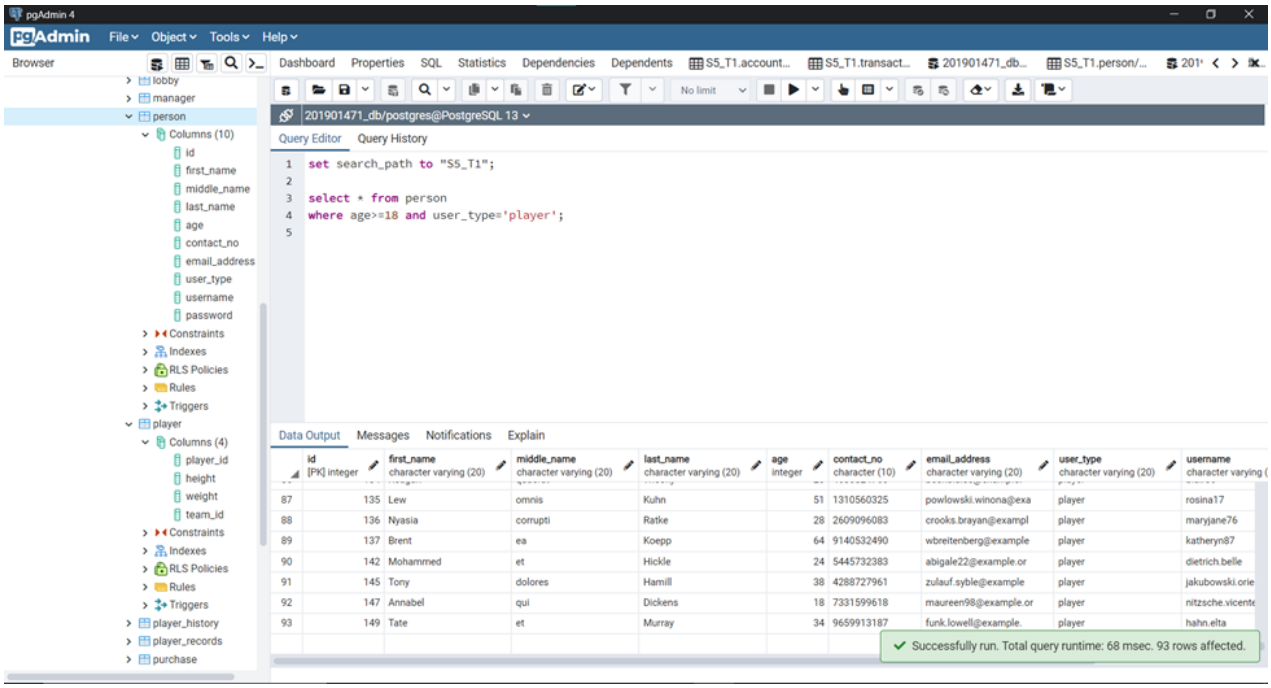
```
select player_id,count(*)
    from game,(select * from player join person on player_id=id) as T1
    where T1.age>game.age and T1.height>game.height and T1.weight>game.weight
    group by player_id
    order by player_id asc;
```



- Number of Tuple = 92

4. List all players whose age is greater than 18.

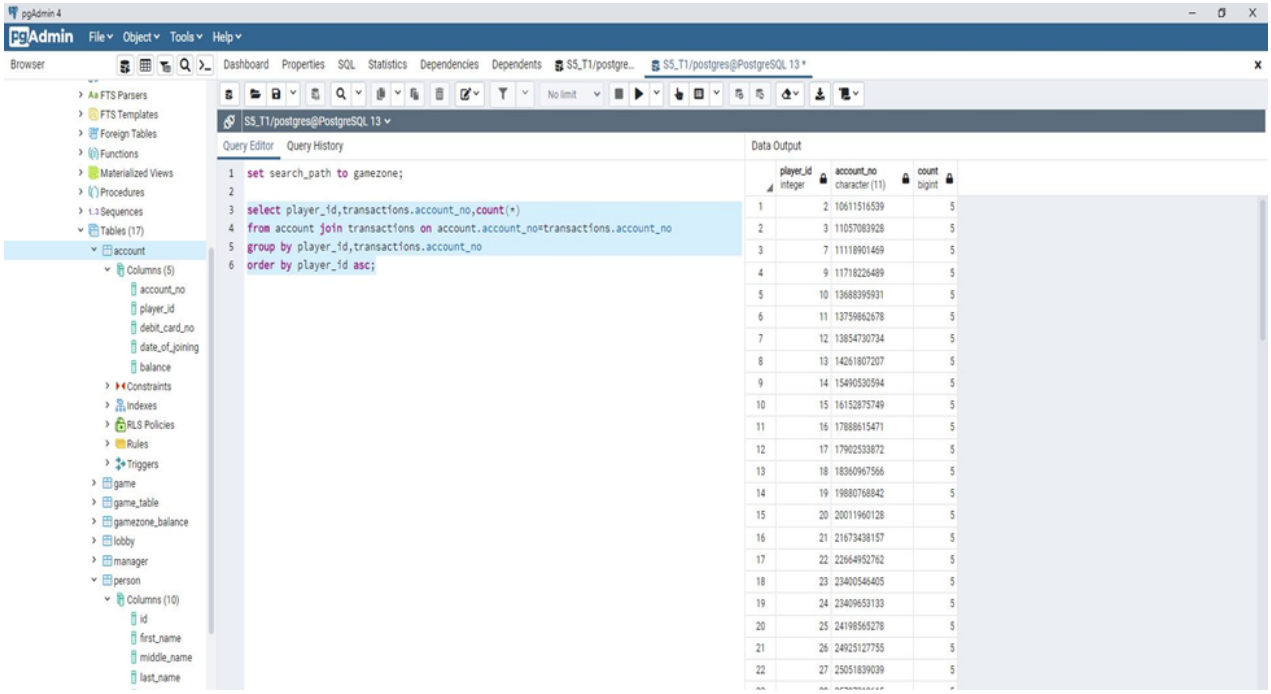
```
Select *
from person
Where age>=18 and user_type =  player ;
```



- Number of Tuple = 93

5. Count the number of transactions done by each player.

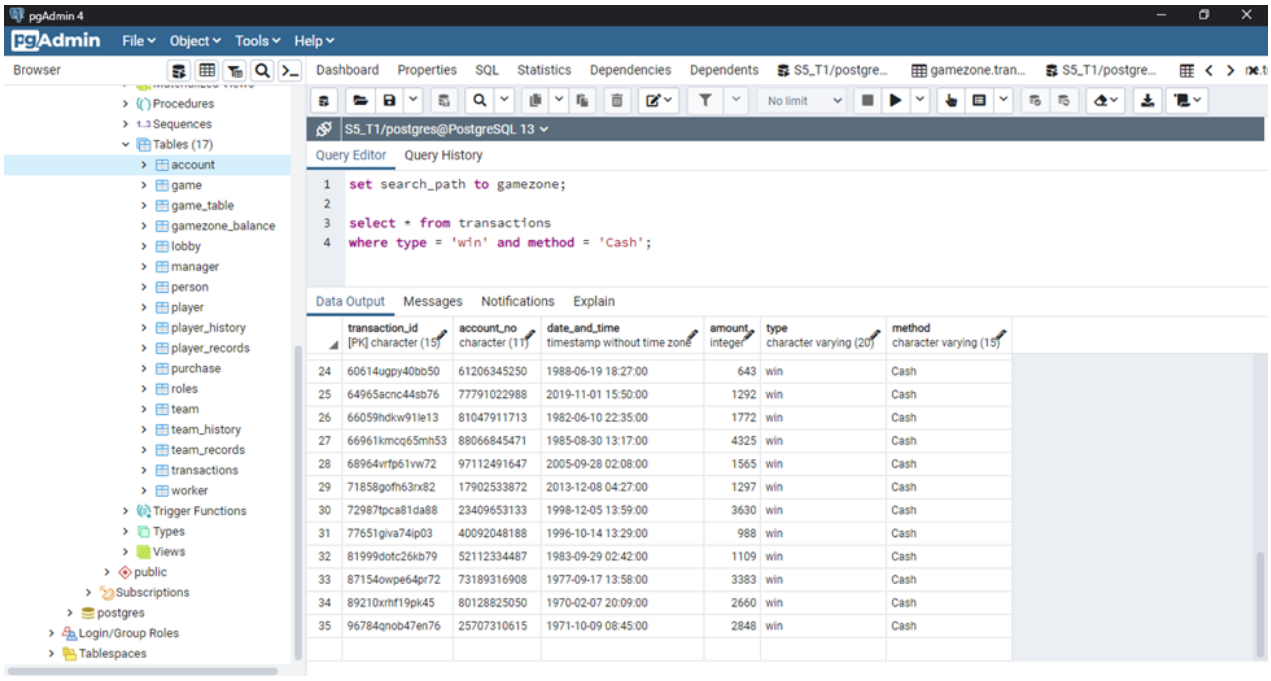
```
select player_id,transactions.account_no, count(*)
from account join transactions on
account.account_no=transactions.account_no
group by player_id, transactions.account_no
order by player_id asc;
```



- Number of Tuple = 114

6. List all the transactions in which the player gets rewarded the amount in Cash.

```
select *
from transactions
where type = 'win' and method ='Cash';
```

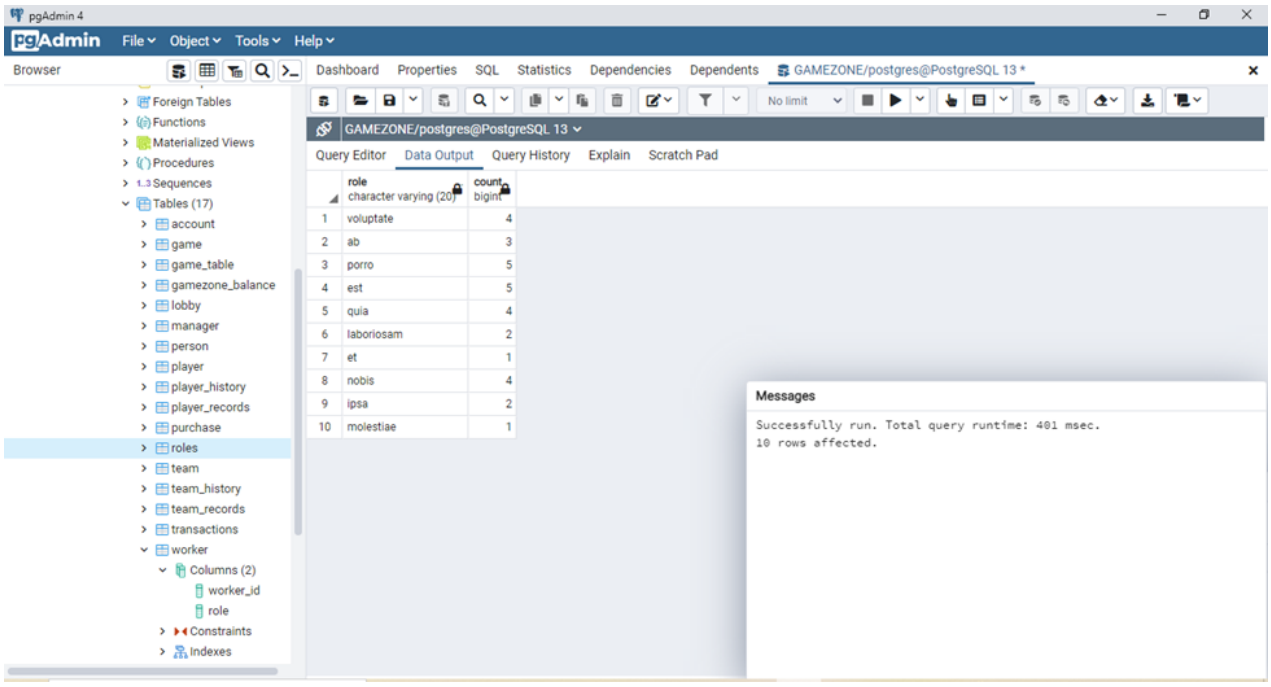


	transaction_id [PK] character (15)	account_no character (11)	date_and_time timestamp without time zone	amount integer	type character varying (20)	method character varying (15)
24	60614ugpy40bb50	61206345250	1988-06-19 18:27:00	643	win	Cash
25	64965acnc44sb76	77791022988	2019-11-01 15:50:00	1292	win	Cash
26	66059hdkw91le13	81047911713	1982-06-10 22:35:00	1772	win	Cash
27	66961kmcq65mh53	88066845471	1985-08-30 13:17:00	4325	win	Cash
28	68964vrfp61vw72	97112491647	2005-09-28 02:08:00	1565	win	Cash
29	71858gofh63rx82	17902533872	2013-12-08 04:27:00	1297	win	Cash
30	72987tpca81da88	23409653133	1998-12-05 13:59:00	3630	win	Cash
31	77651giva74ip03	40092048188	1996-10-14 13:29:00	988	win	Cash
32	81999dotc26kb79	52112334487	1983-09-29 02:42:00	1109	win	Cash
33	87154owpe64pr72	73189316908	1977-09-17 13:58:00	3383	win	Cash
34	89210xrhf19pk45	80128825050	1970-02-07 20:09:00	2660	win	Cash
35	96784qnob47en76	25707310615	1971-10-09 08:45:00	2848	win	Cash

- Number of Tuple = 35

7. Count the number of workers for each role.

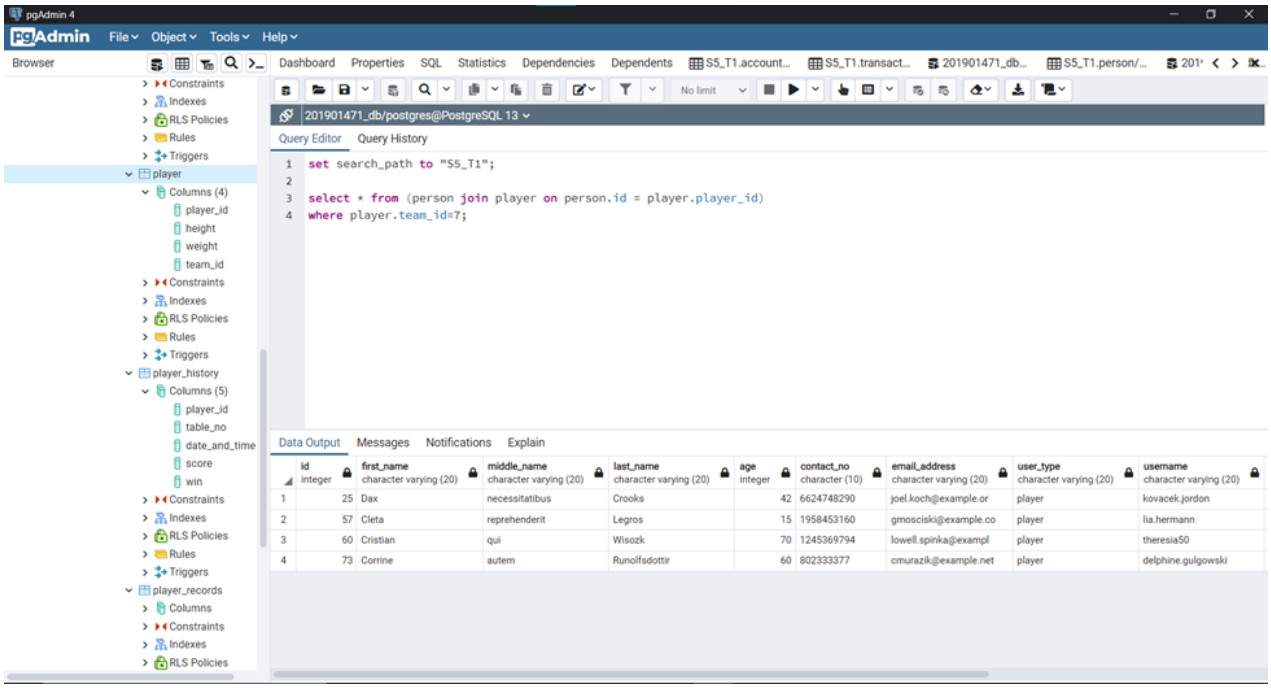
```
SELECT role, COUNT(*)
FROM worker
GROUP BY role;
```



- Number of Tuple = 10

8. Find details of players who have team_id 7.

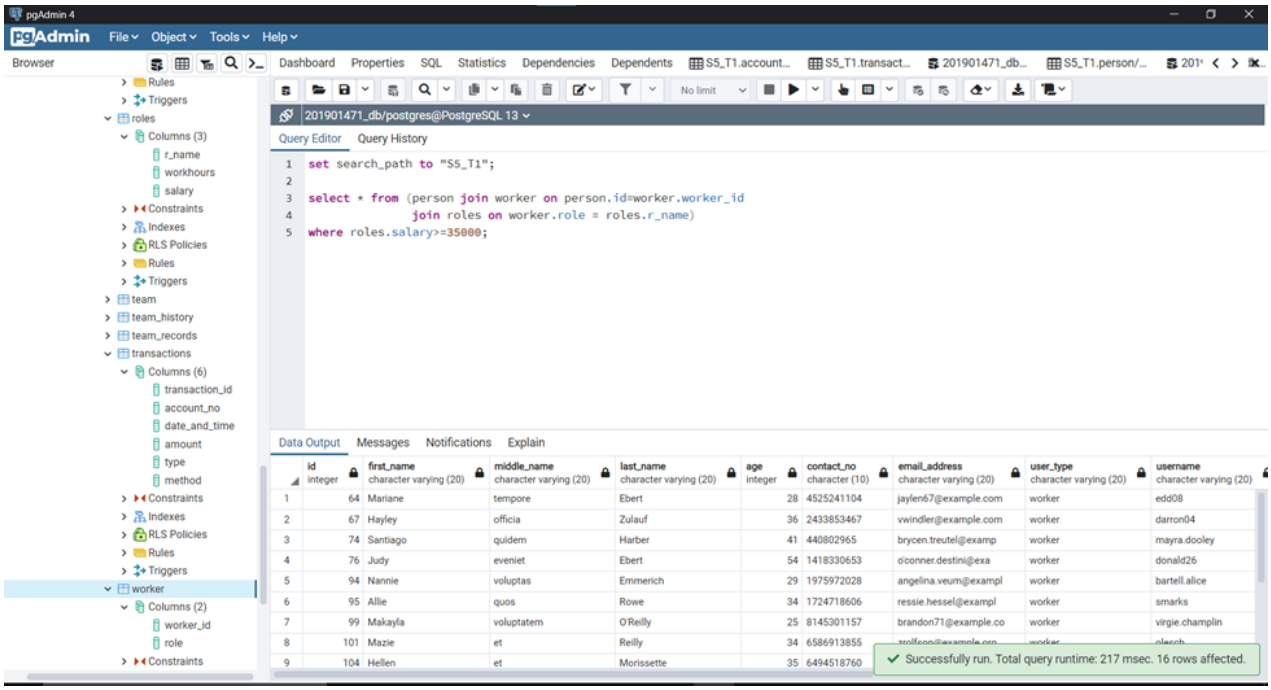
```
SELECT *
FROM (person JOIN player ON person.id=player.player_id)
WHERE player.team_id = 7
```



- Number of Tuple = 4

9. Find details of workers who have more than 35000 salary.

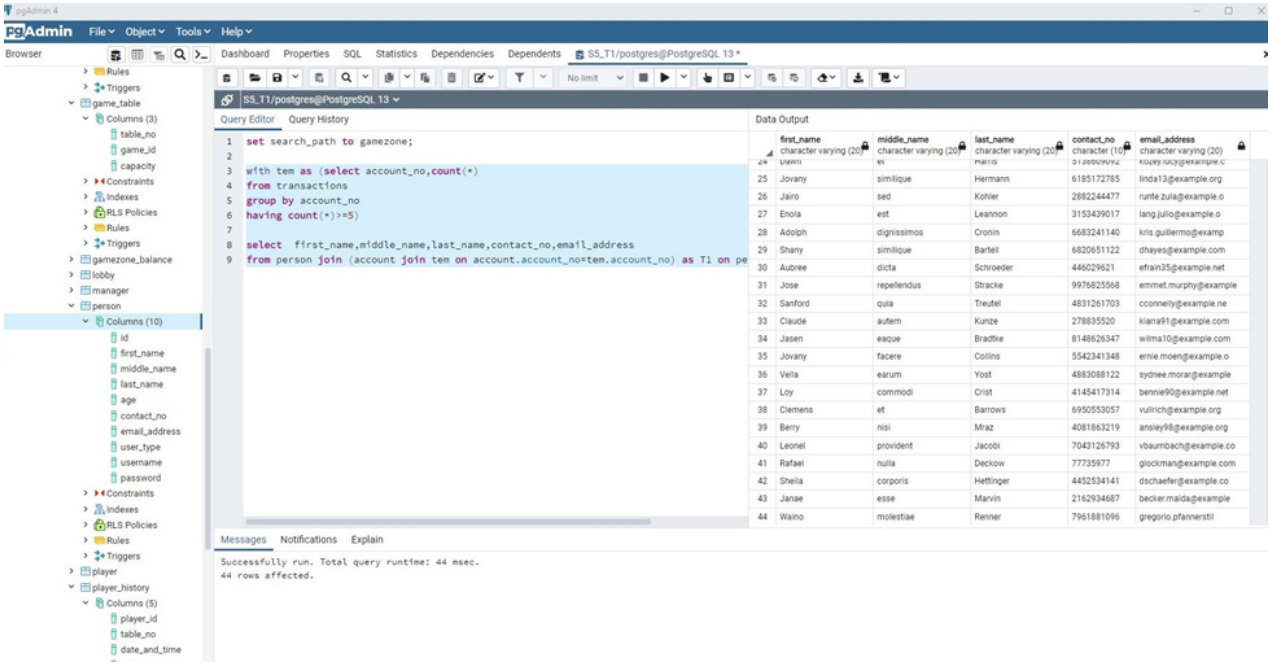
```
select * from (person join worker on person.id=worker.worker_id join roles
    on worker.role = roles.r_name)
    where roles.salary <= 35000;
```



- Number of Tuple = 16

10. Find the person details who have more than 3 transactions.

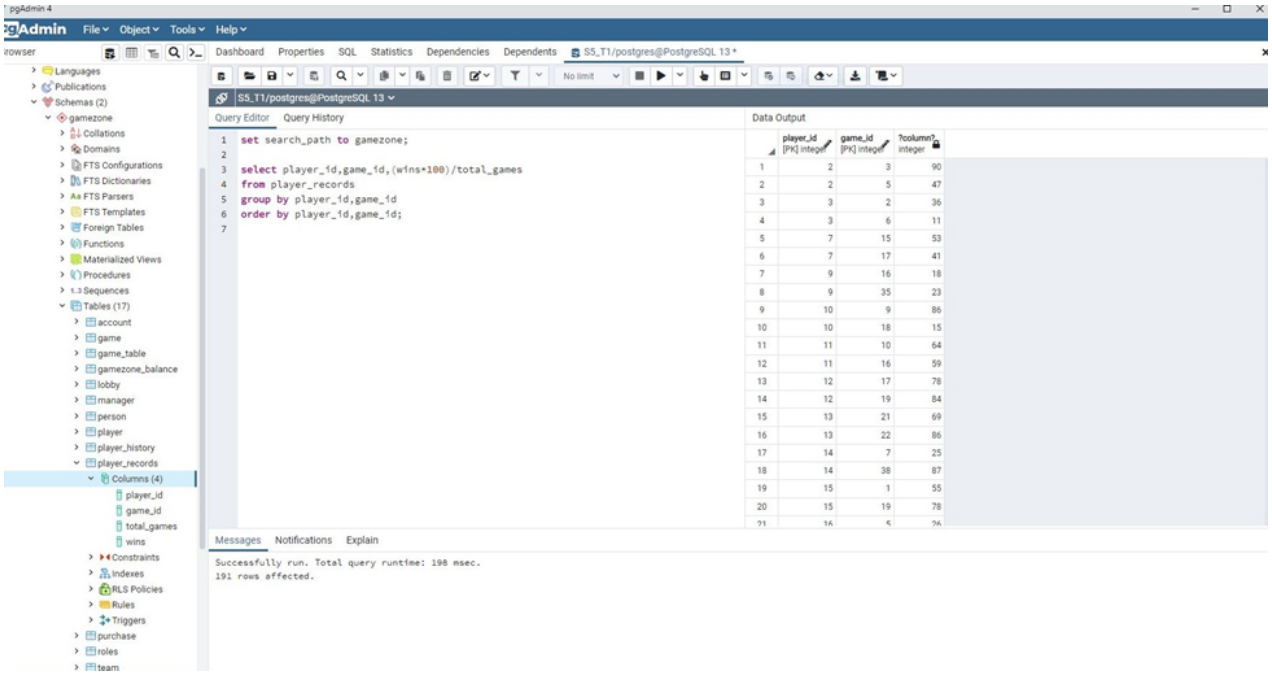
```
with tem as (select account_no,count(*) from transactions group by
    account_no having count(*) <=5)

select first_name,middle_name,last_name,contact_no,email_address from
    person join (account join tem on account.account_no=tem.account_no) as
    T1 on person.id=T1.player_id;
```



- Number of Tuple = 44

11. Find the win rate of each player game wise.

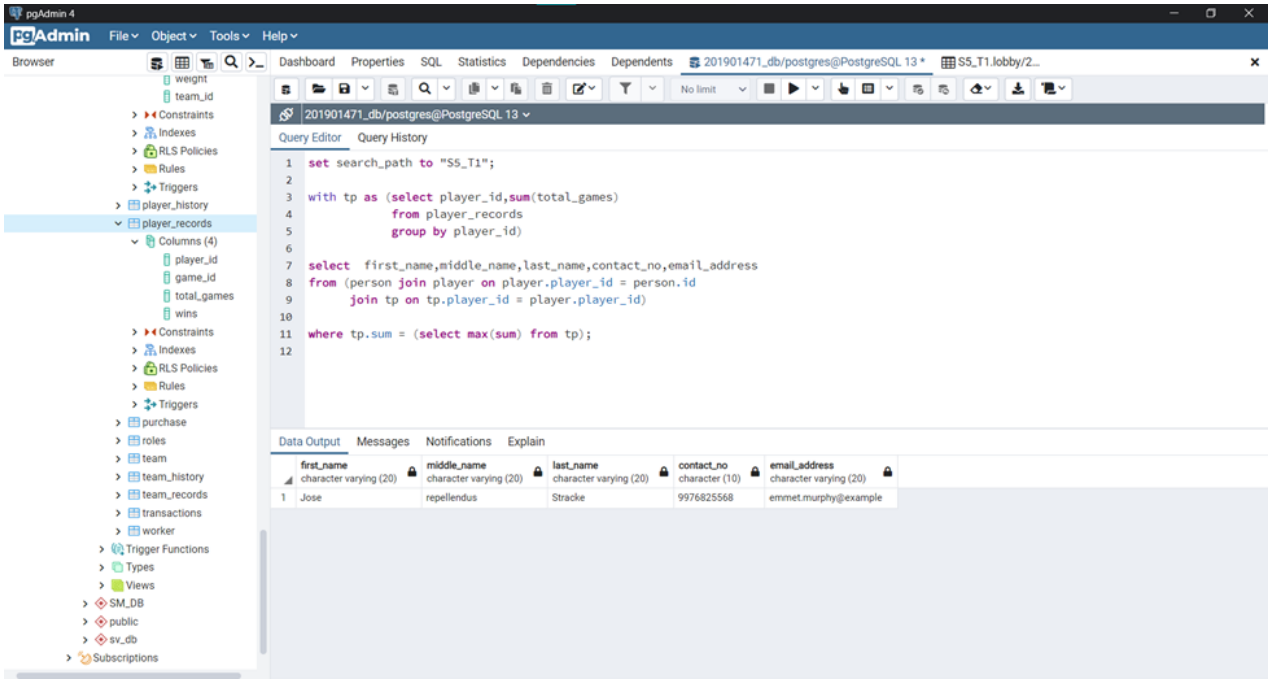
```
select player_id,game_id,(wins*100)/total_games from player_records group by
    player_id,game_id
order by player_id,game_id;
```



- Number of Tuple = 191

12. Find the player detail who played most games.

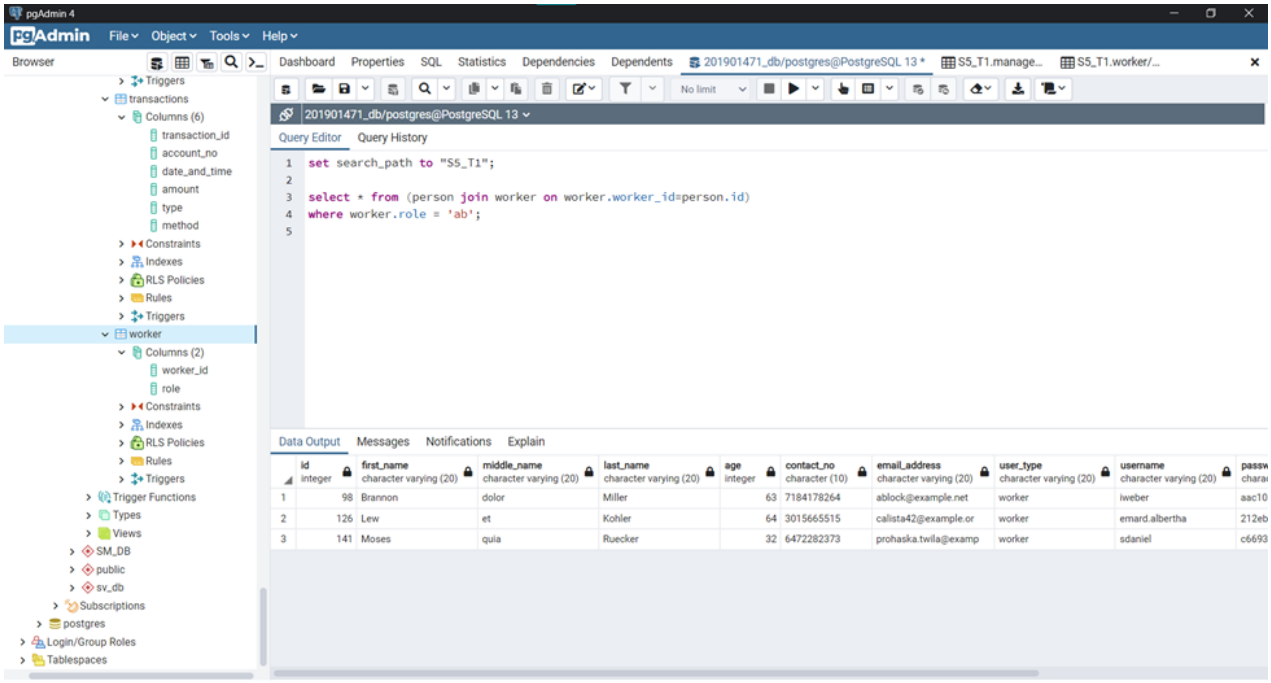
```
with tp as (select player_id,sum(total_games) from player_records
group by player_id)

select first_name,middle_name,last_name,contact_no,email_address from
    (person join player on player.player_id = person.id join tp on
    tp.player_id = player.player_id)
    where tp.sum = (select max(sum) from tp);
```



- Number of Tuple = 1

13. Find the Worker list who has the role “ab”.

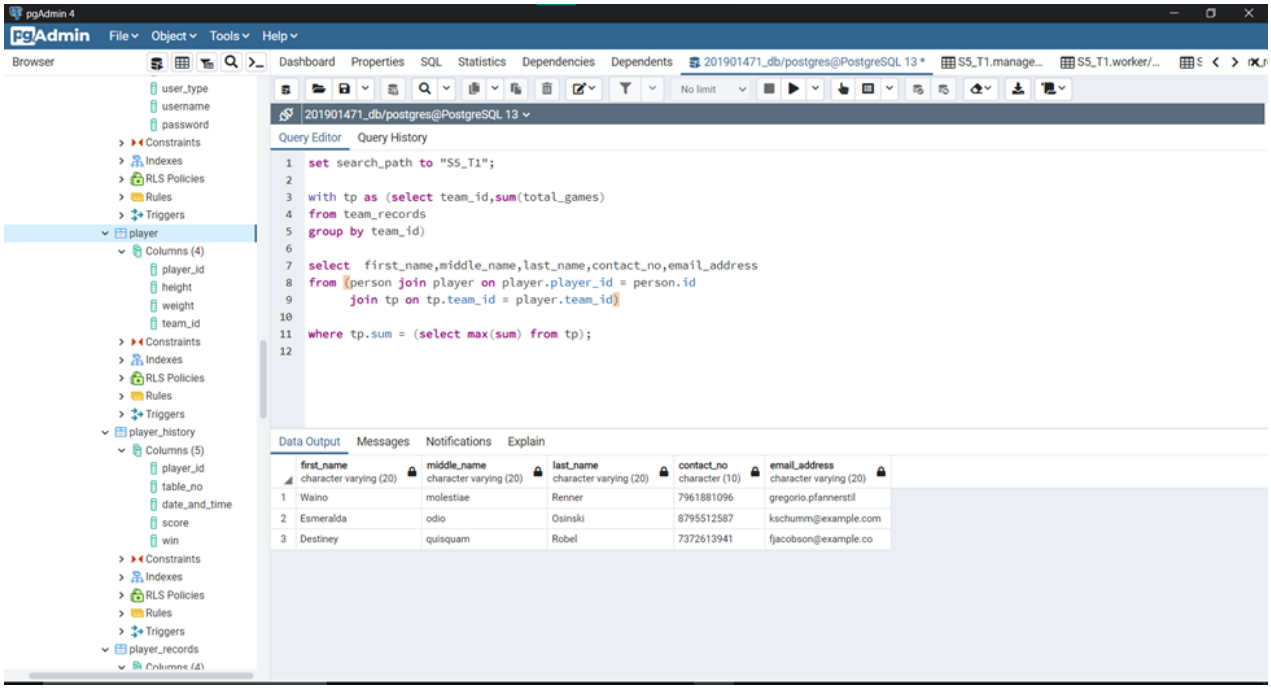
```
select * from (person join worker on worker.worker\_id=person.id)
where worker.role = 'ab';
```



- Number of Tuple = 3

14. List the team players who played most games in teams.

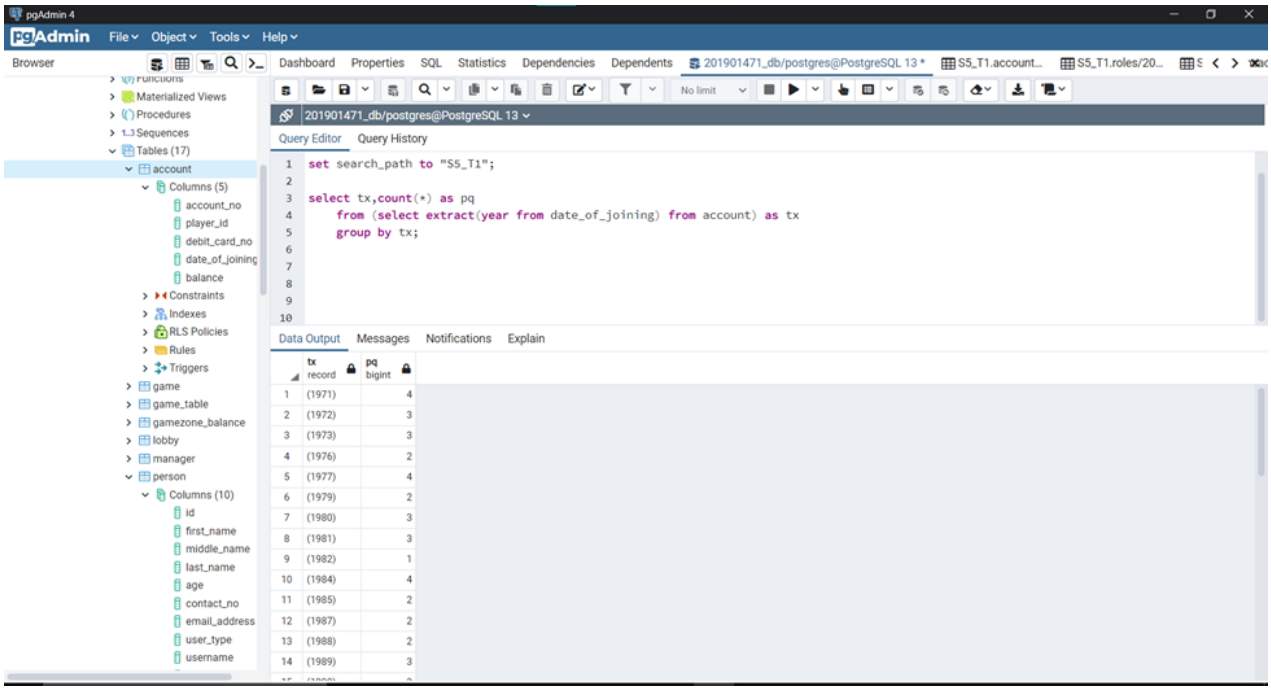
```
with tp as (select team_id,sum(total_games) from team_records group by
    team_id)

select first_name,middle_name,last_name,contact_no,email_address from
    (person join player on player.player_id = person.id join tp on
    tp.team_id = player.team_id)
        where tp.sum = (select max(sum) from tp);
```



- Number of Tuple = 3

15. Show data how many players joined each year.

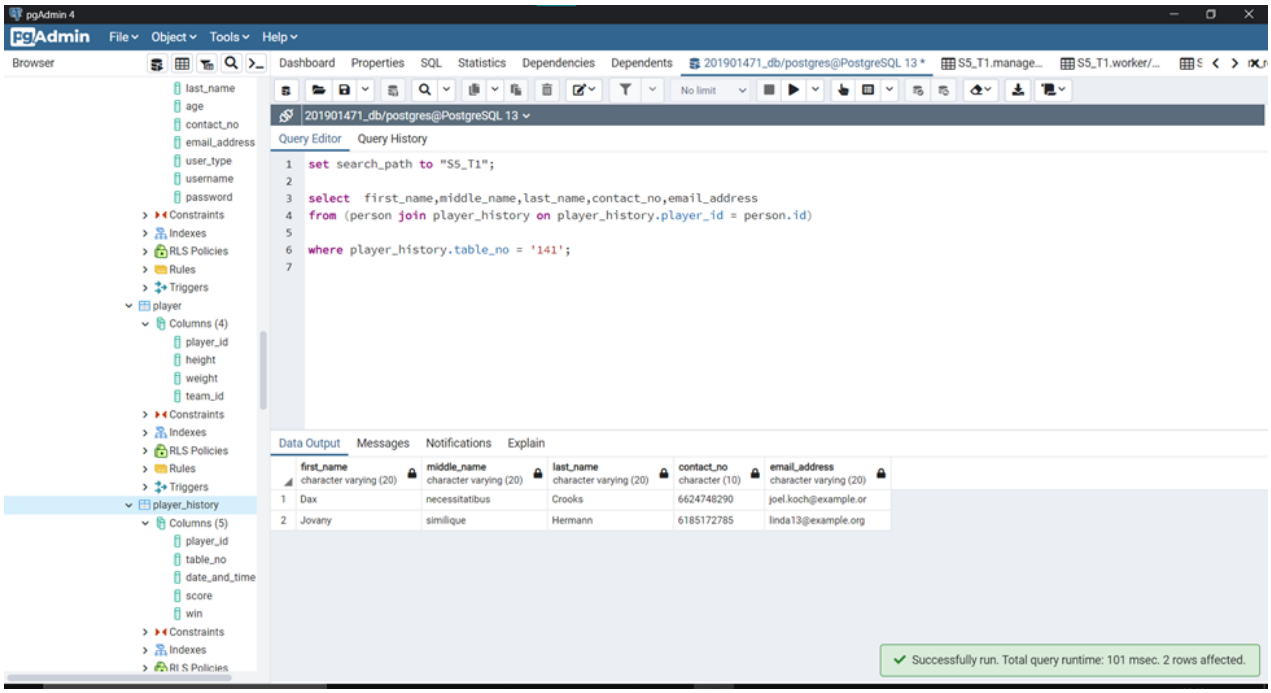
```
select tx,count(*) as pq from (select extract(year from date\_of\_joining)
   from account) as tx
group by tx;
```



- Number of Tuple = 45

16. List all players who played on table number 141.

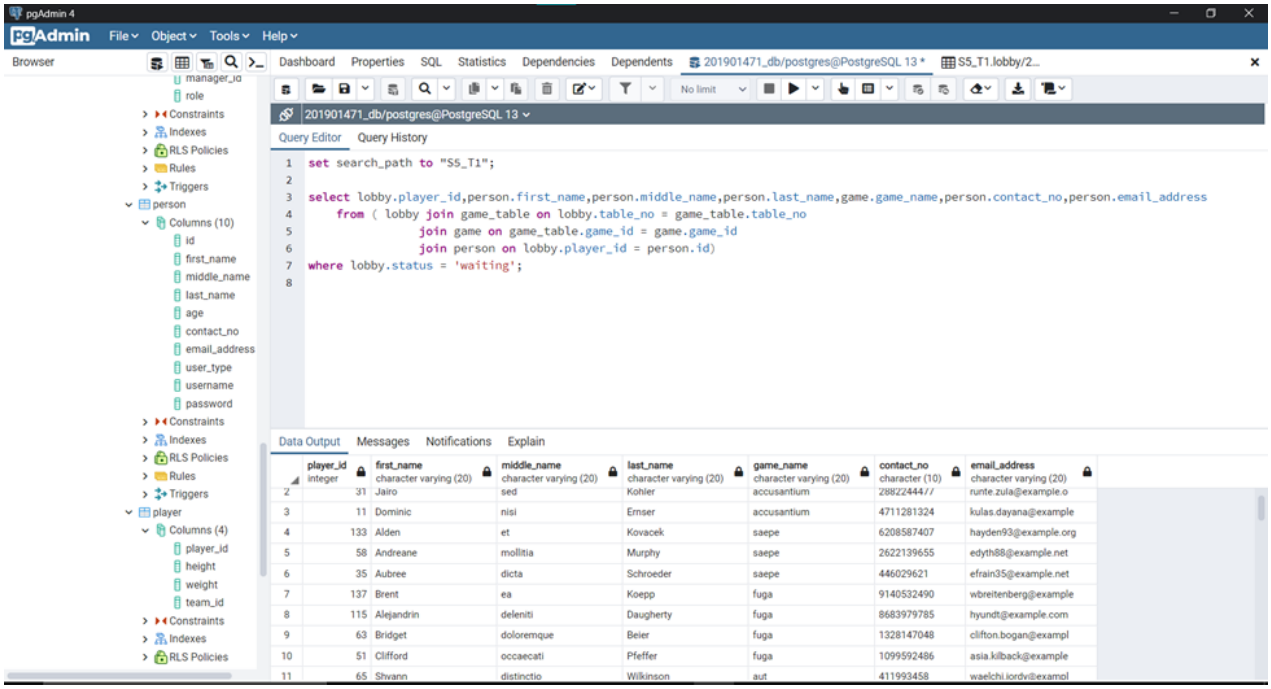
```
select first_name,middle_name,last_name,contact_no,email_address
from (person join player_history on player_history.player_id = person.id)
where player\_history.table\_no = '141';
```



- Number of Tuple = 2

17. Find the list of players who are waiting in the lobby and also show the game name for which they are waiting.

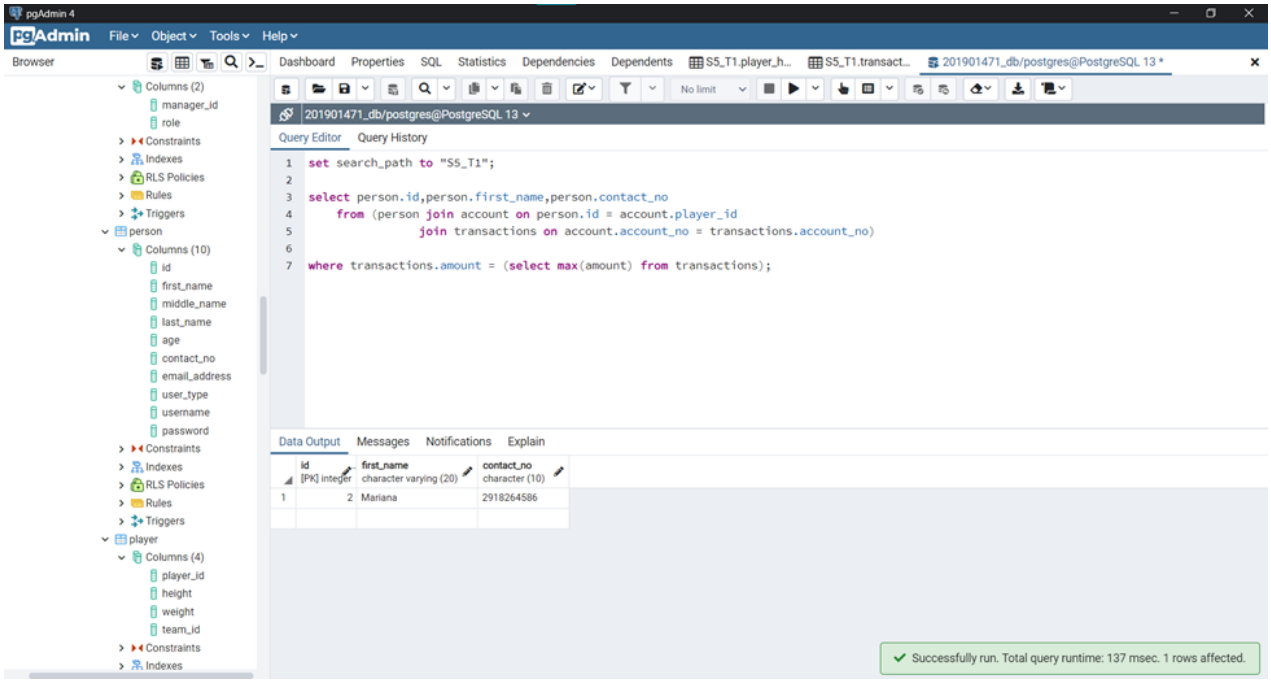
```
select lobby.player_id,person.first_name,person.middle_name,
person.last_name,game.game_name,person.contact_no, person.email_address
from ( lobby join game_table on lobby.table_no = game_table.table_no
join game on game_table.game_id = game.game_id
join person on lobby.player_id = person.id)
where lobby.status = 'waiting';
```



- Number of Tuple = 57

18. Find the person id, contact number and name who has the biggest transaction amount.

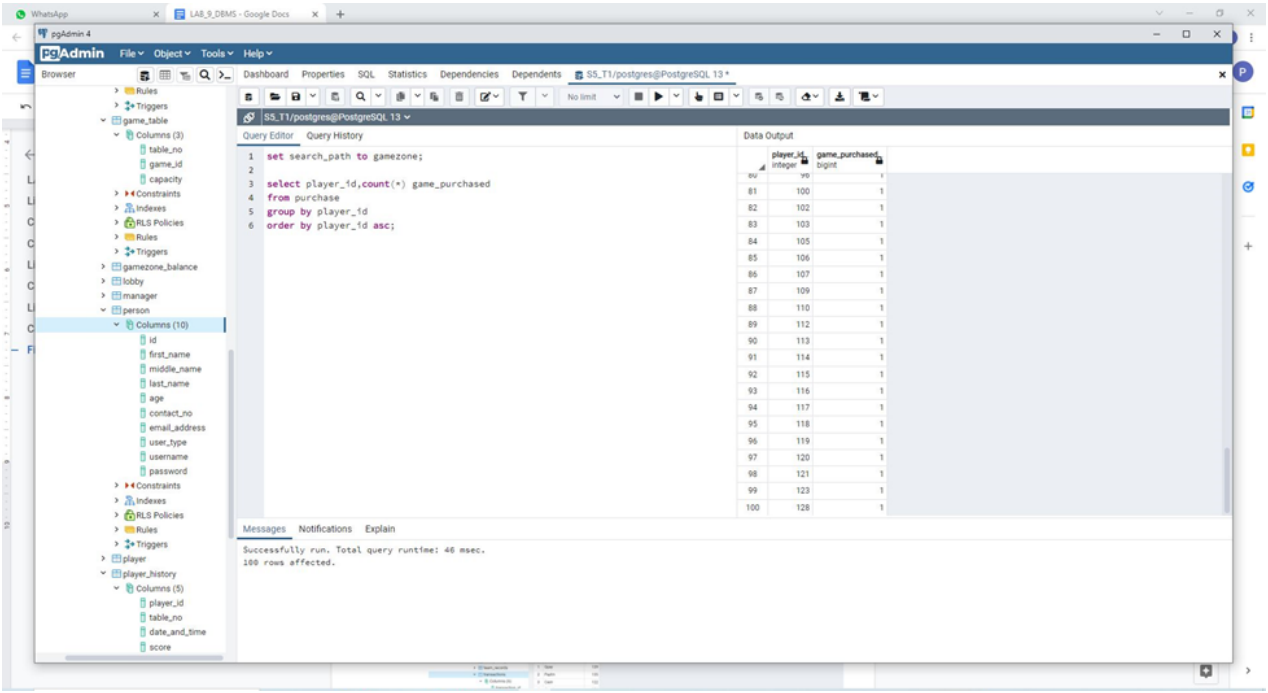
```
select person.id,person.first_name,person.contact_no
from (person join account on person.id = account.player_id
join transactions on account.account_no = transactions.account_no)
where transactions.amount = (select max(amount) from transactions);
```



- Number of Tuple = 1

19. Number of game purchased by each player.

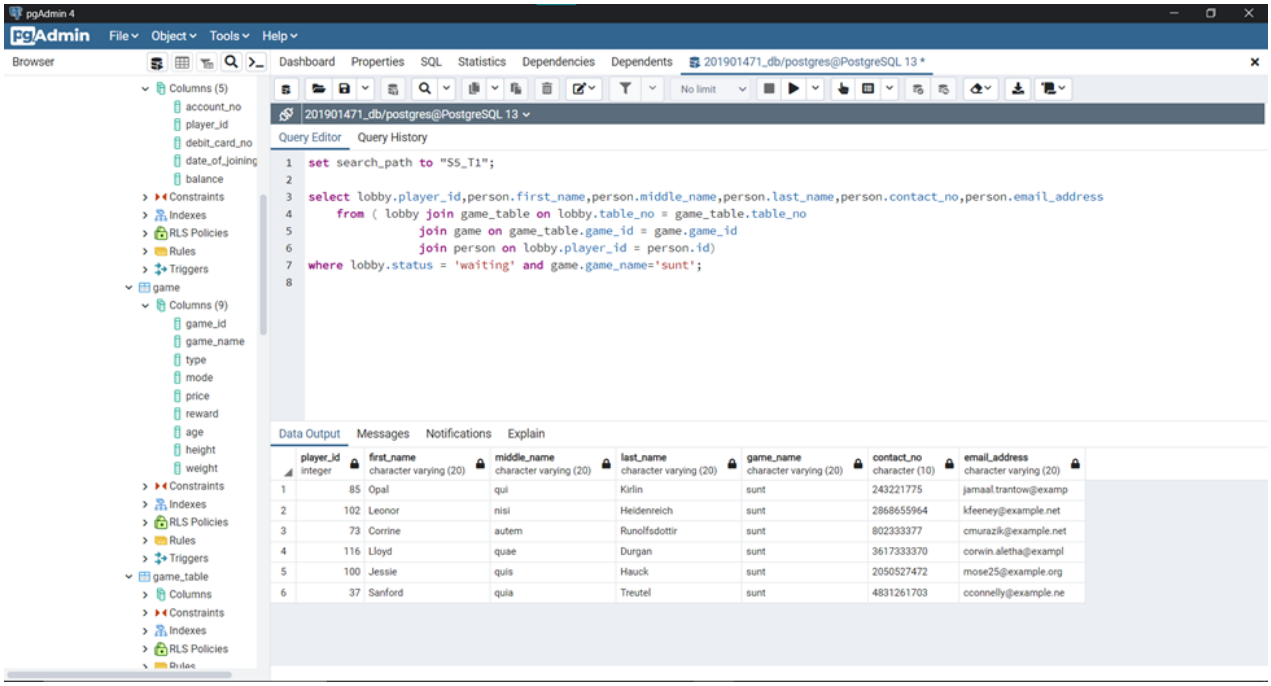
```
select player_id,count(*) game_purchased from purchase
group by player_id
order by player_id asc;
```



- Number of Tuple = 100

20. Find the player details who are waiting for the game “sunt”.

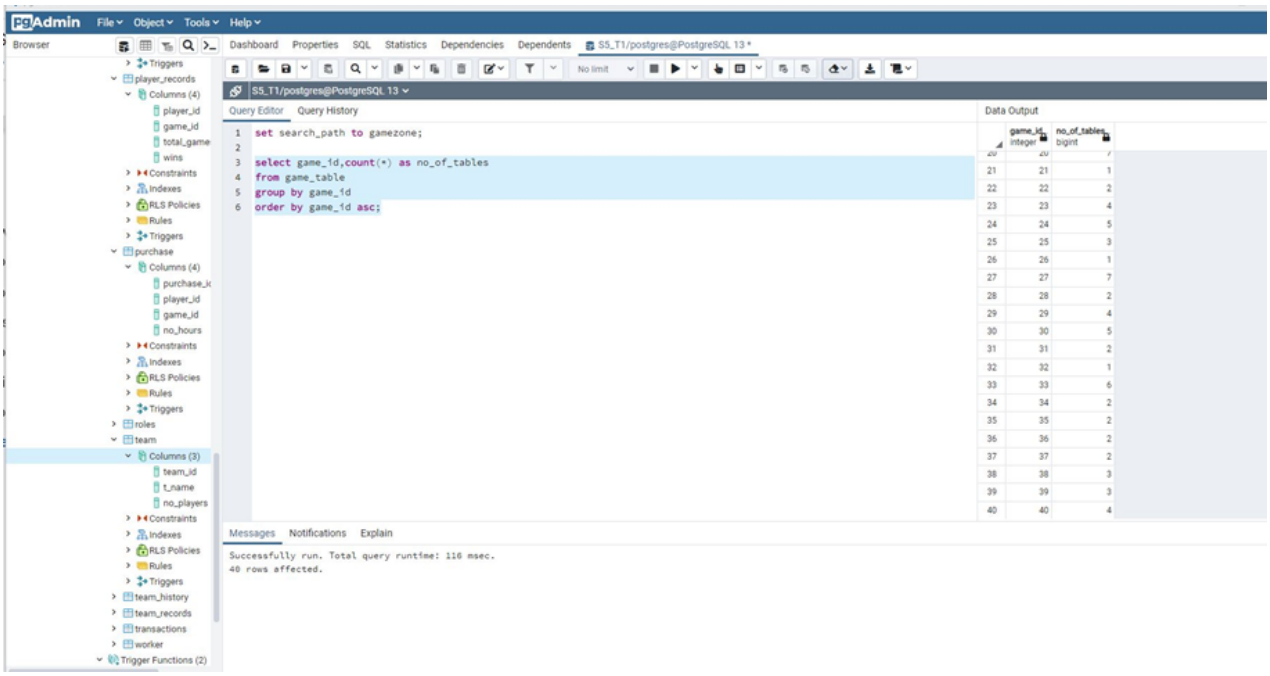
```
select lobby.player_id,person.first_name,person.middle_name,
person.last_name,person.contact_no,person.email_address
from ( lobby join game_table on lobby.table_no = game_table.table_no join
   game on game_table.game_id = game.game_id
join person on lobby.player_id = person.id)
where lobby.status = 'waiting' and game.game_name='sunt';
```



- Number of Tuple = 6

21. Find the number of tables for each game.

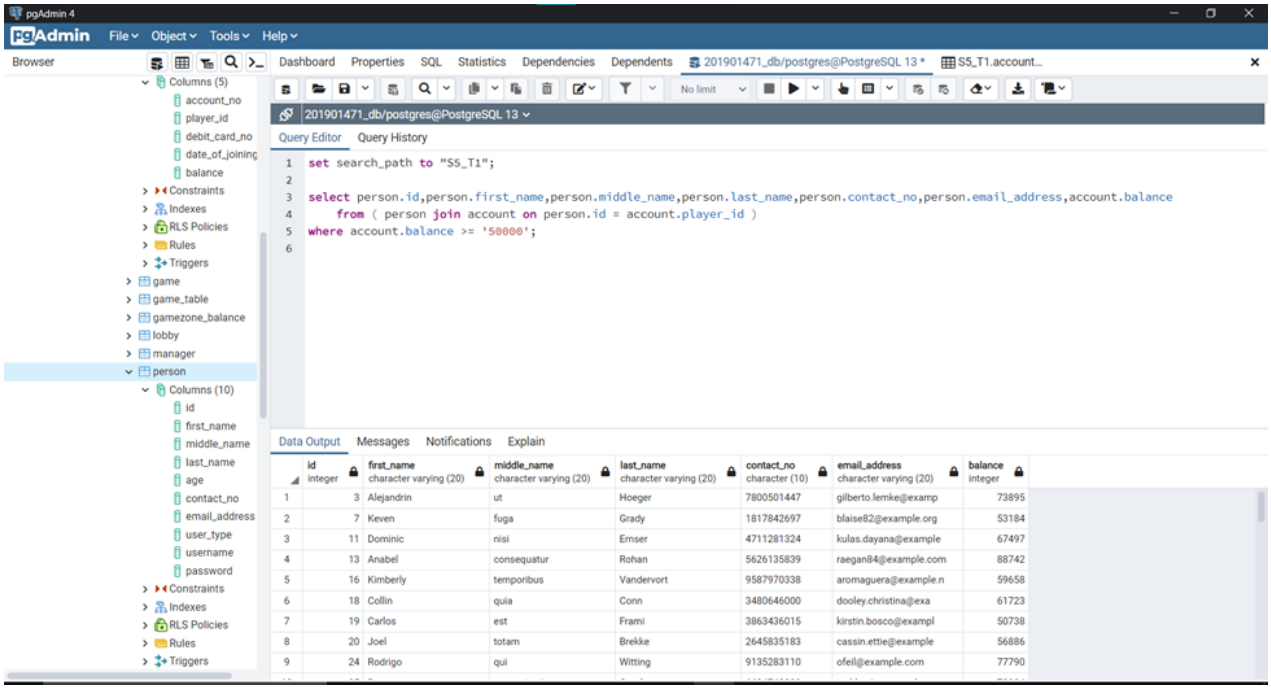
```
select game_id,count(*) as no_of_tables from game_table
group by game_id
order by game_id asc;
```



- Number of Tuple = 40

22. Find the player details who has balance more than “50000”.

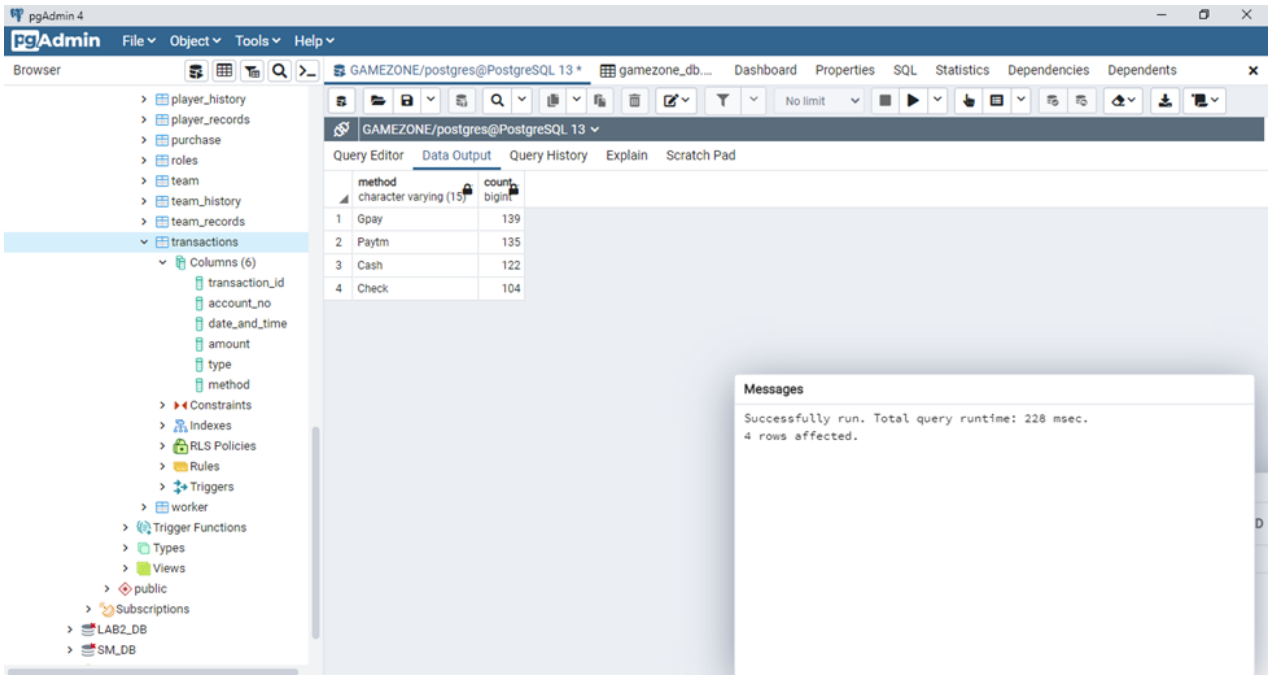
```
select person.id,person.first_name,person.middle_name,person.last_name,
person.contact_no,person.email_address,account.balance
from ( person join account on person.id = account.player_id )
where account.balance >= '50000';
```



- Number of Tuple = 49

23. Find the number of transactions done by each method.

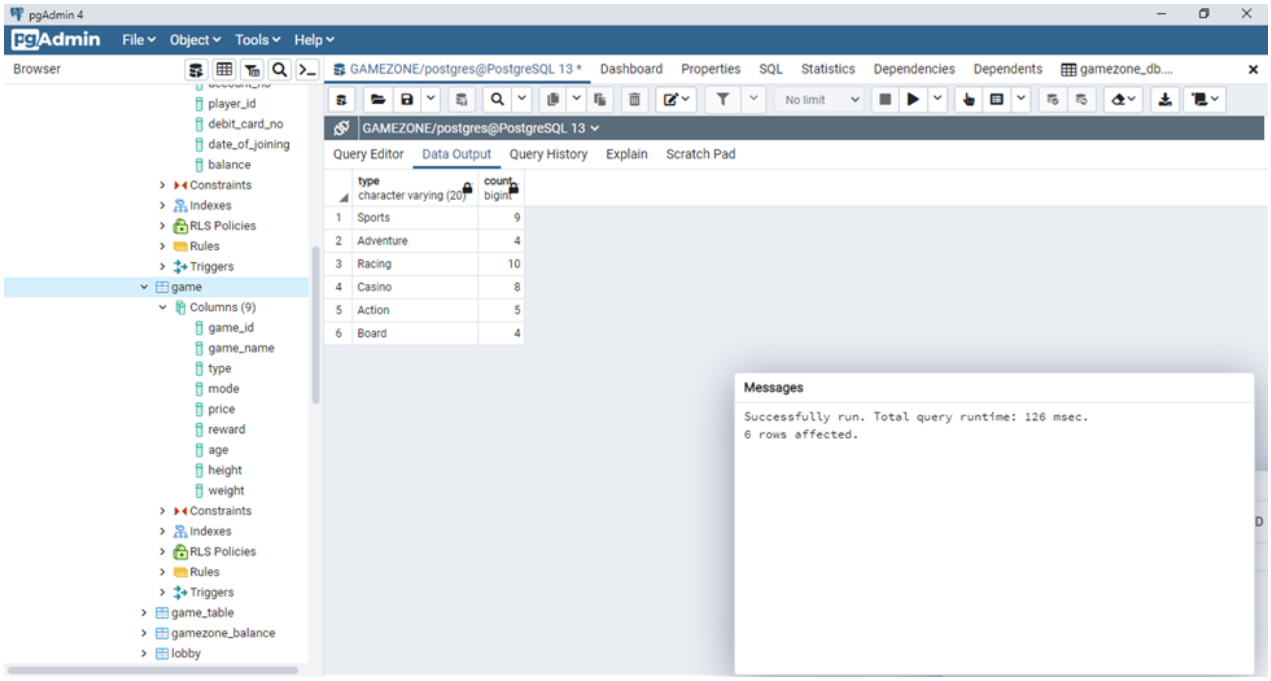
```
SELECT transactions.method, COUNT(*)
FROM transactions
GROUP BY transactions.method
```



- Number of Tuple = 4

24. Find the number of games for each type.

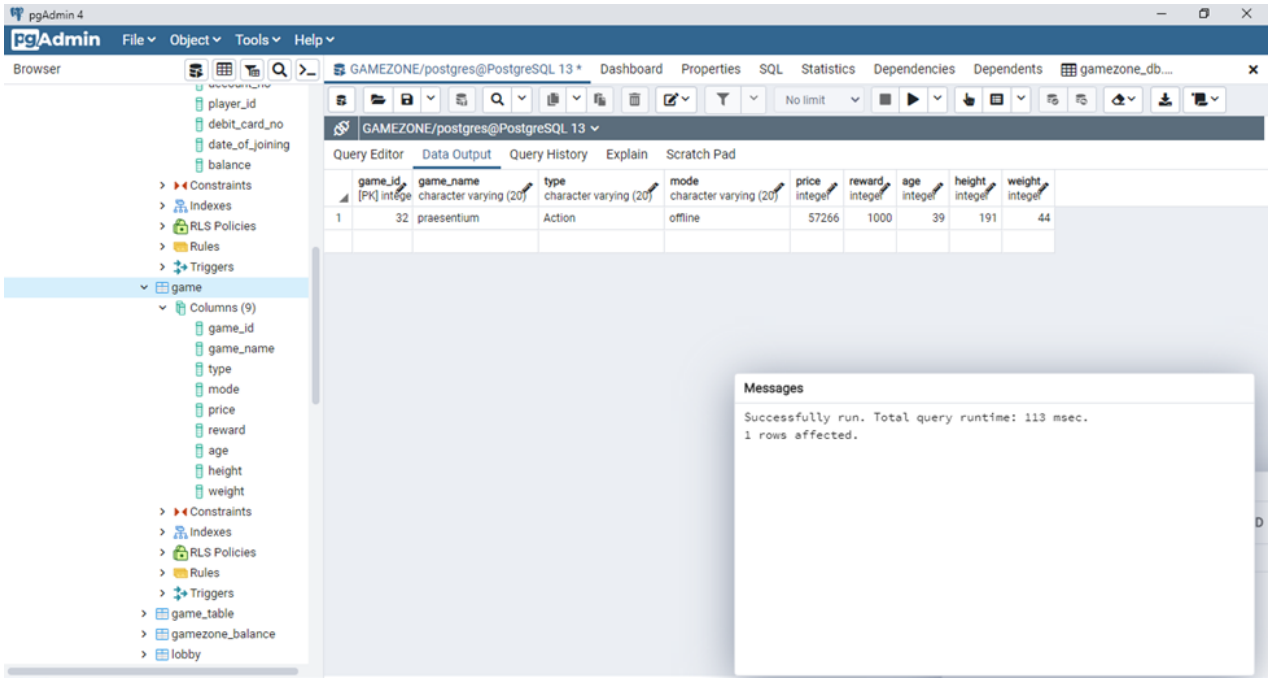
```
SELECT game.type, count(*)
FROM game GROUP BY game.type;
```



- Number of Tuple = 6

25. Find the game details which have the highest reward.

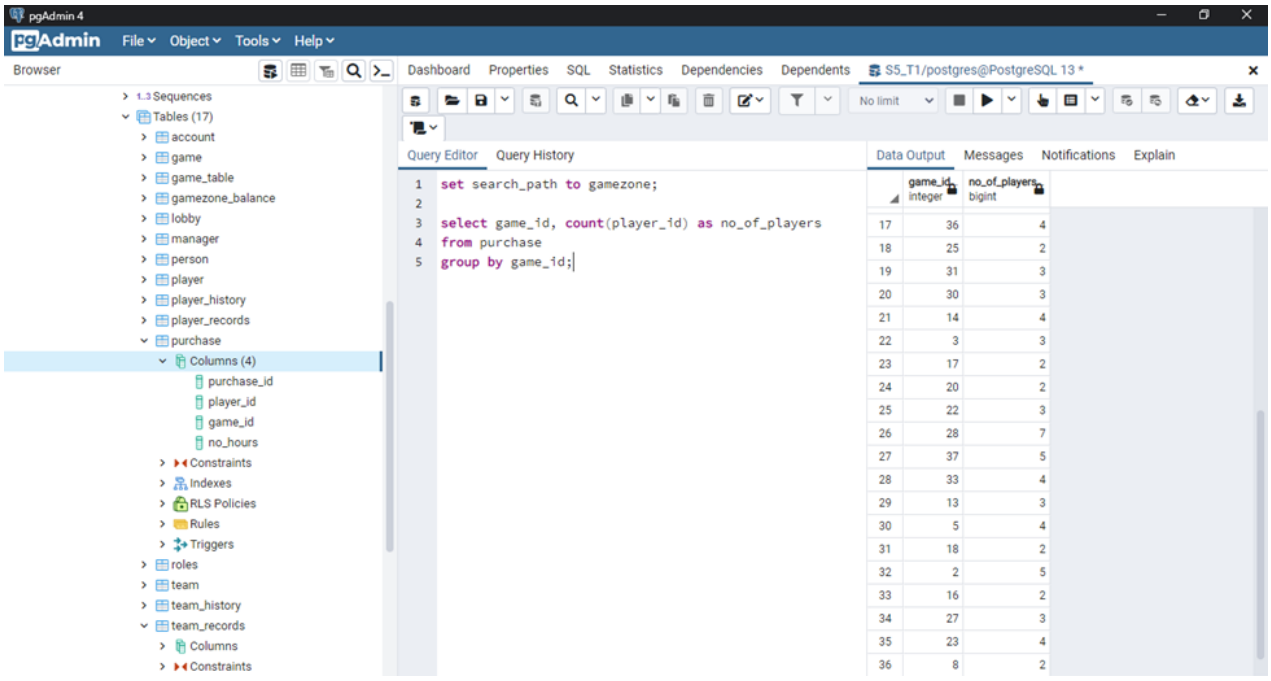
```
SELECT * FROM game
WHERE game.reward =
(SELECT MAX(game.reward)
FROM game)
```



- Number of Tuple = 1

26. Number of players purchase each game.

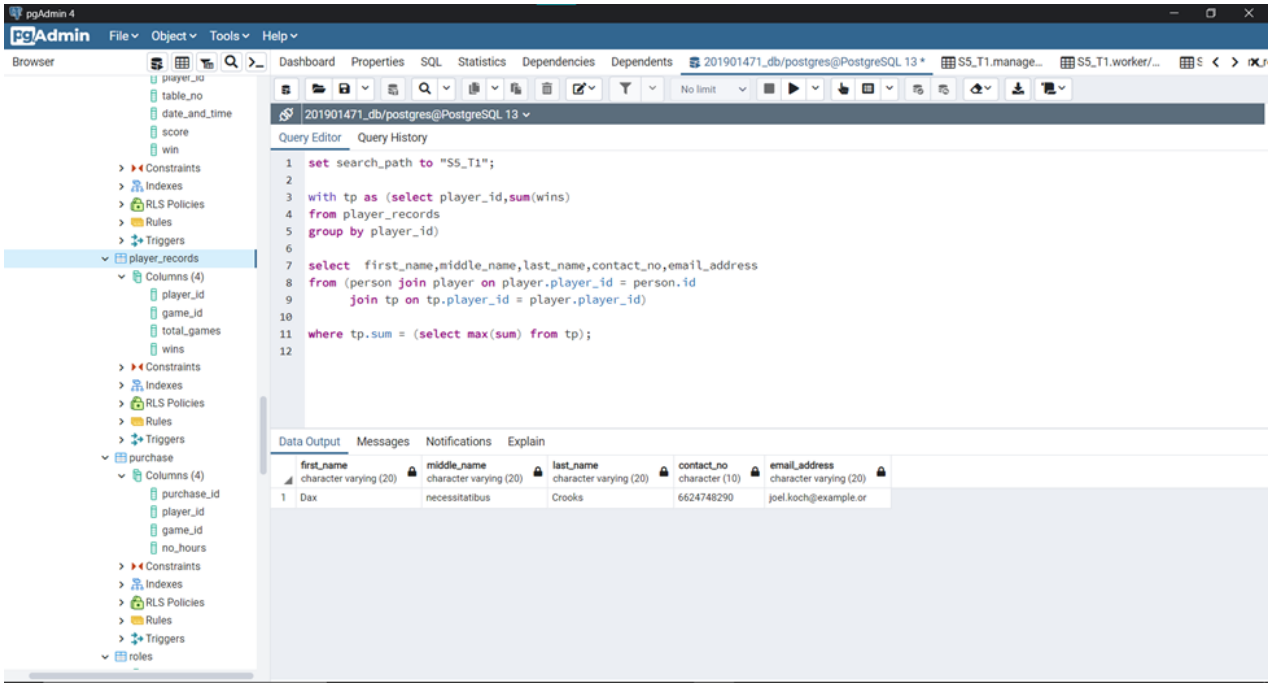
```
select game_id, count(player_id) as no_of_players
from purchase
group by game_id;
```



- Number of Tuple = 37

27. Find the player who wins most times.

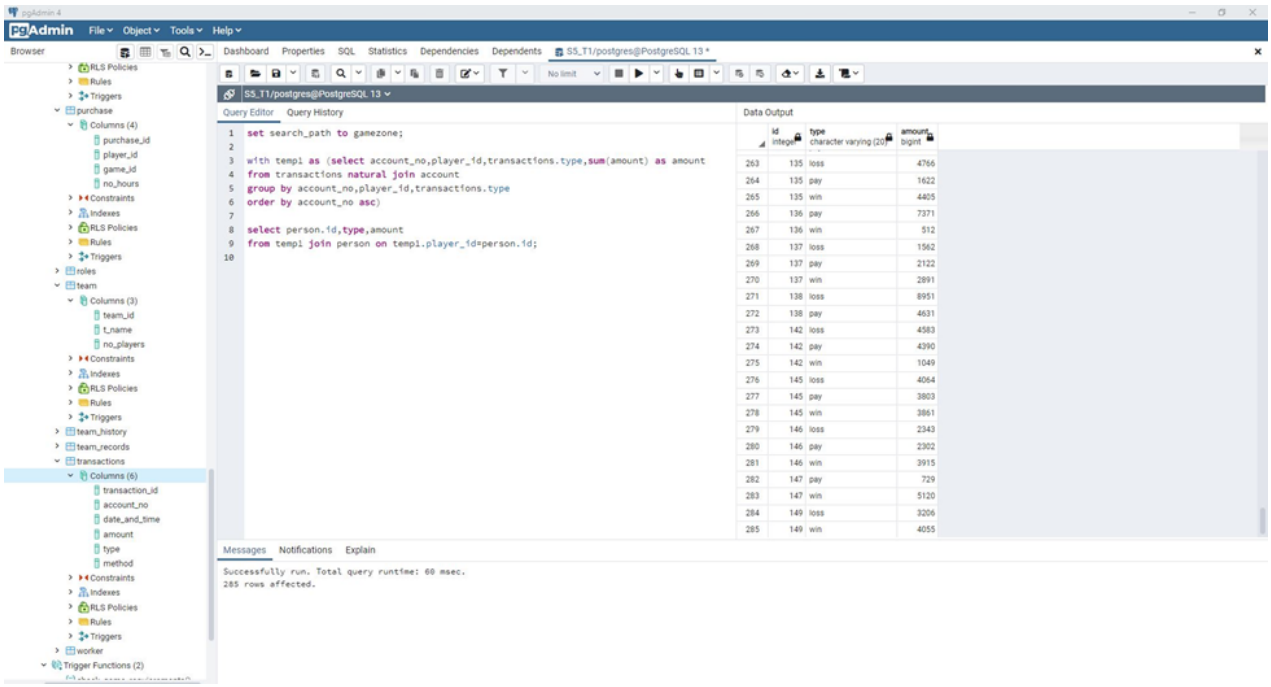
```
with tp as (select player_id,sum(wins)
from player_records
group by player_id)

select  first_name,middle_name,last_name,contact_no,email_address
from (person join player on player.player_id = person.id
     join tp on tp.player_id = player.player_id)

where tp.sum = (select max(sum) from tp);
```



- Number of Tuple = 1

28. List the total reward, loss and pay amount of each player.

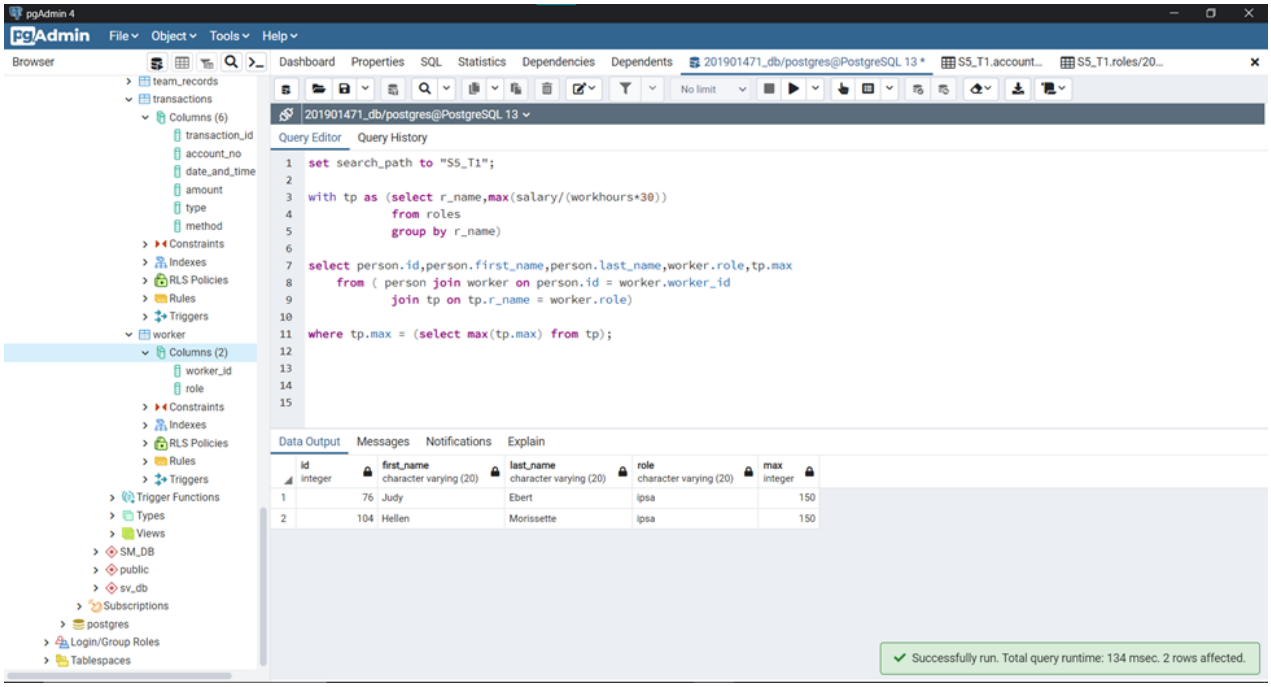
```
with temp1 as (select account_no,player_id,transactions.type,sum(amount) as
    amount
from transactions natural join account
group by account_no,player_id,transactions.type
order by account_no asc)

select person.id,type,amount
from temp1 join person on temp1.player_id=person.id;
```



- Number of Tuple = 285

29. Find the details of the worker who has the highest salary for 1 hour.

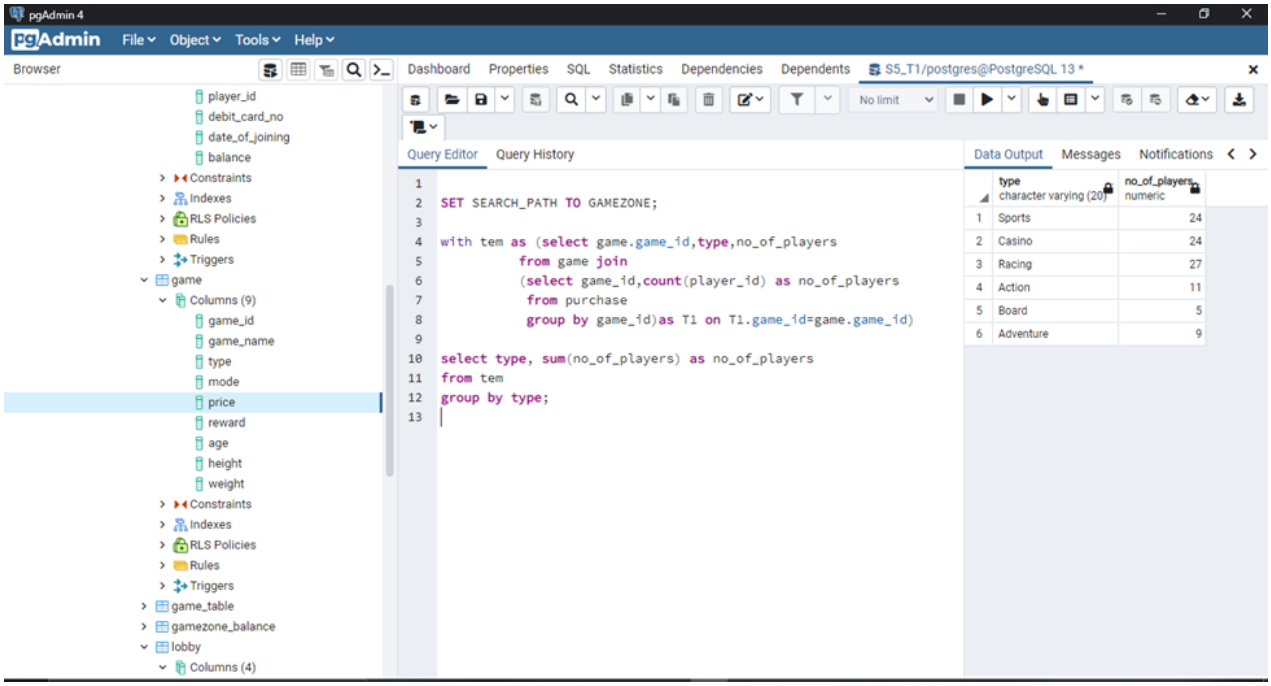
```
with tp as (select r_name,max(salary/(workhours*30))
            from roles
            group by r_name)

select person.id,person.first_name,person.last_name,worker.role,tp.max
   from ( person join worker on person.id = worker.worker_id
          join tp on tp.r_name = worker.role)

where tp.max = (select max(tp.max) from tp);
```



- Number of Tuple = 2

30. Which type of game is played most?

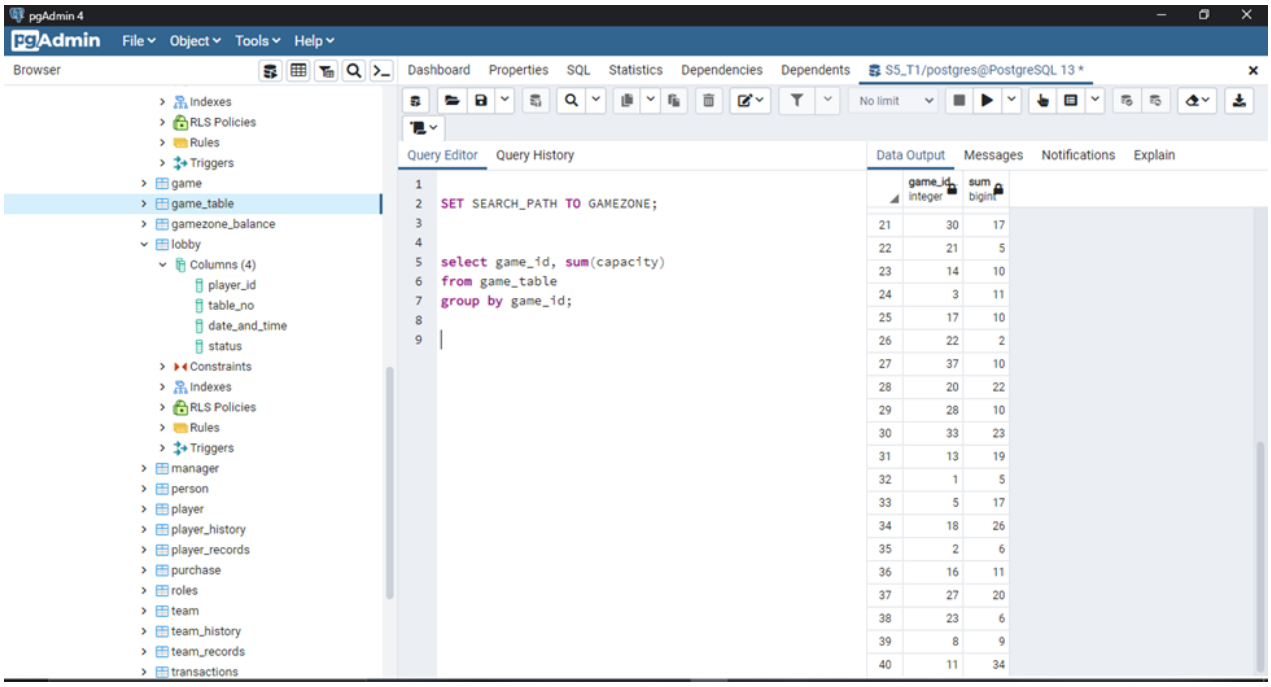
```
with tem as (select game.game_id,type,no_of_players
        from game join
        (select game_id,count(player_id) as no_of_players
         from purchase
         group by game_id)as T1 on T1.game_id=game.game_id)

select type, sum(no_of_players) as no_of_players
from tem
group by type;
```



- Number of Tuple = 6

31. Find the number of players can play particular game at same time.

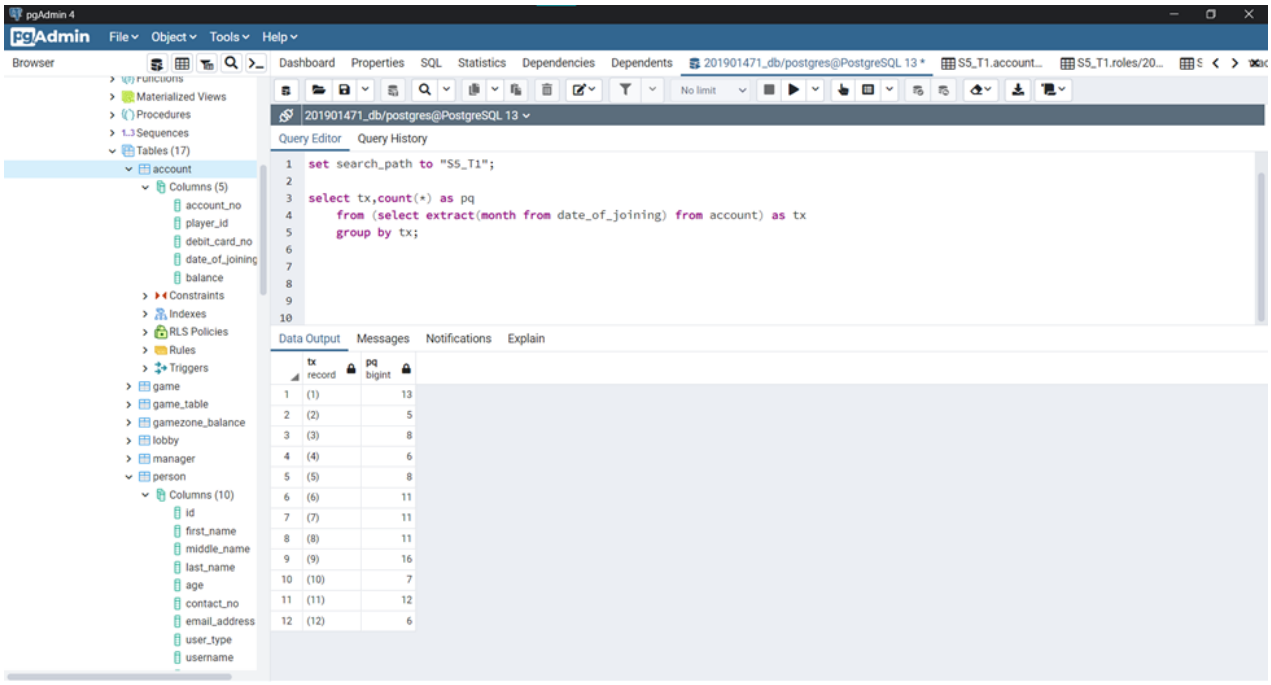
```
select game_id, sum(capacity)
from game_table
group by game_id;
```



- Number of Tuple = 40

32. Analyze the data and list the total number of players joined based on months.

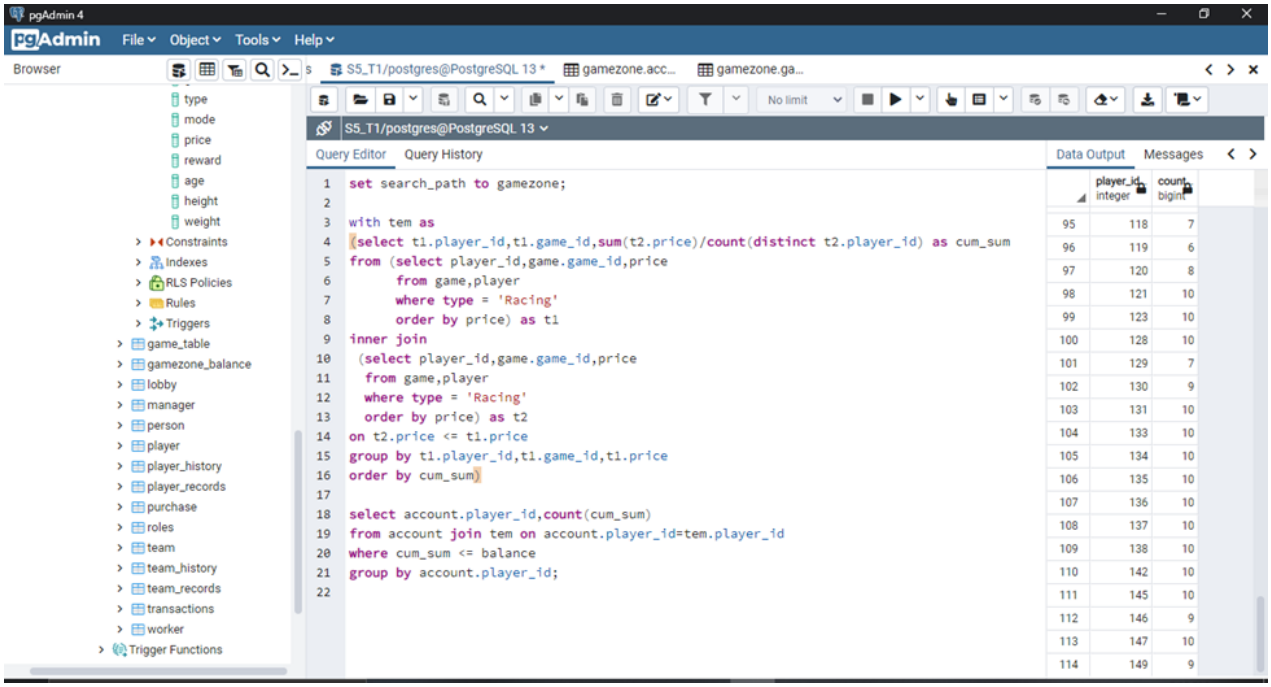
```
select tx, count(*) as pq
   from (select extract(month from date_of_joining) from account) as tx
   group by tx;
```



- Number of Tuple = 12

33. Count the number of different games of 'Racing' type a player can purchase from his/her balance.

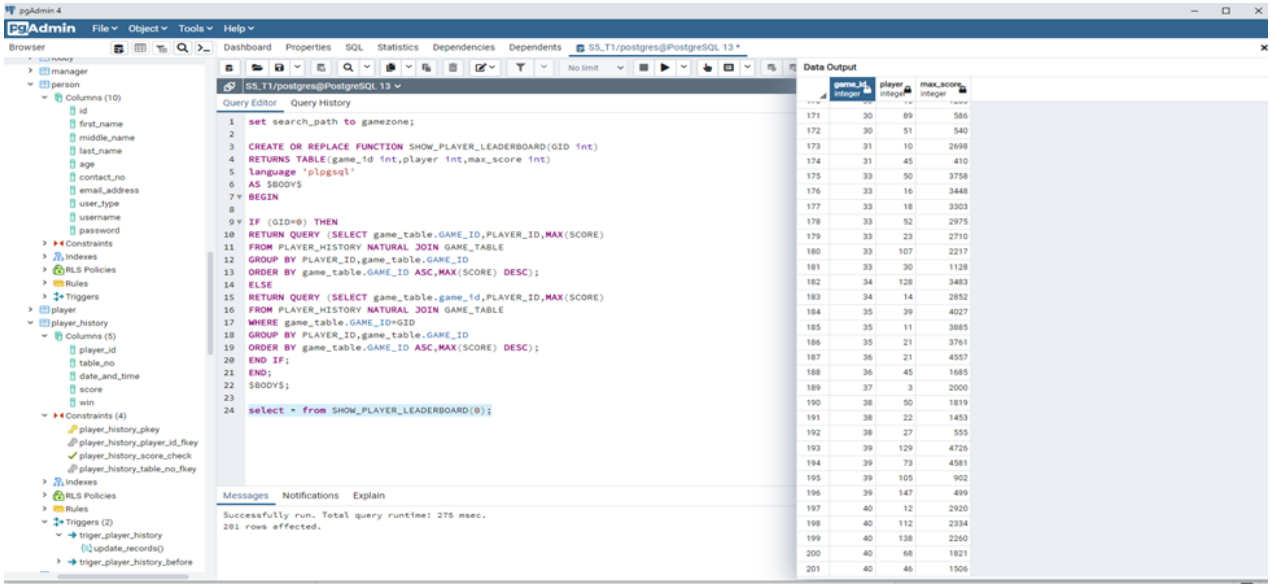
```
with tem as (select t1.player_id,t1.game_id,sum(t2.price)/count(distinct
    t2.player_id) as cum_sum
from (select player_id,game.game_id,price from game,player where type =
    'Racing' order by price) as t1
inner join (select player_id,game.game_id,price from game,player where type
    = 'Racing' order by price) as t2 on t2.price <= t1.price
group by t1.player_id,t1.game_id,t1.price
order by cum_sum)

select account.player_id,count(cum_sum) from account join tem on
    account.player_id=tem.player_id
where cum_sum <= balance
group by account.player_id;
```



- Number of Tuple = 114

34. Show Leaderboard of all games.

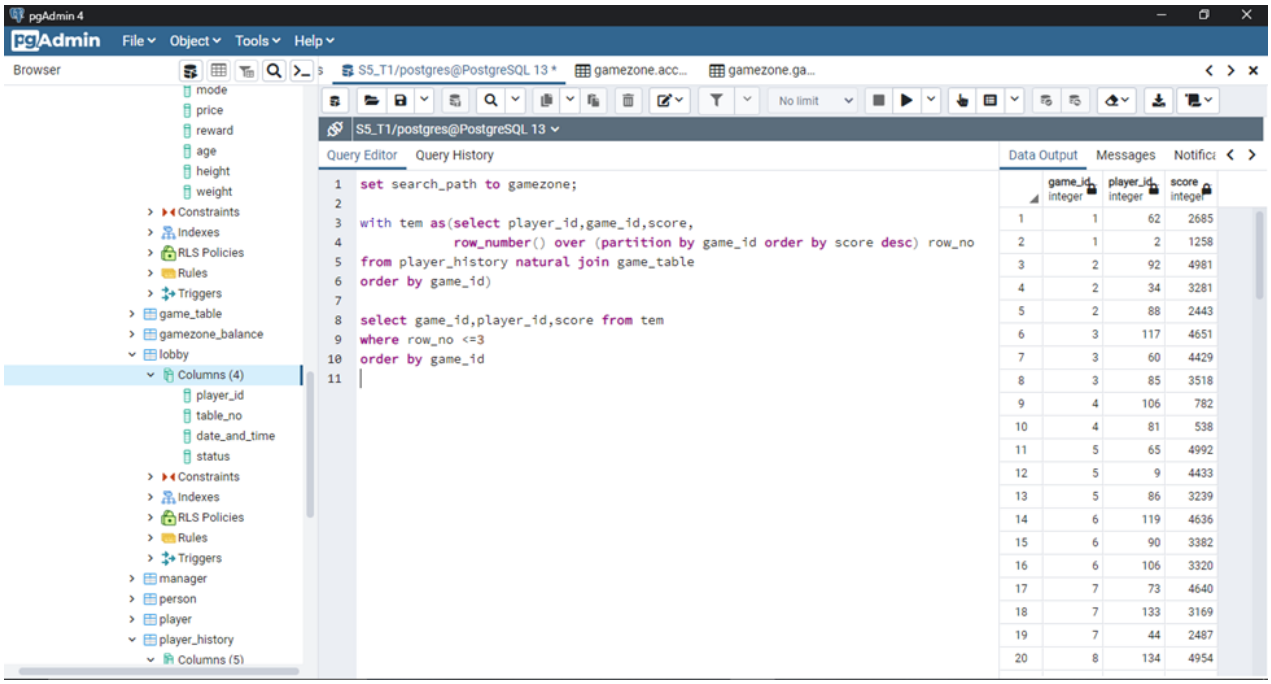
```
select * from SHOW_PLAYER_LEADERBOARD(0);
```



- Number of Tuple = 201

35. List the top 3 scoring players of each game.

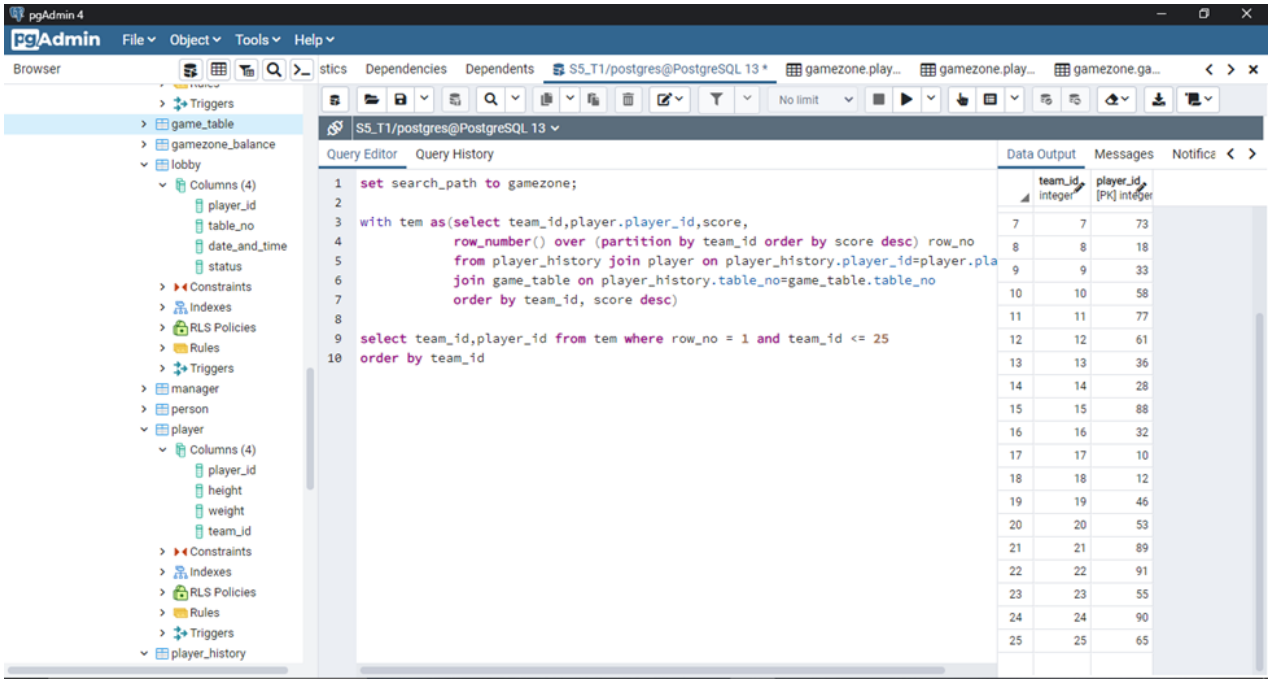
```
with tem as(select player_id,game_id,score,
        row_number() over (partition by game_id order by score desc) row_no
from player_history natural join game_table
order by game_id)

select game_id,player_id,score from tem
where row_no <=3
order by game_id
```



- Number of Tuple = 100

36. List the best player in each team according to higher scores.

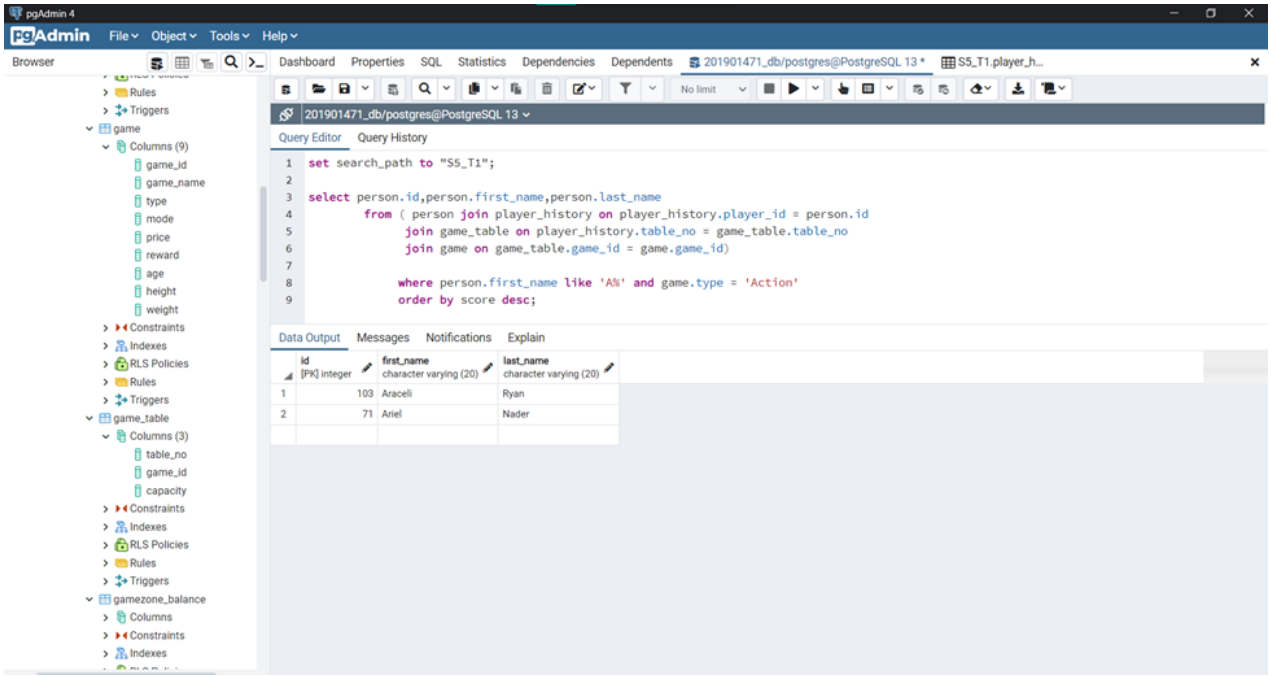
```
with tem as(select team_id,player.player_id,score,
        row_number() over (partition by team_id order by score desc) row_no
        from player_history join player on
           player_history.player_id=player.player_id
        join game_table on player_history.table_no=game_table.table_no
        order by team_id, score desc)

select team_id,player_id from tem where row_no = 1 and team_id <= 25
order by team_id
```



- Number of Tuple = 25

37. List the player id and name whose name starts with letter ‘A’ and order them descending order of their score in Action games.

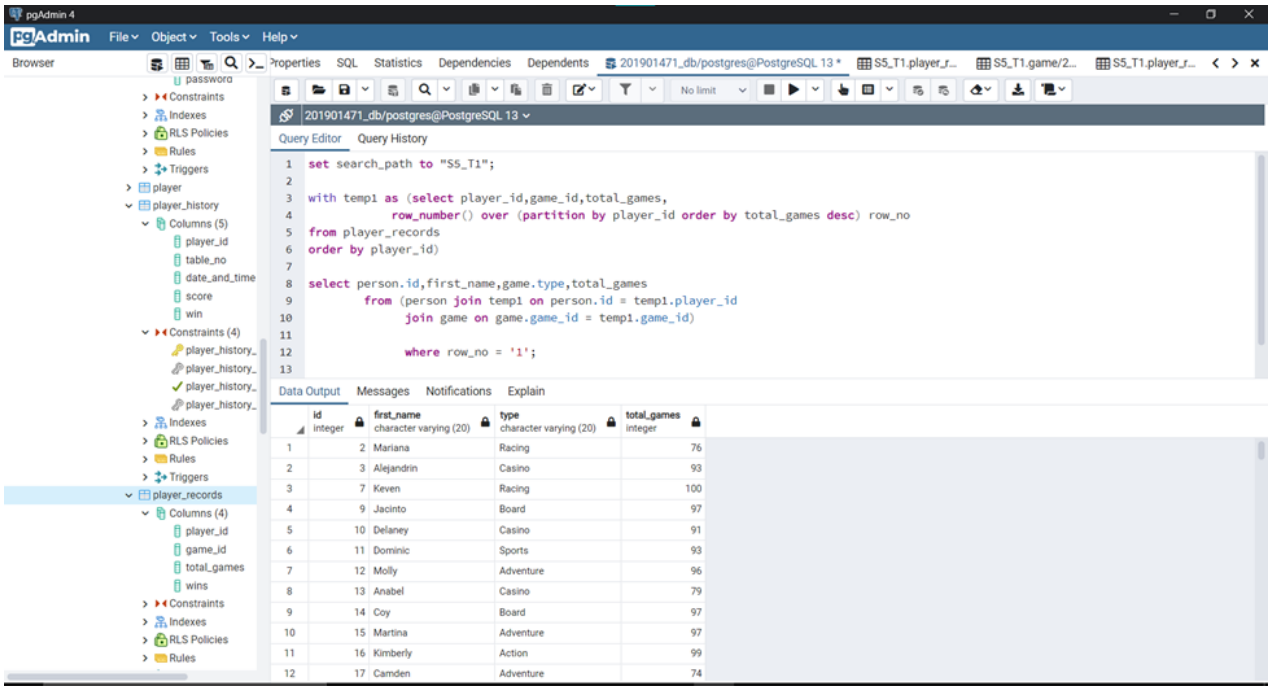
```
select person.id,person.first_name,person.last_name
      from ( person join player_history on player_history.player_id =
          person.id
           join game_table on player_history.table_no = game_table.table_no
           join game on game_table.game_id = game.game_id)

          where person.first_name like 'A%' and game.type = 'Action'
          order by score desc;
```



- Number of Tuple = 2

38. List the player details and which type of game he played most.

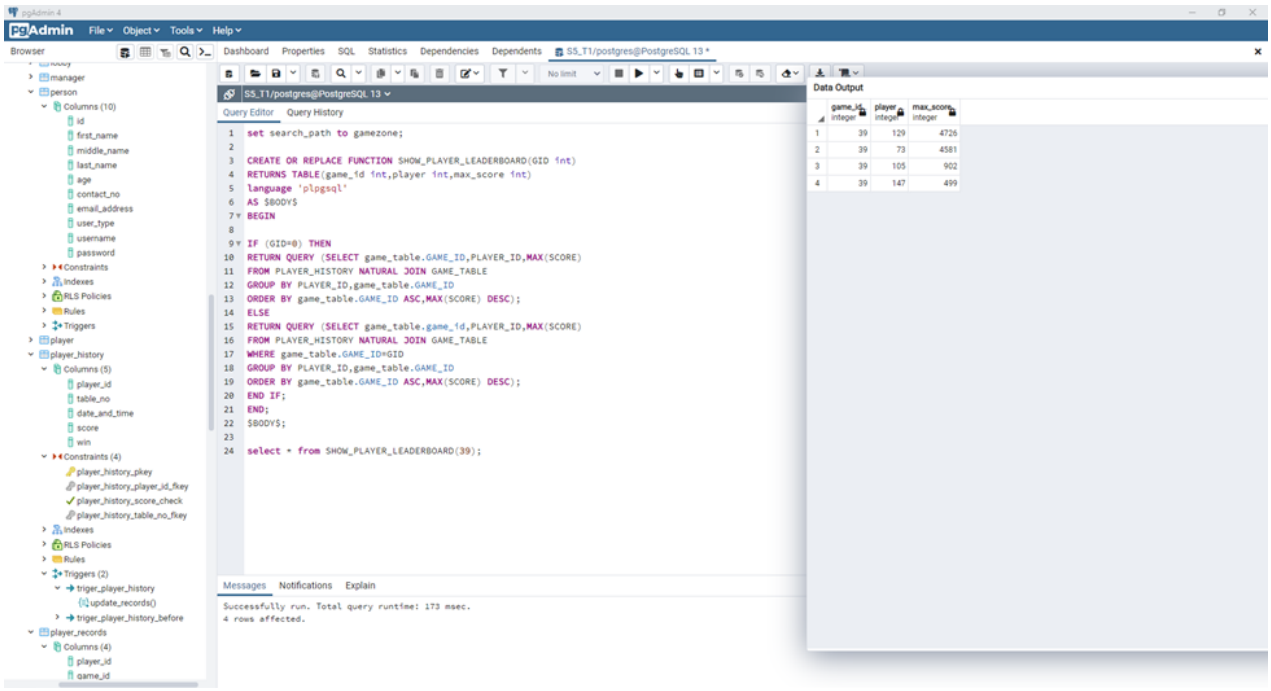
```
with temp1 as (select player_id,game_id,total_games,
        row_number() over (partition by player_id order by total_games
           desc) row_no
from player_records
order by player_id)

select person.id,first_name,game.type,total_games
     from (person join temp1 on person.id = temp1.player_id
          join game on game.game_id = temp1.game_id)

          where row_no = '1';
```



- Number of Tuple = 114

39. Show Leaderboard of game_id 39.

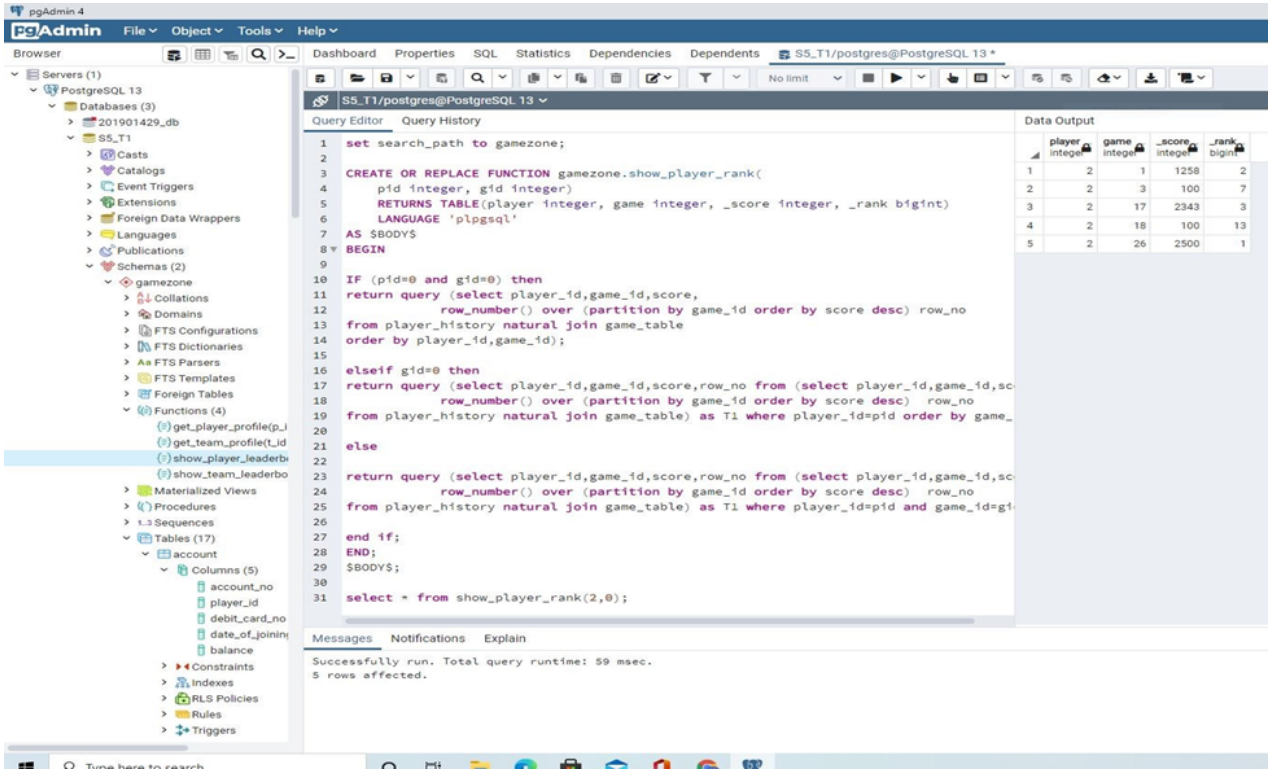
```
select * from SHOW_PLAYER_LEADERBOARD(39);
```



- Number of Tuple = 4

40. Show the rank of player_id 2 in each purchased game.

```
select * from show_player_rank(2,0);
```



- Number of Tuple = 5

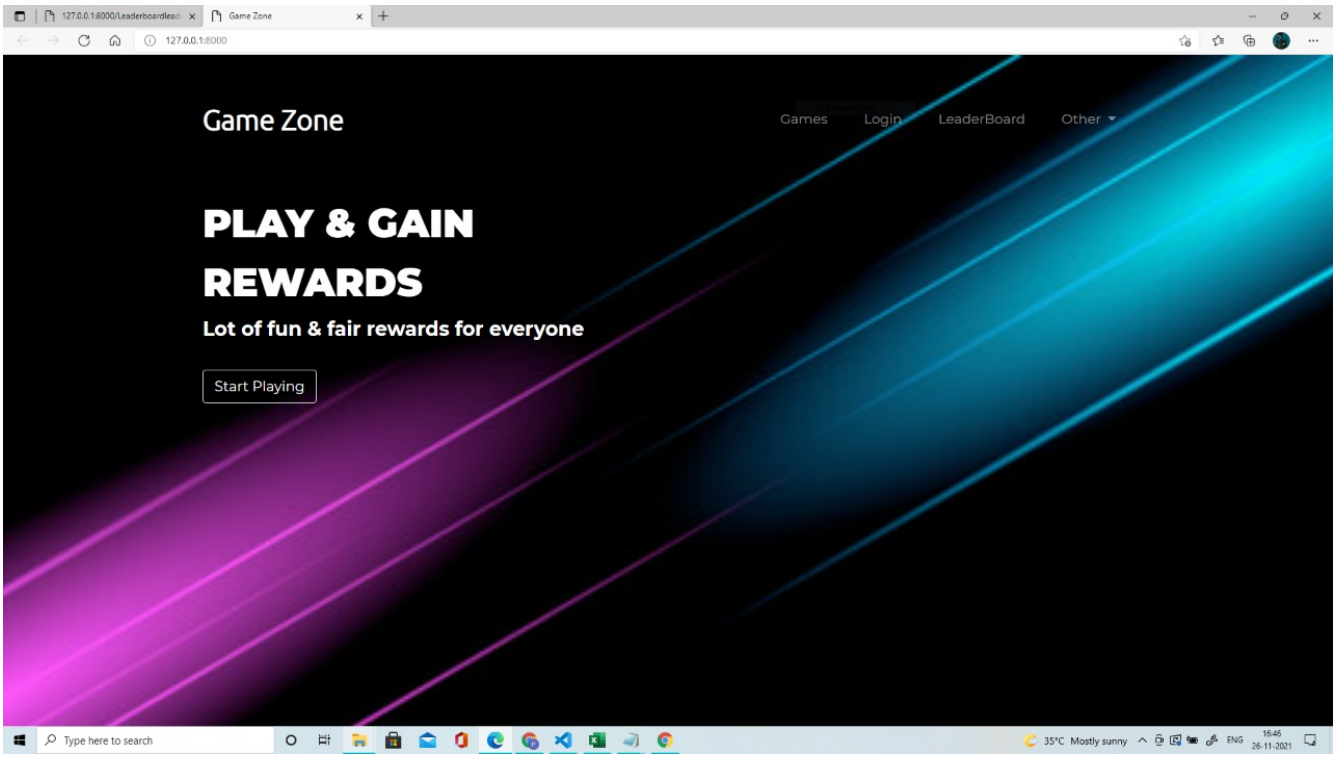
Section 7

Project Code with output screenshots

17 Front-End

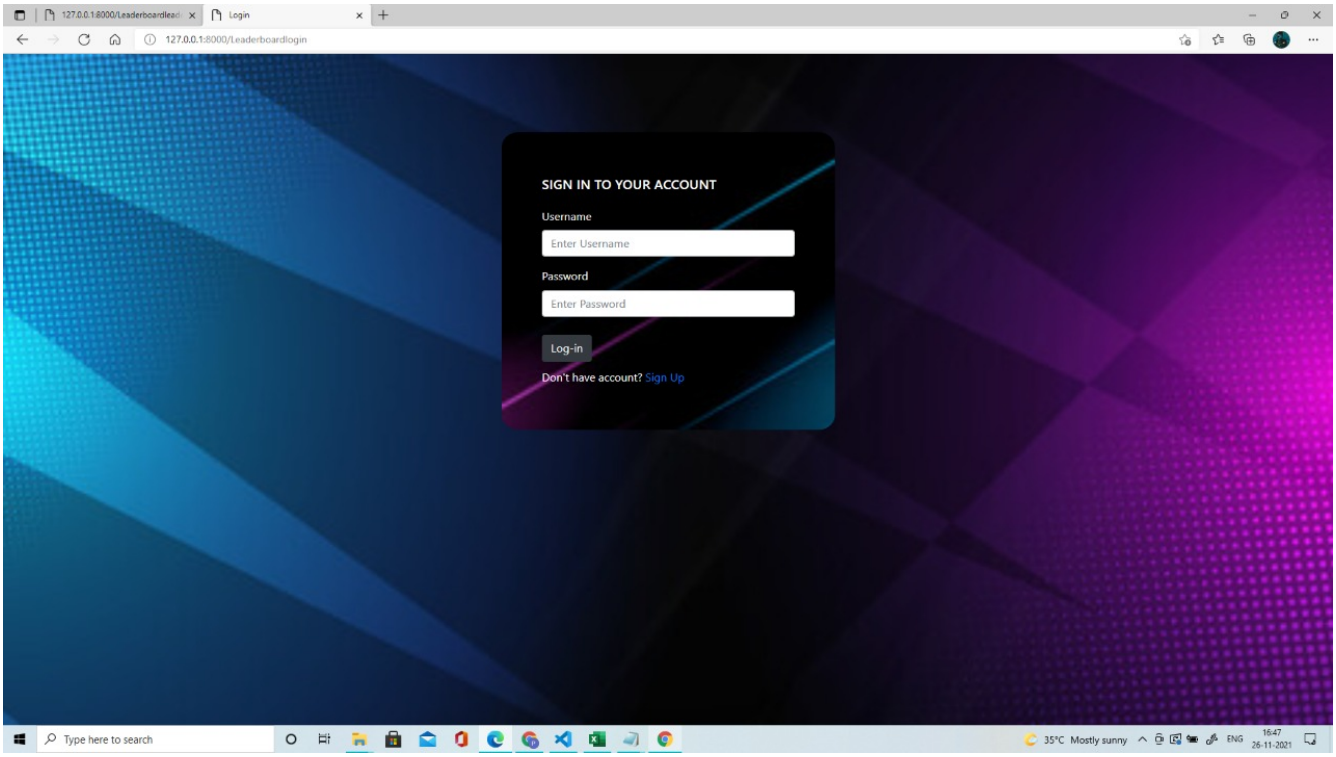
For Front-End Development, we have used the **Django framework**. Django is a high-level Python web framework, it's a collection of python libraries which allow us to create quick, efficient and a quality web application that is suitable for both frontend and backend. To connect the postgresql database to the Django framework we have used the **'psycopg2'** library, which is the most popular PostgreSQL database adapter for the Python programming language. For now, We have created some web pages and implemented some features and SQL queries. We have put screenshots of web-pages and output of queries below.

HOME PAGE:

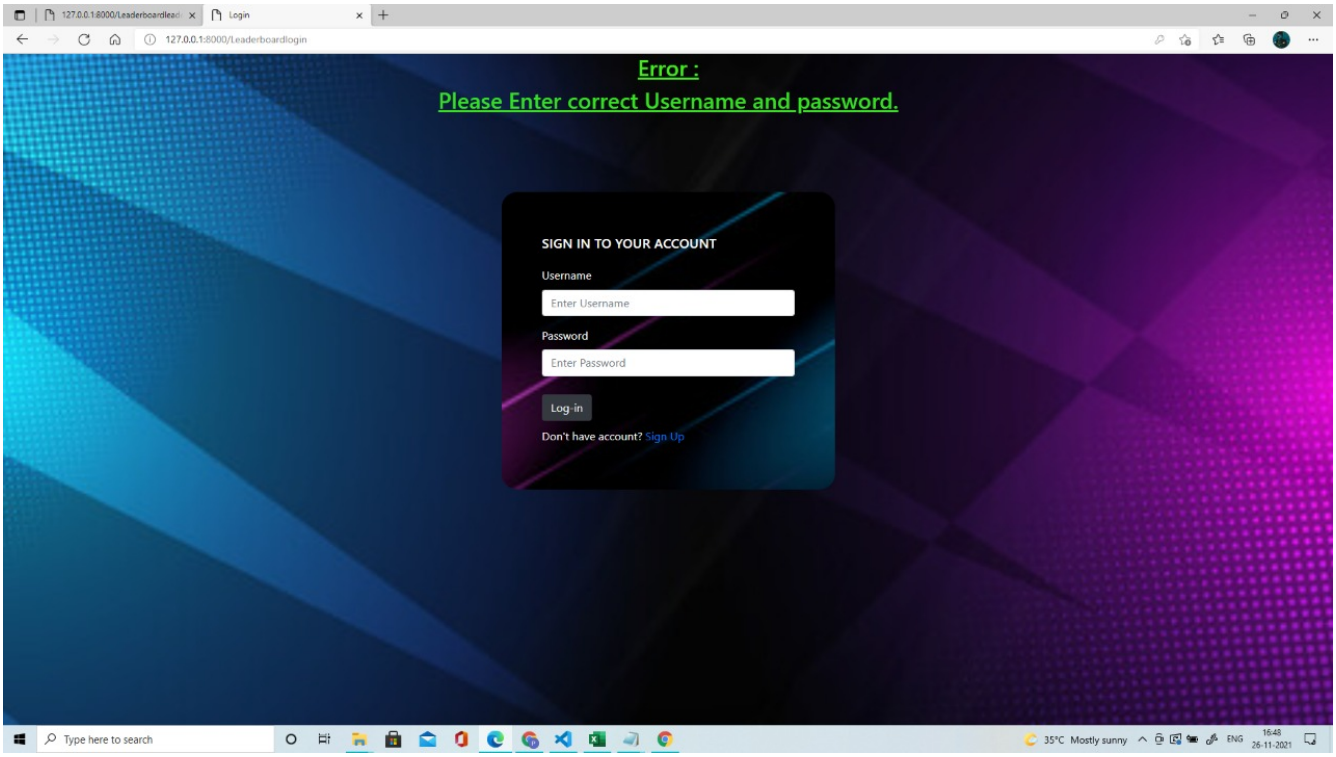


LOGIN PAGE:

After clicking on “Login” in “homepage”, we can move to the login page.

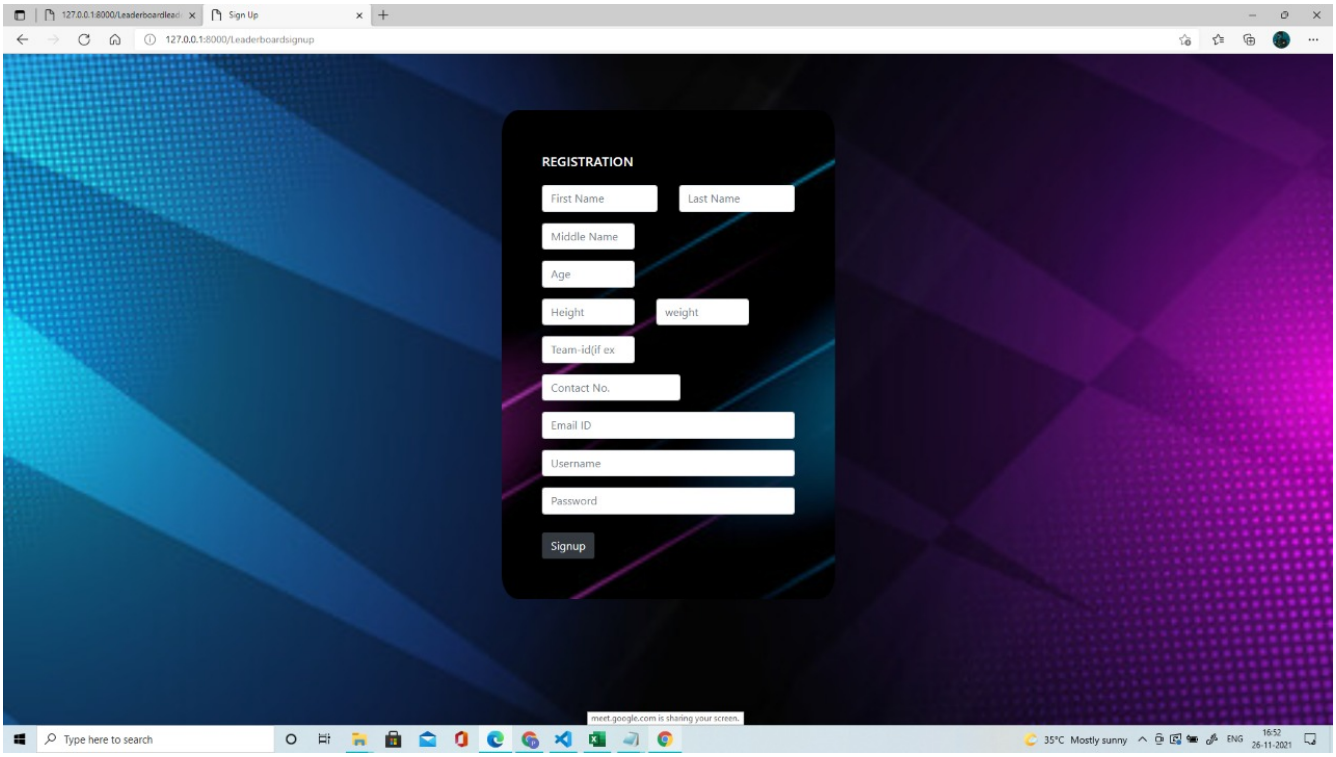


If we enter the wrong “username” or “Password”, the page shows the error message.

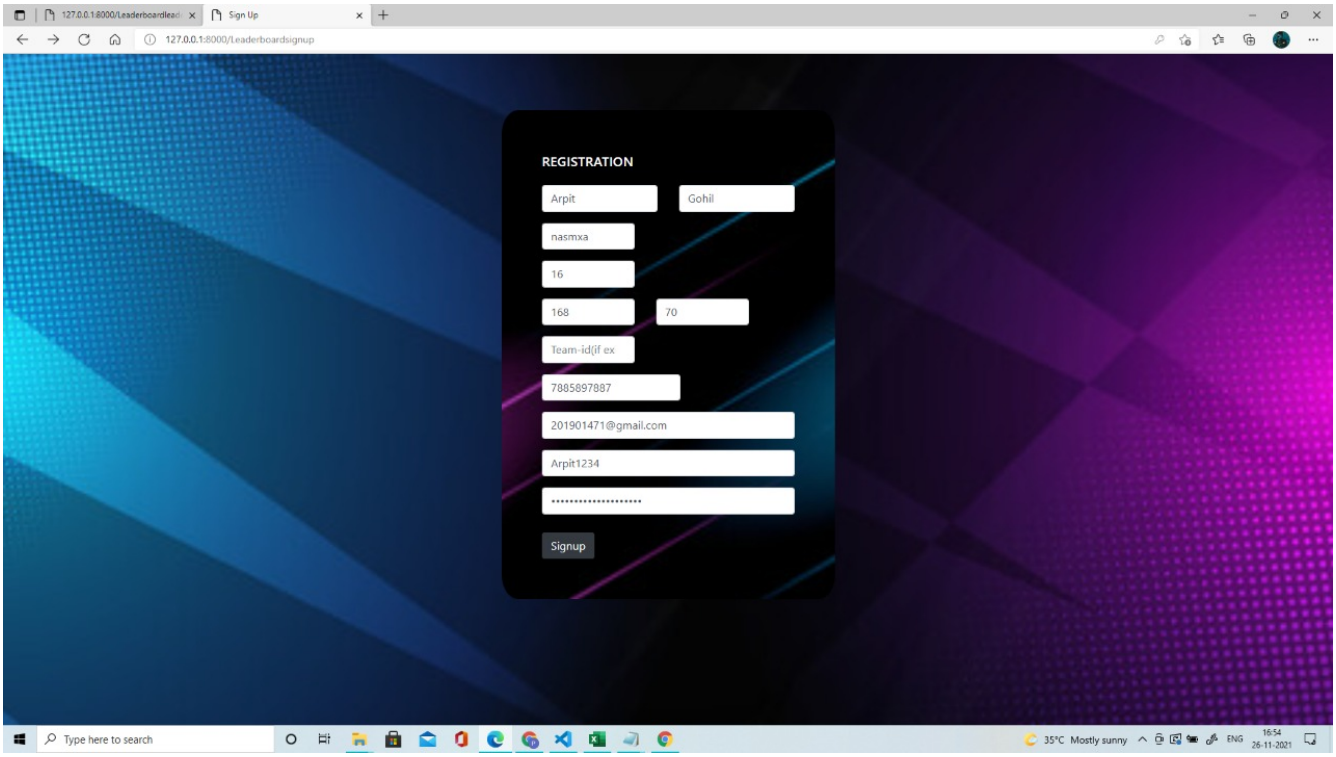


SIGN_UP PAGE:

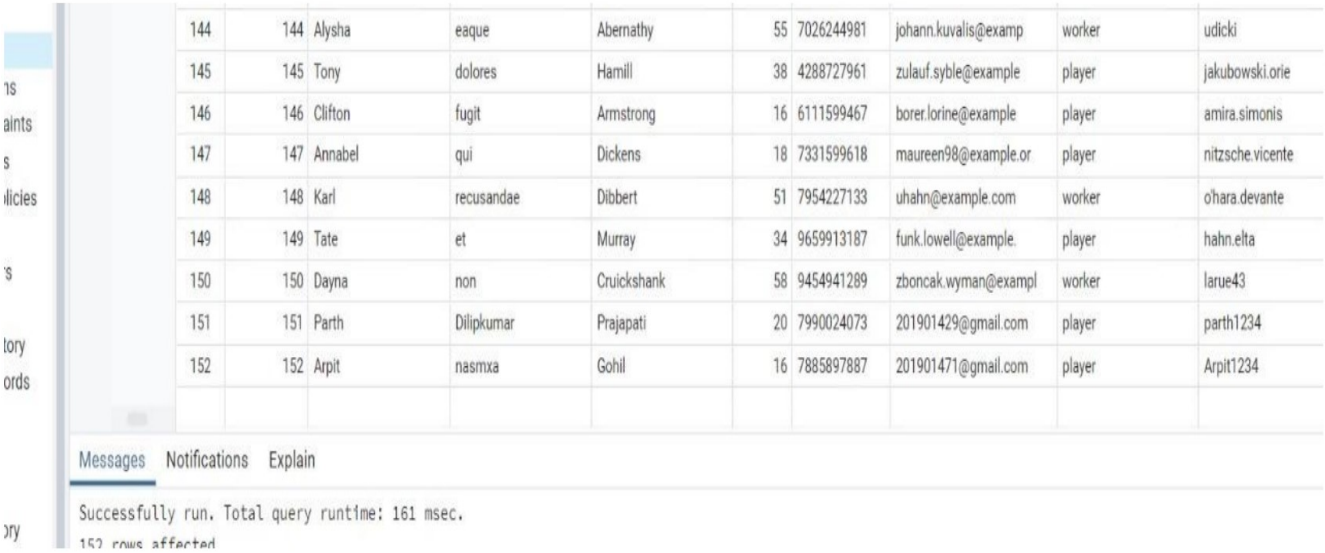
New user can sign up using below page.



Example Demo: New user has filled the above details and sign up.



Now you can see the details of “new user” added in the DATABASE.



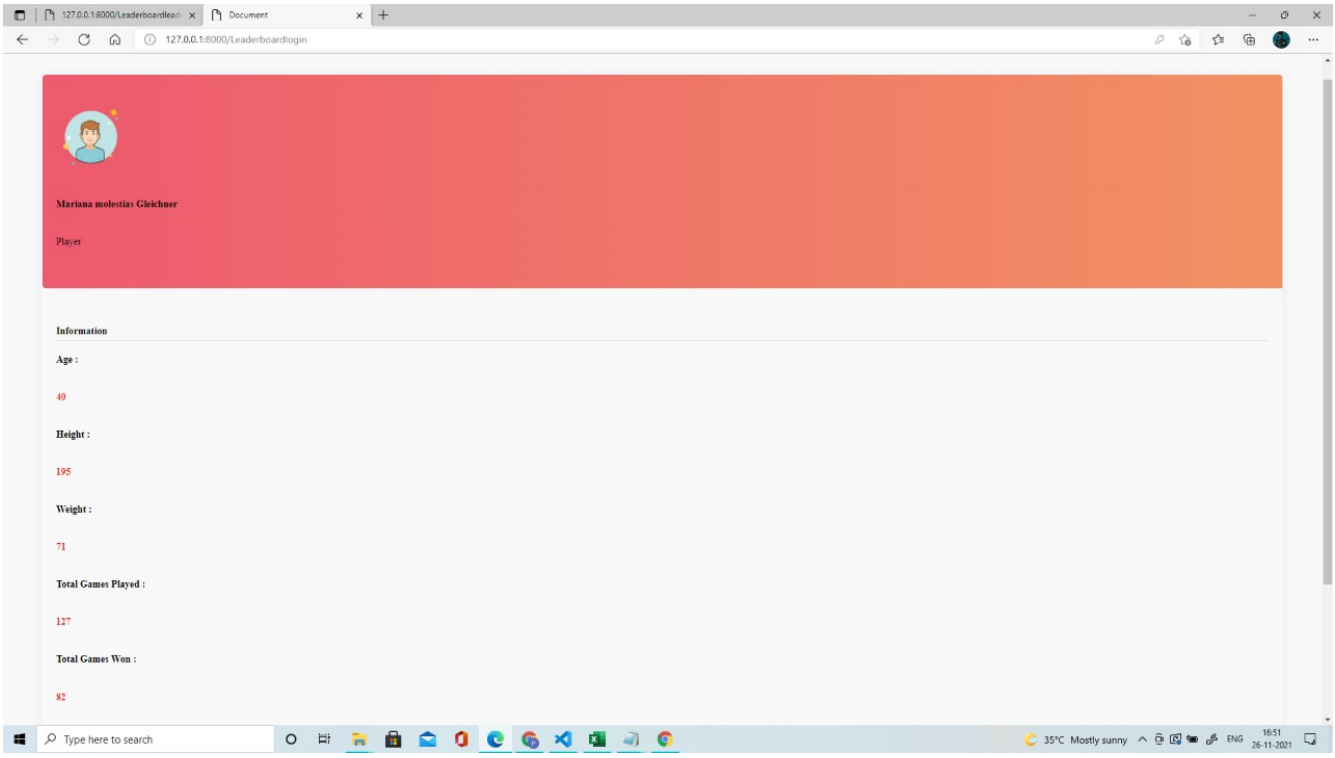
144	144	Alysha	eaque	Abernathy	55	7026244981	johann.kuvalis@examp	worker	udicki
145	145	Tony	dolores	Hamill	38	4288727961	zulauf.syble@example	player	jakubowski.orie
146	146	Clifton	fugit	Armstrong	16	6111599467	borer.lorine@example	player	amira.simonis
147	147	Annabel	qui	Dickens	18	7331599618	maureen98@example.or	player	nitzsche.vicente
148	148	Karl	recusandae	Dibbert	51	7954227133	uhahn@example.com	worker	o'hara.devante
149	149	Tate	et	Murray	34	9659913187	funk.lowell@example.	player	hahn.elta
150	150	Dayna	non	Cruickshank	58	9454941289	zboncak.wyman@exampl	worker	larue43
151	151	Parth	Dilipkumar	Prajapati	20	7990024073	201901429@gmail.com	player	parth1234
152	152	Arpit	nasmxa	Gohil	16	7885897887	201901471@gmail.com	player	Arpit1234

Messages Notifications Explain

Successfully run. Total query runtime: 161 msec.

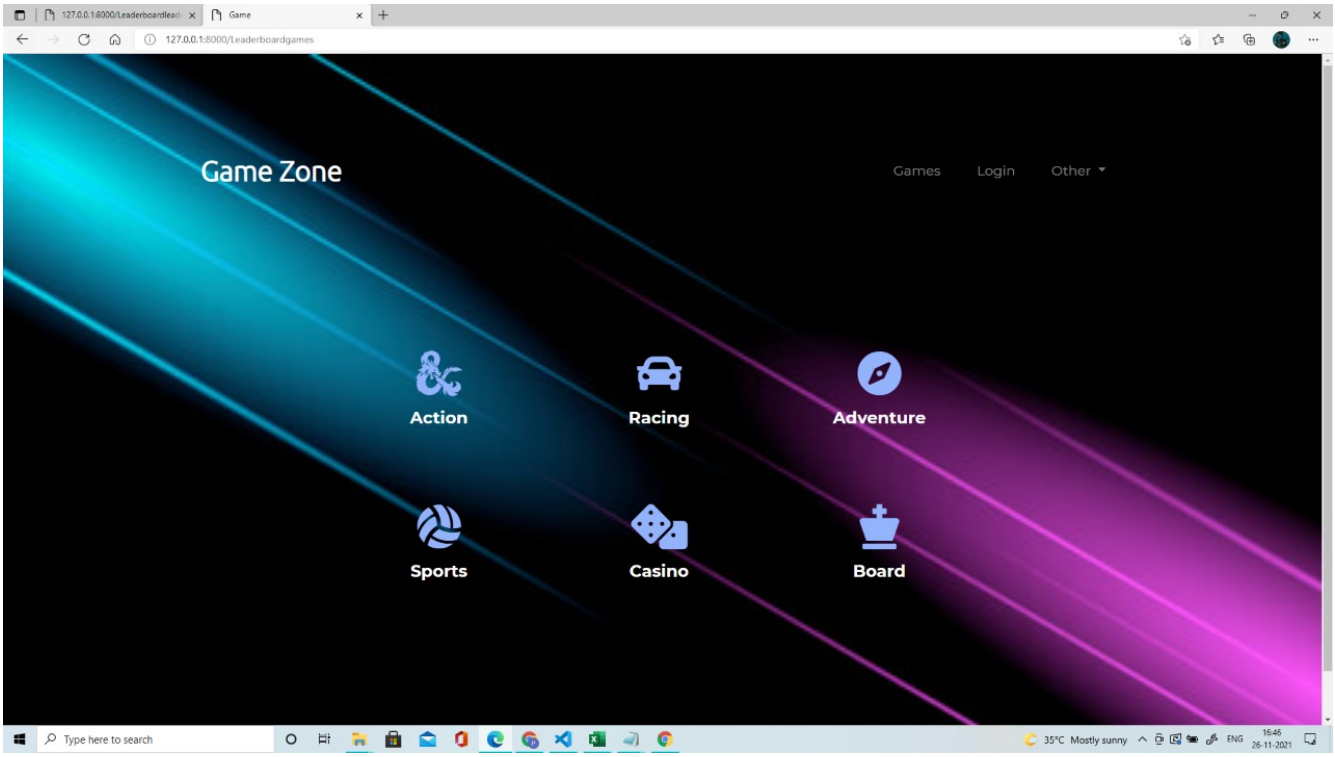
PROFILE PAGE:

After login, users can see their profile.

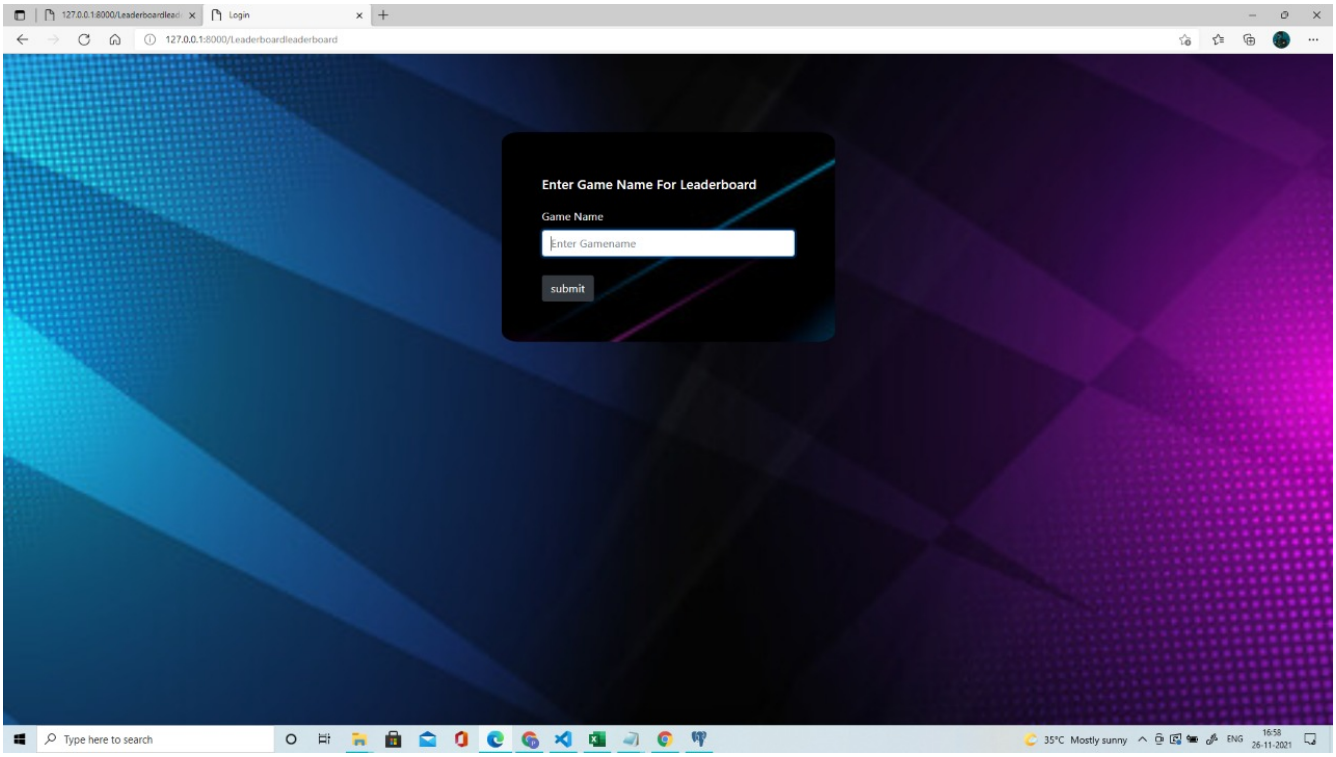


Games:

After clicking on “Games” in “Homepage” the user can see the page below.

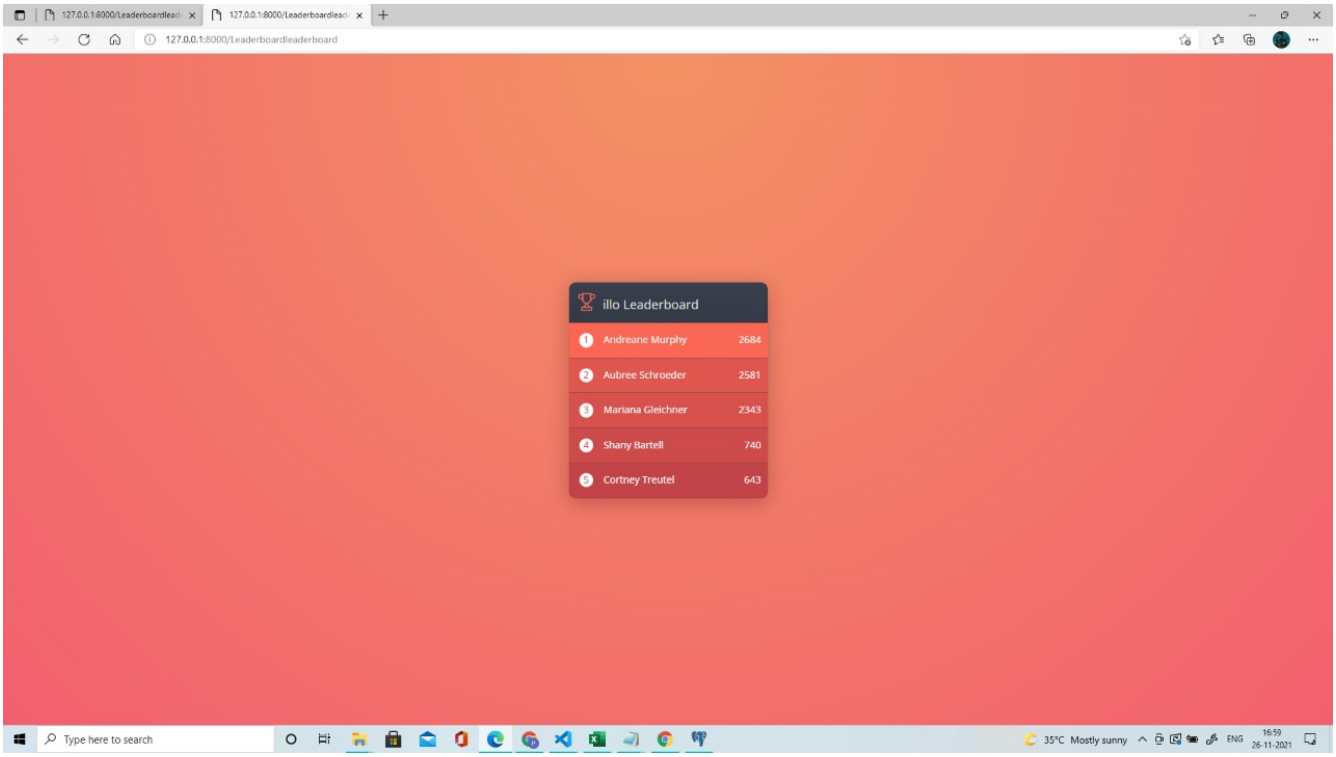


Now user can see the each type of game by clicking on the “Action”, “Racing”, “Adventure”, “Sports”, “Casino”, and “Board”Players can see the leaderboard of each game by entering the “Game Name.



LEADER BOARD:

Player entered the “illo” game and you can see the leader board below.



Project Code link :

https://drive.google.com/drive/folders/1vgTo_5TaNccgsxewNv5uPXFHMXXJdxYH?usp=sharing.

THANK YOU