

Departamento de Engenharia Informática e de Sistemas
Instituto Superior de Engenharia de Coimbra
Tecnologias e Programação de Sistemas de Informação

BD
2017/2018

Base de Dados para a gestão de contas de um jogo

A21270093
A21270246

Leandro Adão Fidalgo
Pedro dos Santos Alves

Índice

Índice	1
1. Introdução	2
2. Enquadramento da proposta	3
2.1. Diagnóstico da Situação Atual	3
2.2. Descrição da Solução Proposta	3
3. Funcionalidades da aplicação	4
3.1. Requisitos e/ou Restrições de implementação	4
3.2. Estrutura da Aplicação	5
3.3. Módulos da aplicação e protótipos	5
Análise de Dados	11
3.4. Entidades	11
• Entidade Servidor	11
• Entidade Conta	12
• Entidade Personagem	13
• Entidade Banco	14
• Entidade Classe	14
• Entidade Clan	15
• Entidade Pertence	16
3.5. Relacionamentos	17
• Relacionamento: contém	17
• Relacionamento: possui	18
• Relacionamento: deposita	19
• Relacionamento: tem	19
• Relacionamento: pertence	20
3.6. Modelo Entidade Relacionamento	23
4. Modelo Físico	24
4.1. Tabelas	24
• Tabela Servidor	24
• Tabela Conta	25
• Tabela Personagem	26
• Tabela Banco	27
• Tabela Classe	28
• Tabela Clan	28
• Tabela Pertence	29
4.2. Diagrama de Tabelas	30
4.3. Script de criação da Base de Dados	31
5. Pesquisas SQL	34
6. Conclusões	35

1. Introdução

O objectivo deste projecto é implementar um sistema que permita efetuar a gestão automatizada de contas de um jogo.

O jogo Shrekovski foi criado pelos irmãos Shrekovski (Leandro e Pedro), com o objetivo de entreter pessoas. O principal objetivo do jogo é matar monstros e ganhar dinheiro.

O jogo possui um conjunto de contas, que vão sendo acumuladas na base de dados.

2. Enquadramento da proposta

Neste capítulo faz-se um enquadramento da proposta de trabalho na gestão de contas do jogo Shrekovski. Na secção 2.1 descreve-se o funcionamento actual do negócio de gestão de contas do jogo. A secção 2.2 descreve a proposta de solução que permitirá solucionar os problemas existentes e satisfazer as necessidades encontradas.

2.1. Diagnóstico da Situação Atual

Atualmente a gestão de contas do jogo é efetuada através de uma interface, que está interligada com a base de dados. Esta interface permite criar, editar, apagar, e visualizar o conteúdo das contas.

2.2. Descrição da Solução Proposta

A base de dados a implementar deve ser capaz de armazenar dados de conta, servidor, personagem, clã, banco e classe das personagens.

É criado uma base de dados, que será o servidor, e do outro lado para o cliente, irá existir uma aplicação de modo a conseguir interagir com a base de dados do jogo e fazer a gestão das contas do jogo.

O administrador do jogo terá á sua disposição tudo o que necessitar para a gestão das contas.

Por outro lado, os utilizadores poderão criar contas, fazer login e visualizar o seu perfil de conta.

3. Funcionalidades da aplicação

3.1. Requisitos e/ou Restrições de implementação

Nesta secção enumeram-se as restrições que a aplicação deve satisfazer de modo a permitir um correto funcionamento da gestão de contas de um jogo. As restrições a implementar são as seguintes:

- Não deve poder ser criado um utilizador que já exista (o *username* e o email terão de ser diferentes).
- Não deve ser permitido o *login* quando a conta estiver banida (o atributo *isBanned* = 1).
- Não deve ser permitido o *login* quando a conta já estiver logada (o atributo *isLogged* = 1).
- Não deve ser permitido criar mais que 5 personagens na mesma conta (o atributo *slot* <= 5).
- Não podem ser criadas personagens com um nome já existente.
- Uma conta só pode estar ligada a um servidor.
- Cada personagem só pode ter um clã.
- O nome do clã não se pode repetir.
- Uma personagem só pode ter uma classe.
- Um clã tem de ter obrigatoriamente um líder.
- Um clã só pode ter um líder (*isLeader*).
- Só pode existir 10 sublíderes em cada clã.

3.2. Estrutura da Aplicação

Os módulos a implementar são os seguintes:

- **Gestão de servidor**
- **Gestão de contas**
- **Gestão de personagens**
- **Gestão de clãs**
- **Gestão de classes**
- **Visualização de perfil**

3.3. Módulos da aplicação e protótipos

- **Gestão de servidores**

Após a selecção da opção “Gestão de servidores” no menu principal da aplicação deve ser apresentada a interface abaixo:

O módulo de gestão dos servidores deve permitir na mesma interface todas as operações possíveis de executar relativamente à informação dos servidores. Nomeadamente:

- Inserir a informação de um novo servidor. Aquando da inserção, devem ser executadas as seguintes operações:
 - Geração automática do código interno que será associado a esse servidor. Este código deve ser gerado de uma forma sequencial.
 - Verificar se a informação introduzida satisfaz as restrições definidas na base de dados:
 - o nome do servidor não é nulo;
 - o nome do servidor é único, não existe ainda na base de dados;
 - a região do servidor não é nula.
- Editar a informação de um servidor já existente. Aquando da alteração, devem ser executadas as seguintes operações:
 - Selecionar o servidor cuja informação se pretende alterar. Esta selecção deve ser possível através da escolha do nome do respectivo servidor.
 - Após a selecção do servidor e as alterações desejadas terem sido efectuadas, é necessário verificar se a informação introduzida satisfaz as restrições definidas na base de dados:
 - o nome do servidor não é nulo;

- o nome do servidor é único, não existe ainda na base de dados;
 - a região do servidor não é nula.
- Apagar a informação de um servidor já existente. Aquando da remoção, devem ser executadas as seguintes operações:
 - Selecionar o servidor cuja informação se pretende remover. Esta selecção deve ser possível através da escolha do nome do respectivo servidor.
 - Após a seleção do servidor, deve-se pedir ao utilizador que confirme a intenção de remover esse servidor. Se existirem contas nesse servidor deve-se indicar ao utilizador que irão ser apagadas contas.
- Fechar a interface de gestão de servidores.

• **Gestão de contas**

Após a seleção da opção “Gestão de contas” no menu principal da aplicação deve ser apresentada a interface abaixo:

O módulo de gestão das contas deve permitir na mesma interface todas as operações possíveis de executar relativamente à informação das contas. Nomeadamente:

- Inserir a informação de uma nova conta. Aquando da inserção, devem ser executadas as seguintes operações:
 - Geração automática do código interno que será associado a essa conta. Este código deve ser gerado de uma forma sequencial.
 - Verificar se a informação introduzida satisfaz as restrições definidas na base de dados:
 - o *username* da conta não é nulo e deve ser escrito em minúscula;
 - o nome do *username* é único, não existe ainda na base de dados;
 - a password não é nula;
 - o email não é nulo.

- Editar a informação de uma conta já existente. Aquando da alteração, devem ser executadas as seguintes operações:
 - Selecionar a conta cuja informação se pretende alterar. Esta seleção deve ser possível através da escolha do *username* da respetiva conta.
 - Após a seleção da conta e as alterações desejadas terem sido efectuadas, é necessário verificar se a informação introduzida satisfaz as restrições definidas na base de dados:
 - o *username* da conta não é nulo e deve ser escrito em minúscula;
 - o nome do *username* é único, não existe ainda na base de dados;
 - a password não é nula;
 - o email não é nulo.
- Apagar a informação de uma conta já existente. Aquando da remoção, devem ser executadas as seguintes operações:
 - Selecionar a conta cuja informação se pretende remover. Esta seleção deve ser possível através da escolha do *username* da respetiva conta.
 - Após a seleção da conta, deve-se pedir ao utilizador que confirme a intenção de remover essa conta. Se existirem personagens nessa conta deve-se indicar ao utilizador que irão ser apagadas as personagens.
- Fechar a interface de gestão de contas.

- **Gestão de personagens**

Após a selecção da opção “Gestão de personagens” no menu principal da aplicação deve ser apresentada a interface abaixo:

O módulo de gestão de personagens deve permitir na mesma interface todas as operações possíveis de executar relativamente à informação das personagens. Nomeadamente:

- Inserir a informação de uma nova personagem. Aquando da inserção, devem ser executadas as seguintes operações:

- Geração automática do código interno que será associado a essa personagem. Este código deve ser gerado de uma forma sequencial.
- Verificar se a informação introduzida satisfaz as restrições definidas na base de dados:
 - o nome da personagem não é nula;
 - o nome da personagem é único, não existe ainda na base de dados;
 - o slot da personagem não é nulo;
 - o género da personagem não é nulo;
 - o ID_Clan pode ser nulo;
 - o ID_Class não pode ser nulo;
 - o ID_Conta não pode ser nulo.
- Editar a informação de uma personagem já existente. Aquando da alteração, devem ser executadas as seguintes operações:
 - Seleccionar a personagem cuja informação se pretende alterar. Esta selecção deve ser possível através da escolha do nome da respectiva personagem.
 - Após a selecção da personagem e as alterações desejadas terem sido efectuadas, é necessário verificar se a informação introduzida satisfaz as restrições definidas na base de dados:
 - o nome da personagem não é nula;
 - o nome da personagem é único, não existe ainda na base de dados;
 - o slot da personagem não é nulo;
 - o género da personagem não é nulo;
 - o ID_Clan pode ser nulo;
 - o ID_Class não pode ser nulo;
 - o ID_Conta não pode ser nulo.
- Apagar a informação de uma personagem já existente. Aquando da remoção, devem ser executadas as seguintes operações:
 - Selecionar a personagem cuja informação se pretende remover. Esta selecção deve ser possível através da escolha do nome da respectiva personagem.
 - Após a selecção da personagem, deve-se pedir ao utilizador que confirme a intenção de remover essa personagem.
- Fechar a interface de gestão de personagens.

- **Gestão de clãs**

Após a selecção da opção “Gestão de clãs” no menu principal da aplicação deve ser apresentada a interface abaixo:

O módulo de gestão dos clãs deve permitir na mesma interface todas as operações possíveis de executar relativamente à informação dos clãs. Nomeadamente:

- Inserir a informação de um novo clã. Aquando da inserção, devem ser executadas as seguintes operações:
 - Geração automática do código interno que será associado a esse clã. Este código deve ser gerado de uma forma sequencial.
 - Verificar se a informação introduzida satisfaz as restrições definidas na base de dados:
 - o nome do clã não é nulo;
 - o nome do clã é único, não existe ainda na base de dados.
- Editar a informação de um clã já existente. Aquando da alteração, devem ser executadas as seguintes operações:
 - Seleccionar o clã cuja informação se pretende alterar. Esta selecção deve ser possível através da escolha do nome do respectivo clã.
 - Após a selecção do clã e as alterações desejadas terem sido efectuadas, é necessário verificar se a informação introduzida satisfaz as restrições definidas na base de dados:
 - o nome do clã não é nulo;
 - o nome do clã é único, não existe ainda na base de dados.
- Apagar a informação de um clã já existente. Aquando da remoção, devem ser executadas as seguintes operações:
 - Selecionar o clã cuja informação se pretende remover. Esta selecção deve ser possível através da escolha do nome do respectivo clã.
 - Após a selecção do clã, deve-se pedir ao utilizador que confirme a intenção de remover esse clã. Se existirem personagens nesse clã deve-se indicar ao utilizador que irão ser removidas as personagens.
- Fechar a interface de gestão de clãs.

- **Gestão de classes**

Após a selecção da opção “Gestão de classes” no menu principal da aplicação deve ser apresentada a interface abaixo:

O módulo de gestão de classes deve permitir na mesma interface todas as operações possíveis de executar relativamente à informação das classes. Nomeadamente:

- Inserir a informação de uma nova classe. Aquando da inserção, devem ser executadas as seguintes operações:
 - Geração automática do código interno que será associado a essa classe. Este código deve ser gerado de uma forma sequencial.
 - Verificar se a informação introduzida satisfaz as restrições definidas na base de dados:
 - o nome do classe não é nulo;
 - o nome do classe é único, não existe ainda na base de dados;
 - o TipoDano e TipoAtaque não são nulos.
- Editar a informação de uma classe já existente. Aquando da alteração, devem ser executadas as seguintes operações:
 - Selecionar a classe cuja informação se pretende alterar. Esta seleção deve ser possível através da escolha do nome da respectiva classe.
 - Após a seleção da classe e as alterações desejadas terem sido efectuadas, é necessário verificar se a informação introduzida satisfaz as restrições definidas na base de dados:
 - o nome do classe não é nulo;
 - o nome do classe é único, não existe ainda na base de dados;
 - o TipoDano e TipoAtaque não são nulos.
- Apagar a informação de uma classe já existente. Aquando da remoção, devem ser executadas as seguintes operações:
 - Selecionar a classe cuja informação se pretende remover. Esta seleção deve ser possível através da escolha do nome do respectiva classe.
 - Após a seleção da classe, deve-se pedir ao utilizador que confirme a intenção de remover essa classe.
- Fechar a interface de gestão de classes.

Análise de Dados

3.4. Entidades

Nesta secção vão ser descritas todas as Entidades relevantes para o negócio da gestão de contas do jogo Shrekovski. Após uma análise aprofundada do modelo da gestão de contas, constatou-se a necessidade das seguintes Entidades:

- **Servidor**
 - **Conta**
 - **Personagem**
 - **Banco**
 - **Clan**
 - **Classe**
 - **Pertence**
- **Entidade Servidor**

A entidade servidor representa a informação relativa aos servidores que existem no jogo.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
ID_Servidor	Numérico	Código interno único atribuído a cada servidor. É um número de 1 dígito gerado sequencialmente. Ex. 2
Região	3 Caracteres	Nome da região em maiúscula. Ex. USA
Nome_Serv	15 Caracteres	Nome do servidor. Ex. Sagitário

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_Servidor	N	S	Identificador (chave primária), não admite nulos. Não existem dois servidores com o mesmo ID.
Região	N	N	Identificador, não admite nulos, só admite maiúsculas. Podem existir duas regiões com o mesmo nome.
Nome_Serv	N	S	Identificador (chave candidata), não admite nulos. Não existem dois servidores com o mesmo nome.

Relacionamentos da Entidade:

Nome da relação	Cardinalidade	Entidade Relacionada	Participação Obrigatória	Página
contem	1 : N	Conta	Conta	

• Entidade Conta

A entidade conta representa a informação relativa às contas dos utilizadores que existem nos servidores. É inserido uma nova conta nesta Entidade, sempre que um utilizador se registar.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
ID_Conta	Numérico	Código interno único atribuído a cada conta. É um número de 8 dígitos gerado sequencialmente. Ex. 92233002
Username	12 Caracteres	Nome do utilizador, tendo no máximo 12 caracteres. Ex. tunafish
Email	50 Caracteres	Email do utilizador. Ex. abc@asd.pt
Password_Conta	15 Caracteres	Password do utilizador. Ex. Pedrito1234
ID_Servidor	Numérico	Chave estrangeira, contém o ID do servidor, onde se localiza a conta. Ex. 2
isGM	Numérico (1 dígito)	Valor que indica se a conta é administradora no servidor. Este valor só pode ser 0 ou 1. Ex. 1
isLogged	Numérico (1 dígito)	Valor que indica se a conta está ligada ao servidor. Este valor só pode ser 0 ou 1. Ex. 1
isBanned	Numérico (1 dígito)	Valor que indica se a conta está banida no servidor. Este valor só pode ser 0 ou 1. Ex. 0

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_Conta	N	S	Identificador (chave primária), não admite nulos. Não existem duas contas com o mesmo ID.
Username	N	S	Identificador do utilizador, tendo no máximo 12 caracteres.
Email	N	S	Identificador do email, tendo no máximo 50 caracteres.
Password_Conta	N	N	Identificador da password, tendo no máximo 15 caracteres.
ID_Servidor	N	S	Identificador (chave estrangeira), não admite nulos. Não existem dois servidores com o mesmo ID.
isGM	N	N	Valor que indica se a conta é administradora no servidor. Este valor só pode ser 0 ou 1. Ex. 1
isLogged	N	N	Este valor só pode ser 0 ou 1. Ex. 1
isBanned	N	N	Este valor só pode ser 0 ou 1. Ex. 0

Relacionamentos da Entidade:

Nome da relação	Cardinalidade	Entidade Relacionada	Participação Obrigatória	Página
-----------------	---------------	----------------------	--------------------------	--------

Está contida	N : 1	Servidor	Conta	
Possui	1 : N	Personagem	Personagem	

• Entidade Personagem

A entidade personagem representa a informação relativa às personagens existentes nas contas. É inserido uma nova personagem nesta Entidade, sempre que o utilizador criar uma personagem.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
ID_Personagem	Numérico	Código interno único atribuído a cada personagem. É um número de 15 dígitos gerado sequencialmente. Ex. 922330234432567
ID_Conta	Numérico	Código interno único atribuído a cada conta. É um número de 8 dígitos gerado sequencialmente. Ex. 92233002
ID_Class	Numérico	Código interno único atribuído a cada class. É um número de 2 dígitos gerado sequencialmente. Ex. 14
Nome_Pers	15 Caracteres	Nome da personagem. Ex. PedritoAves
Slot	Numérico (1 dígitos)	Valor indicador do lugar onde a personagem se encontra. Sendo apenas valores de 1 a 5. Ex. 4
Genero	1 Caracteres	Carater indicador do género da personagem. Apenas pode ser atribuído os valores 'M' ou 'F'. Ex: M

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_Personagem	N	S	Identificador (chave primária), não admite nulos. Não existem duas personagens com o mesmo ID.
ID_Conta	N	N	Não admite nulos.
ID_Class	N	N	Não admite nulos.
Nome	N	S	Identificador (chave candidata), não admite valor nulos.
Slot	N	N	Valor não nulo.
Genero	N	N	Valor não nulo com o valor M ou F.

Relacionamentos da Entidade:

Nome da relação	Cardinalidade	Entidade Relacionada	Participação Obrigatória	Página
Possui	N : 1	Conta	Personagem	
Deposita	N : 1	Banco	Personagem	
Pertence	N : 1	Clan	Clan	
Tem	N : 1	Class	Personagem	

• Entidade Banco

A entidade banco representa a informação relativa aos créditos(dinheiro) das personagem.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
ID_Personagem	Numérico	Código interno único atributo a cada personagem. É um número de 15 dígitos gerado sequencialmente. Ex. 922330234432567
Creditos	Numérico (7 dígitos)	Informação referente aos créditos que uma personagem tem, tendo no máximo 7 dígitos. Ex. 9999999

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_Personagem	N	S	Identificador (chave primária), não admite nulos. Não existem duas contas bancárias com o mesmo ID.
Creditos	N	N	Não admite nulos.

Relacionamentos da Entidade:

Nome da relação	Cardinalidade	Entidade Relacionada	Participação Obrigatória	Página
É depositado	1 : N	Personagem	Personagem	

• Entidade Classe

A entidade classe representa a informação relativa às classes que cada personagem pode ter.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
ID_Class	Numérico	Código interno único atributo a cada class. É um número de 2 dígitos gerado sequencialmente. Ex. 14
Nome_Class	12 Caracteres	Nome do classe, tendo no máximo 12 caracteres. Ex. Ranger
TipoDano	10 Caracteres	Tipo do dano da classe. Ex. Fisico
TipoAtaque	10 Caracteres	Tipo do ataque da classe. Ex. Longe

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_Class	N	S	Identificador (chave primária), não admite nulos. Não existem duas classes com o mesmo ID.
Nome_Class	N	S	Identificador (chave candidata), não admite nulos. Não existem duas classes com o mesmo nome.
TipoDano	N	N	Não admite nulos.
TipoAtaque	N	N	Não admite nulos

Relacionamentos da Entidade:

Nome da relação	Cardinalidade	Entidade Relacionada	Participação Obrigatória	Página
Tem	1 : N	Personagem	Personagem	

• Entidade Clan

A entidade clã representa a informação relativa aos clãs das personagens. É inserido um novo clan nesta Entidade, sempre que uma personagem criar um clã novo.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
ID_Clan	Numérico	Código interno único atributo a cada clan. É um número de 4 dígitos gerado sequencialmente. Ex. 9224
Nome_Clan	12 Caracteres	Nome do clan, tendo no máximo 12 caracteres. Ex. osmachomens

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_Clan	N	S	Identificador (chave primária), não admite nulos. Não existem dois clans com o mesmo ID.
Nome_Clan	N	S	Identificador (chave candidata), não admite nulos. Não existem dois clans com o mesmo nome.

Relacionamentos da Entidade:

Nome da relação	Cardinalidade	Entidade Relacionada	Participação Obrigatória	Página
Pertence	1 : N	Personagem	Clan	

• Entidade Pertence

A entidade pertence representa a informação relativa às personagens que pertencem a um clã.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
ID_Clan	Numérico	Código interno único atributo a cada clan. Ex. 14
ID_Personagem	Numérico	Código interno único atributo a cada personagem. Ex. 1
IsLeader	Numérico (1 dígito)	Valor que indica se a personagem é leader. Este valor só pode ser 0 ou 1. Ex. 1
IsSubleader	Numérico (1 dígito)	Valor que indica se a personagem é subleader. Este valor só pode ser 0 ou 1. Ex. 1

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_Clan	N	S	Identificador (chave primária), não admite nulos. Não existem dois clans com o mesmo ID.
ID_Personagem	N	S	Identificador (chave primária), não admite nulos. Não existem duas personagens com o mesmo ID.
IsLeader	N	N	Não admite nulos.
IsSubleader	N	N	Não admite nulos.

3.5. Relacionamentos

Nesta secção são descritos todos os relacionamentos existentes entre as várias entidades. Após uma análise aprofundada da gestão de contas de um jogo, constatou-se a necessidade dos seguintes relacionamentos:

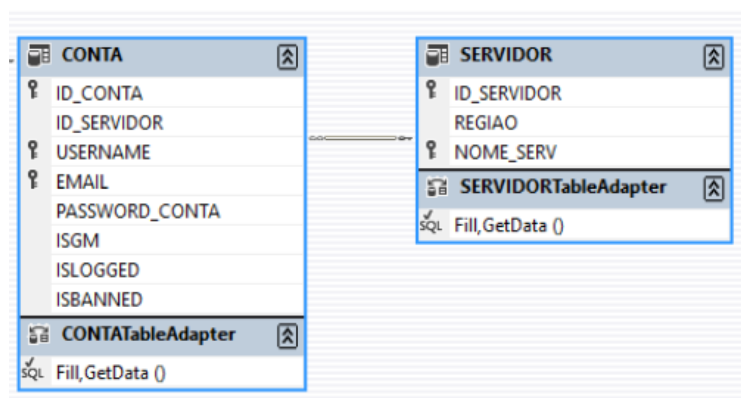
- contém
 - possui
 - deposita
 - tem
 - pertence
- **Relacionamento: contém**

Este relacionamento pretende expressar a relação existente entre as Entidades Servidor e Conta. O objectivo é expressar quais os Servidores que contém que Contas e vice-versa.

Após uma análise do funcionamento da gestão de contas, e as orientações definidas pelo cliente, definiu-se as seguintes condições:

 - um servidor contém várias contas,
 - uma conta está contida num servidor,
 - uma conta está necessariamente num servidor, caso contrário não se regista a conta.

Tomando estas condições em consideração, definiu-se as seguintes características



Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Servidor	Não	1 : N	Sim	Conta
Observações				
<ul style="list-style-type: none"> • uma conta, está obrigatoriamente contida em apenas um servidor • um servidor pode conter várias contas registadas 				

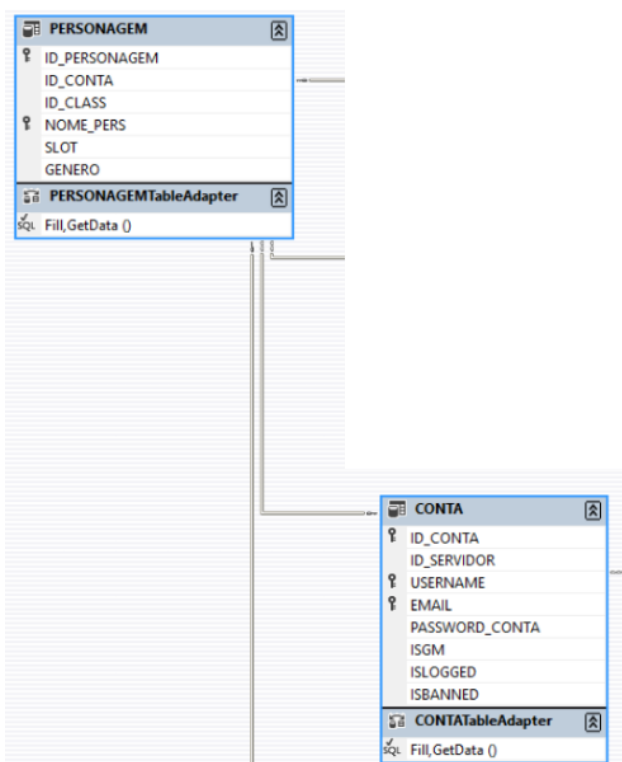
• Relacionamento: possui

Este relacionamento pretende expressar a relação existente entre as Entidades Conta e Personagem. O objectivo é expressar quais as Contas que possuem que Personagens e vice-versa.

Após uma análise do funcionamento da gestão de contas, e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- uma conta possui várias personagens,
- uma personagem está possuída numa conta,
- uma personagem está necessariamente numa conta, caso contrário não se regista a personagem.

Tomando estas condições em consideração, definiu-se as seguintes características



Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Conta	Não	1 : N	Sim	Personagem
Observações				
<ul style="list-style-type: none"> • uma personagem, está obrigatoriamente possuída em apenas uma conta • uma conta pode conter várias personagens registadas 				

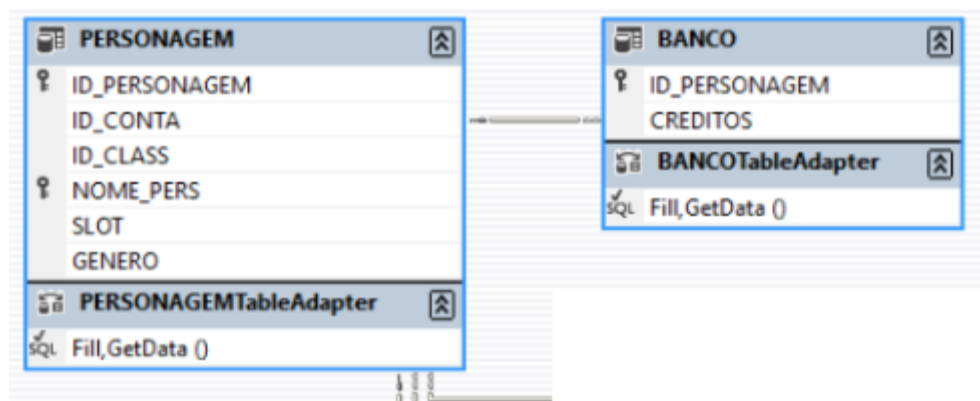
• Relacionamento: deposita

Este relacionamento pretende expressar a relação existente entre as Entidades Personagem e Banco. O objectivo é expressar quais as Personagens que depositaram no Banco e vice-versa.

Após uma análise do funcionamento da gestão de contas, e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- uma personagem deposita num banco,
- num banco é depositado por várias personagens,
- uma personagem obrigatoriamente deposita num banco, caso contrário não se regista a personagem.

Tomando estas condições em consideração, definiu-se as seguintes características



Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Personagem	Sim	N : 1	Não	Banco
Observações				
<ul style="list-style-type: none"> • uma personagem, <u>obrigatoriamente</u> deposita em apenas <u>um</u> banco • num banco <u>pode</u> estar depositado o dinheiro de <u>várias</u> personagens registadas 				

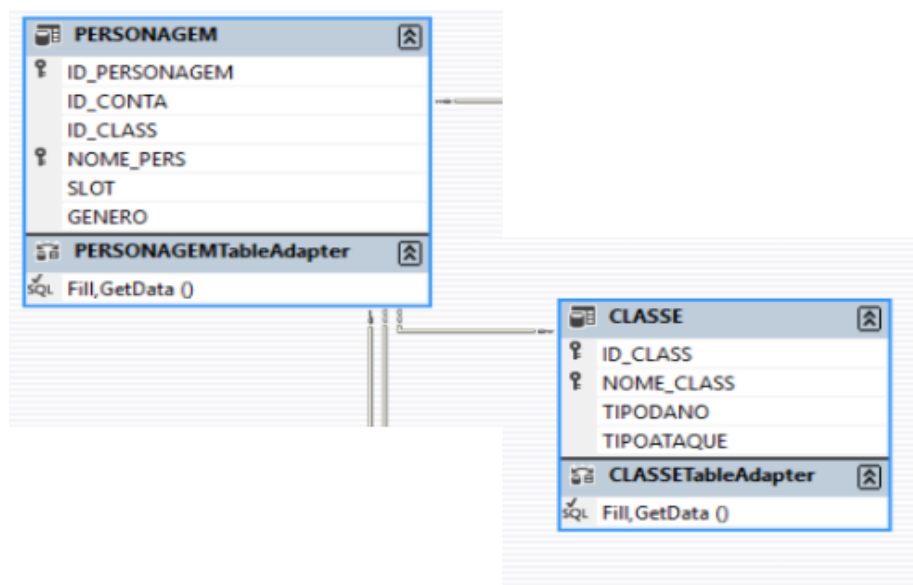
• Relacionamento: tem

Este relacionamento pretende expressar a relação existente entre as Entidades Personagem e Class. O objectivo é expressar quais as Personagens que têm que Classes e vice-versa.

Após uma análise do funcionamento da gestão de contas, e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- uma personagem tem uma classe,
- uma classe tem várias personagens,
- uma personagem tem necessariamente uma classe, caso contrário não se regista a personagem.

Tomando estas condições em consideração, definiu-se as seguintes características



Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Personagem	Sim	N : 1	Não	Classe
Observações				
<ul style="list-style-type: none"> uma personagem, tem obrigatoriamente apenas uma classe uma classe pode ter várias personagens registadas 				

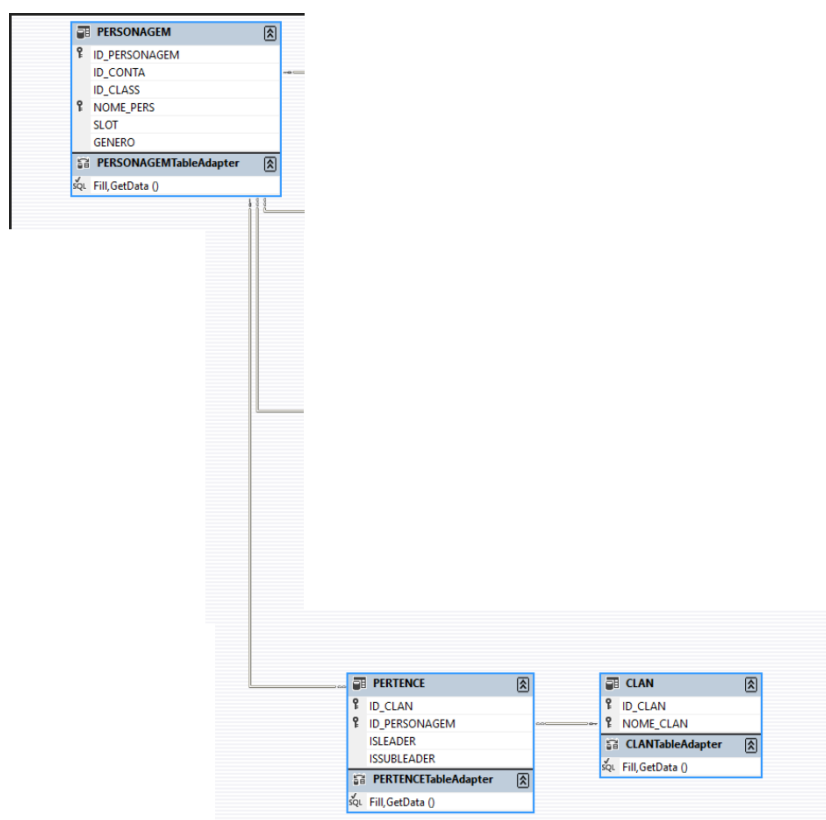
• Relacionamento: pertence

Este relacionamento pretende expressar a relação existente entre as Entidades Personagem e Clan. O objectivo é expressar quais as Personagens que pertencem a um Clans e vice-versa.

Após uma análise do funcionamento da gestão de contas, e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- uma personagem pertence a um clan,
- a um clan pertencem várias personagens,
- a um clan pertence necessariamente uma personagem, caso contrário não se regista o clan.

Tomando estas condições em consideração, definiu-se as seguintes características



Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Personagem	Não	N : 1	Sim	Clan
Observações				
<ul style="list-style-type: none"> a um clan, pertence obrigatoriamente uma ou várias personagens uma personagem pode apenas pertencer a um clan 				

Atributos relevantes do Relacionamento:

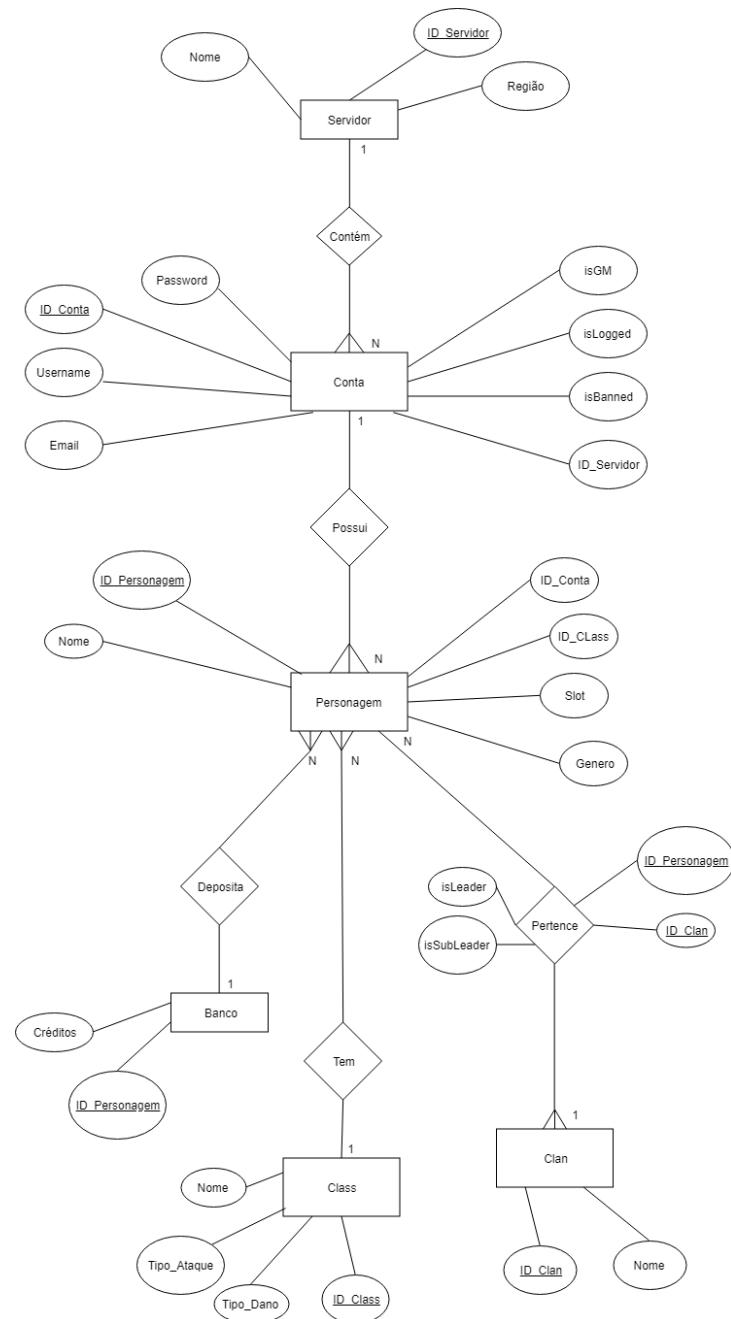
Nome do atributo	Tipo de Dados	Descrição
ID_Clan	Numérico (4 dígitos)	Código interno único atributo a cada clan. É um número de 4 dígitos gerado sequencialmente. Ex. 9224
ID_Personagem	Numérico (15 dígitos)	Código interno único atributo a cada personagem. É um número de 15 dígitos gerado sequencialmente. Ex. 922330234432567
isLeader	Numérico (1 dígitos)	Valor que indica se a personagem é líder do clan. Este valor só pode ser 0 ou 1. Ex. 1
isSubLeader	Numérico (1 dígitos)	Valor que indica se a personagem é líder do clan. Este valor só pode ser 0 ou 1. Ex. 1

Restrições dos atributos do Relacionamento:

Nome do atributo	Aceita Nulos ?	Valores Únicos?	Observações
ID_Clan	N	S	Identificador (chave primária), não admite nulos. Não existem dois clans com o mesmo ID.
ID_Personagem	N	S	Identificador (chave primária), não admite nulos. Não existem dois clans com o mesmo nome.
isLeader	N	N	Valor não nulo com o valor 0 ou 1.
isSubLeader	N	N	Valor não nulo com o valor 0 ou 1.

3.6. Modelo Entidade Relacionamento

O modelo de Entidade Relacionamento completo é o seguinte.



4. Modelo Físico

4.1. Tabelas

Nesta secção são descritas as tabelas necessárias para implementar o negócio de venda de livros da livraria. Estas tabelas foram extraídas, tomando em consideração o modelo Entidade / Relacionamento definido no capítulo 4. As tabelas são as seguintes:

- **Servidor**
 - **Conta**
 - **Personagem**
 - **Banco**
 - **Clan**
 - **Classe**
 - **Pertence**
- **Tabela Servidor**

A tabela servidor representa a informação relativa aos servidores que existem no jogo.

Atributos relevantes da tabela: Servidor

Nome do atributo	Tipo de Dados	Descrição
ID_Servidor	Numérico	Código interno único atributo a cada servidor. É um número de 1 dígito gerado sequencialmente. Ex. 2
Região	3 Caracteres	Nome da região em maiúscula. Ex. USA
Nome_Serv	15 Caracteres	Nome do servidor. Ex. Sagitário

Restrições dos atributos da tabela: SERVIDOR

Atributo(s)	Nome da Restrição	Tipo de Restrição	Observações
ID_Servidor	PK_Serv	chave primária	PRIMARY KEY
Regiao	SYS_C00243242	Verificação	"Regiao" IS NOT NULL
Nome_Serv	SYS_C00243243	Vrificação	"Nome_Serv" IS NOT NULL

• Tabela Conta

A tabela conta representa a informação relativa às contas dos utilizadores que existem nos servidores. É inserido uma nova conta nesta tabela, sempre que um utilizador se registar.

Atributos relevantes da tabela: Conta

Nome do atributo	Tipo de Dados	Descrição
ID_Conta	Numérico	Código interno único atribuído a cada conta. É um número de 8 dígitos gerado sequencialmente. Ex. 92233002
Username	12 Caracteres	Nome do utilizador, tendo no máximo 12 caracteres. Ex. tunafish
Email	50 Caracteres	Email do utilizador. Ex. abc@asd.pt
Password_Conta	15 Caracteres	Password do utilizador. Ex. Pedrito1234
ID_Servidor	Numérico	Chave estrangeira, contém o ID do servidor, onde se localiza a conta. Ex. 2
isGM	Numérico (1 dígito)	Valor que indica se a conta é administradora no servidor. Este valor só pode ser 0 ou 1. Ex. 1
isLogged	Numérico (1 dígito)	Valor que indica se a conta está ligada ao servidor. Este valor só pode ser 0 ou 1. Ex. 1
isBanned	Numérico (1 dígito)	Valor que indica se a conta está banida no servidor. Este valor só pode ser 0 ou 1. Ex. 0

Restrições dos atributos da tabela: LIVROS

Atributo(s)	Nome da Restrição	Tipo de Restrição	Observações
ID_Conta	PK_Conta	chave primária	PRIMARY KEY
Username	SYS_C00243247	Verificação	CHECK "USERNAME" IS NOT NULL
Email	SYS_C00243248	Verificação	CHECK "EMAIL" IS NOT NULL
Password_Conta	SYS_C00243249	Verificação	CHECK "PASSWORD_CONTA" IS NOT NULL
ID_Servidor	SYS_C00243246	Verificação	CHECK
ID_Servidor	FK_Serv_Conta	chave estrangeira	CHECK
isGM	SYS_C00243250	Verificação	CHECK "ISGM" IS NOT NULL
isLogged	SYS_C00243251	Verificação	CHECK "ISLOGGED" IS NOT NULL
isBanned	SYS_C00243252	Verificação	CHECK
isGM	SYS_C00243253	Verificação	CHECK ISGM = 0 OR ISGM = 1
isLogged	SYS_C00243254	Verificação	CHECK ISLOGGED = 0 OR ISLOGGED = 1
isBanned	SYS_C00243255	Verificação	CHECK ISBANNED = 0 OR ISBANNED = 1
Username	SYS_C00243258	Verificação	UNIQUE USERNAME
Email	SYS_C00243259	Verificação	UNIQUE EMAIL

Tabelas relacionadas e respectivas restrições de Integridade Referencial

Tabela	Nome da Restrição	Observações
SERVIDOR	FK_Serv_Conta	FOREIGN KEY(ID_Servidor) REFERENCES Servidor(ID_Servidor)

• Tabela Personagem

A tabela personagem representa a informação relativa às personagens existentes nas contas. É inserido uma nova personagem nesta tabela, sempre que o utilizador criar uma personagem.

Atributos relevantes da tabela: Personagem

Nome do atributo	Tipo de Dados	Descrição
ID_Personagem	Numérico	Código interno único atributo a cada personagem. É um número de 15 dígitos gerado sequencialmente. Ex. 922330234432567
ID_Conta	Numérico	Código interno único atributo a cada conta. É um número de 8 dígitos gerado sequencialmente. Ex. 92233002
ID_Class	Numérico	Código interno único atributo a cada class. É um número de 2 dígitos gerado sequencialmente. Ex. 14
Nome_Pers	15 Caracteres	Nome da personagem. Ex. PedritoAves
Slot	Numérico (1 dígitos)	Valor indicador do lugar onde a personagem se encontra. Sendo apenas valores de 1 a 5. Ex. 4
Genero	1 Caracteres	Carater indicador do género da personagem. Apenas pode ser atribuído os valores 'M' ou 'F'. Ex: M

Restrições dos atributos da tabela: PERSONAGEM

Atributo(s)	Nome da Restrição	Tipo de Restrição	Observações
ID_Personagem	PK_Pers	chave primária	PRIMARY KEY
ID_Conta	FK_Conta_Pers	chave estrangeira	FOREIGN KEY
ID_Class	FK_Class_Pers	chave estrangeira	FOREIGN KEY
Nome_Pers	SYS_C00243268	Verificação	CHECK "NOME_PERS" IS NOT NULL
Slot	SYS_C00243269	Verificação	CHECK "SLOT" IS NOT NULL
Genero	SYS_C00243270	Verificação	CHECK "GENERO" IS NOT NULL
ID_Conta	SYS_C00243266	Verificação	CHECK "ID_CONTA" IS NOT NULL
ID_Class	SYS_C00243267	Verificação	CHECK "ID_CLASS" IS NOT NULL
Slot	SYS_C00243271	Verificação	CHECK SLOT >= 1 AND SLOT <= 5
Genero	SYS_C00243272	Verificação	CHECK GENERO = 'M' OR GENERO = 'F'

Nome_Pers	SYS_C00243276	Único	UNIQUE NOME_PERS
-----------	---------------	-------	------------------

Tabelas relacionadas e respectivas restrições de Integridade Referencial

Tabela	Nome da Restrição	Observações
Conta	FK_Conta_Pers	FOREIGN KEY(ID_Conta) REFERENCES Conta(ID_Conta)
Class	FK_Class_Pers	FOREIGN KEY(ID_Class) REFERENCES Classe(ID_Class)

• Tabela Banco

A tabela banco representa a informação relativa aos créditos(dinheiro) das personagem.

Atributos relevantes da tabela: Banco

Nome do atributo	Tipo de Dados	Descrição
ID_Personagem	Numérico	Código interno único atributo a cada personagem. É um número de 15 dígitos gerado sequencialmente. Ex. 922330234432567
Creditos	Numérico (7 dígitos)	Informação referente aos créditos que uma personagem tem, tendo no máximo 7 dígitos. Ex. 9999999

Restrições dos atributos da tabela: BANCO

Atributo(s)	Nome da Restrição	Tipo de Restriçã	Observações
ID_Personagem	PK_Banco	chave primária	PRIMARY KEY
ID_Personagem	FK_Pers_Banco	chave estrangeira	FOREIGN KEY
Creditos	SYS_C00243279	Verificação	CHECK "CREDITOS" IS NOT NULL

Tabelas relacionadas e respectivas restrições de Integridade Referencial

Tabela	Nome da Restrição	Observações
Personagem	FK_Pers_Banco	FOREIGN KEY(ID_Personagem) REFERENCES Personagem(ID_Personagem)

• Tabela Classe

A tabela classe representa a informação relativa às classes que cada personagem pode ter.

Atributos relevantes da tabela: Classe

Nome do atributo	Tipo de Dados	Descrição
ID_Class	Numérico	Código interno único atributo a cada class. É um número de 2 dígitos gerado sequencialmente. Ex. 14
Nome_Class	12 Caracteres	Nome do classe, tendo no máximo 12 caracteres. Ex. Ranger
TipoDano	10 Caracteres	Tipo do dano da classe. Ex. Fisico
TipoAtaque	10 Caracteres	Tipo do ataque da classe. Ex. Longe

Restrições dos atributos da tabela: CLASSE

Atributo(s)	Nome da Restrição	Tipo de Restrição	Observações
ID_Class	pk_class	chave primária	PRIMARY KEY
Nome_Class	SYS_C00243261	Verificação	CHECK "NOME_CLASS" IS NOT NULL
TipoDano	SYS_C00243262	Verificação	CHECK "TIPODANO" IS NOT NULL
TipoAtaque	SYS_C00243263	Verificação	CHECK "TIPOATAQUE" IS NOT NULL
Nome_Class	SYS_C00243265	Único	UNIQUE NOME_CLASS

• Tabela Clan

A tabela clã representa a informação relativa aos clãs das personagens. É inserido um novo clan nesta tabela, sempre que uma personagem criar um clã novo.

Atributos relevantes da tabela: Clan

Nome do atributo	Tipo de Dados	Descrição
ID_Clan	Numérico	Código interno único atributo a cada clan. É um número de 4 dígitos gerado sequencialmente. Ex. 9224
Nome_Clan	12 Caracteres	Nome do clan, tendo no máximo 12 caracteres. Ex. osmachomens

Restrições dos atributos da tabela: CLAN

Atributo(s)	Nome da Restrição	Tipo de Restrição	Observações
ID_Clan	PK_Clan	chave primária	PRIMARY KEY

Nome_Clan	SYS_C00243265	Único	UNIQUE NOME_CLASS
Nome_Clan	SYS_C00243282	Verificação	CHECK "NOME_CLAN" IS NOT NULL
Nome_Clan	SYS_C00243284	Único	UNIQUE NOME_CLAN

• Tabela Pertence

A tabela pertence representa a informação relativa às personagens que pertencem a um clã.

Atributos relevantes da tabela: Pertence

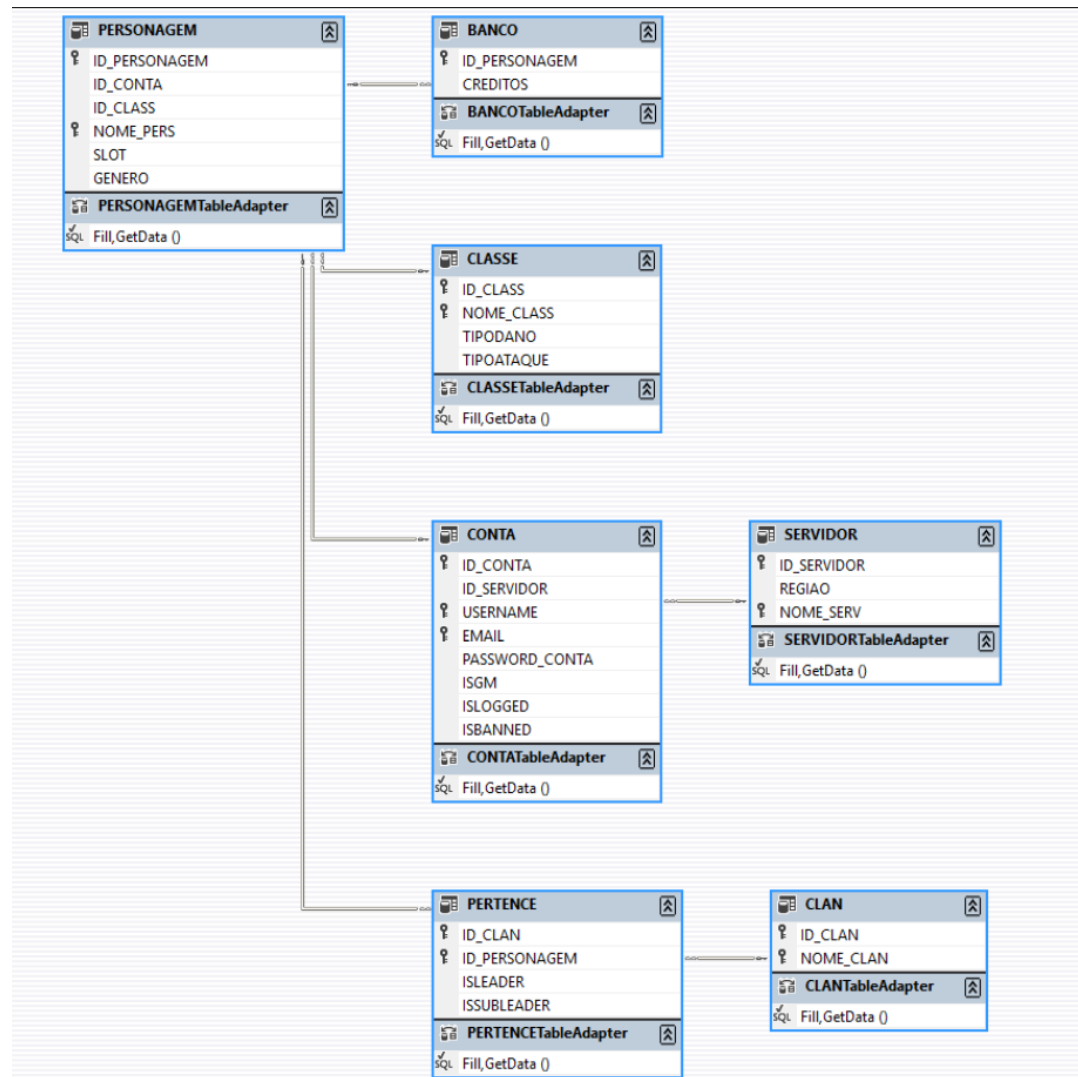
Nome do atributo	Tipo de Dados	Descrição
ID_Clan	Numérico	Código interno único atributo a cada clan. Ex. 14
ID_Personagem	Numérico	Código interno único atributo a cada personagem. Ex. 1
IsLeader	Numérico (1 dígito)	Valor que indica se a personagem é leader. Este valor só pode ser 0 ou 1. Ex. 1
IsSubleader	Numérico (1 dígito)	Valor que indica se a personagem é subleader. Este valor só pode ser 0 ou 1. Ex. 1

Restrições dos atributos da tabela: PERTENCE

Atributo(s)	Nome da Restrição	Tipo de Restrição	Observações
ID_Clan	PK_Pertence	chave primária	PRIMARY KEY
ID_Personagem	PK_Pertence	chave primária	PRIMARY KEY
ID_Clan	SYS_C00243285	Verificação	CHECK "ID_CLAN IS NOT NULL
ID_Personagem	SYS_C00243286	Verificação	CHECK "ID_PERSONAGEM" IS NOT NULL
IsLeader	SYS_C00243287	Verificação	CHECK "ISLEADER" IS NOT NULL
IsSubleader	SYS_C00243288	Verificação	CHECK "ISSUBLEADER" IS NOT NULL
IsLeader	SYS_C00243289	Verificação	CHECK ISLEADER = 0 OR ISLEADER = 1
IsSubleader	SYS_C00243290	Verificação	CHECK ISSUBLEADER = 0 OR ISSUBLEADER = 1

4.2. Diagrama de Tabelas

O modelo Físico (ou diagrama de tabelas) completo com todas as tabelas é o seguinte:



4.3. Script de criação da Base de Dados

```
-- Table Servidor
DROP TABLE Servidor;

CREATE TABLE Servidor (
    ID_Servidor NUMBER,
    Regiao VARCHAR2(3) NOT NULL,
    Nome_Serv VARCHAR2(15) NOT NULL UNIQUE,
    CONSTRAINT "PK_Serv" PRIMARY KEY(ID_Servidor)
);

-- Table Conta
DROP TABLE Conta;

CREATE TABLE Conta (
    ID_Conta NUMBER,
    ID_Servidor NUMBER NOT NULL UNIQUE,
    Username VARCHAR2(12) NOT NULL UNIQUE,
    Email VARCHAR2(50) NOT NULL UNIQUE,
    Password_Conta VARCHAR(15) NOT NULL,
    isGM NUMBER(1) NOT NULL CHECK(isGM = 0 OR isGM = 1),
    isLogged NUMBER(1) NOT NULL CHECK(isLogged = 0 OR isLogged = 1),
    isBanned NUMBER(1) NOT NULL CHECK(isBanned = 0 OR isBanned = 1),
    CONSTRAINT "PK_Conta" PRIMARY KEY(ID_Conta),
    CONSTRAINT "FK_Serv_Conta" FOREIGN KEY(ID_Servidor) REFERENCES
Servidor(ID_Servidor)
);

-- Table Personagem
DROP TABLE Personagem;

CREATE TABLE Personagem (
    ID_Personagem NUMBER,
    ID_Conta NUMBER NOT NULL UNIQUE,
    ID_Class NUMBER NOT NULL UNIQUE,
    Nome_Pers VARCHAR2(15) NOT NULL UNIQUE,
    Slot NUMBER(1) NOT NULL CHECK(Slot >= 1 AND Slot <= 5),
    Genero CHAR(1) NOT NULL CHECK(Genero = 'M' OR Genero = 'F'),
    CONSTRAINT "PK_Pers" PRIMARY KEY(ID_Personagem),
    CONSTRAINT "FK_Conta_Pers" FOREIGN KEY(ID_Conta) REFERENCES
Conta(ID_Conta),
    CONSTRAINT "FK_Class_Pers" FOREIGN KEY(ID_Class) REFERENCES
Classe(ID_Class)
);

-- Table Banco
```



```

DROP TABLE Banco;

CREATE TABLE Banco (
    ID_Personagem NUMBER,
    Creditos NUMBER(7) NOT NULL,
    CONSTRAINT "PK_Banco" PRIMARY KEY(ID_Personagem),
    CONSTRAINT "FK_Pers_Banco" FOREIGN KEY(ID_Personagem) REFERENCES
Personagem(ID_Personagem)
);

-- Table Classe
DROP TABLE Classe;

CREATE TABLE Classe (
    ID_Class NUMBER,
    Nome_Class VARCHAR2(12) NOT NULL UNIQUE,
    TipoDano VARCHAR2(10) NOT NULL,
    TipoAtaque VARCHAR2(10) NOT NULL,
    CONSTRAINT "PK_Class" PRIMARY KEY (ID_Class)
);

-- Table Clan
DROP TABLE Clan;

CREATE TABLE Clan (
    ID_Clan NUMBER,
    Nome_Clan VARCHAR2(12) NOT NULL UNIQUE,
    CONSTRAINT "PK_Clan" PRIMARY KEY(ID_Clan)
);

-- Table Pertence
DROP TABLE Pertence;

CREATE TABLE Pertence (
    ID_Clan NUMBER NOT NULL,
    ID_Personagem NUMBER NOT NULL,
    isLeader NUMBER(1) NOT NULL CHECK(isLeader = 0 OR isLeader =
1),
    isSubLeader NUMBER(1) NOT NULL CHECK(isSubLeader = 0 OR
isSubLeader = 1),
    CONSTRAINT "PK_Pertence" PRIMARY KEY(ID_Clan, ID_Personagem),
    CONSTRAINT "FK_Pert_Clan" FOREIGN KEY(ID_Clan) REFERENCES
Clan(ID_Clan),
    CONSTRAINT "FK_Pert_Pers" FOREIGN KEY(ID_Personagem) REFERENCES
Personagem(ID_Personagem)
);

-- Introduzir valores defeito

```

```
CREATE SEQUENCE SEQ_PK_Serv;

CREATE SEQUENCE SEQ_PK_Conta;

CREATE SEQUENCE SEQ_PK_Personagem;

CREATE SEQUENCE SEQ_PK_Class;

INSERT INTO Servidor
VALUES (SEQ_PK_Serv.NEXTVAL, 'PT', 'Leão');

INSERT INTO Conta
VALUES (SEQ_PK_Conta.NEXTVAL, SEQ_PK_Serv.CURRVAL, 'Shrekstrator',
'shrekovski@shreki.pt', 'adminshreky', 1, 0, 0);

INSERT INTO Personagem
VALUES (SEQ_PK_Personagem.NEXTVAL, SEQ_PK_Conta.CURRVAL,
SEQ_PK_Class.CURRVAL, 'Leandro', 3, 'M');

INSERT INTO Banco
VALUES (SEQ_PK_Personagem.CURRVAL, 9999999);

INSERT INTO Classe
VALUES (SEQ_PK_Class.NEXTVAL, 'Warrior', 'Physical', 'Melee'); add
constraint FK_VENDA_ID_LIVRO_LIVROS
foreign key (CODIGO_LIVRO) references LIVROS (CODIGO_LIVRO);

alter table VENDAS
add constraint FK_VENDA_ID_CLIENTE_CLIENTES
foreign key (CODIGO_CLIENTE) references CLIENTES (CODIGO_CLIENTE);
```

5. Pesquisas SQL

Serve para ir buscar o ID da conta.

```
SELECT ID_CONTA
FROM CONTA, SERVIDOR
WHERE PASSWORD_CONTA = 'this.textBox5.Text'
AND USERNAME = 'this.textBox4.Text'
AND CONTA.ID_SERVIDOR = SERVIDOR.ID_SERVIDOR;
```

Serve para saber se a conta está banida.

```
SELECT ISGM
FROM CONTA
WHERE ID_CONTA = this.LoginID;
```

6. Conclusões

Uma boa construção da base de dados, é fundamental para o bom funcionamento de qualquer empresa ou até mesmo de um jogo. As bases de dados fazem parte do dia-a-dia de milhões de pessoas, que em diferentes situações utilizam o acesso à base de dados.

O objetivo deste trabalho é a aplicação de todos os conhecimentos adquiridos ao longo do semestre da disciplina de base de dados.

Desde a análise da situação inicial do jogo, passando pelo diagrama entidade relacionamento, pela concretização para a base de dados até à implementação dos módulos, todas essas etapas lecionadas nas aulas e colocadas em prática com a realização deste projeto final.

Independentemente deste projeto ser no âmbito escolar, o principal objetivo é adquirir a máxima proximidade e semelhança entre este projeto escolar e o mundo do mercado de trabalho.