Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores

Trabalho Prático Fase 1

47283 : Ricardo Duarte Cardoso Bernardino (a47283@alunos.isel.pt)

47249 : Miguel Henriques Couto de Almeida (a47249@alunos.isel.pt)

Relatório para a Unidade Curricular de Sistemas de Informação
da Licenciatura em Engenharia Informática e de Computadores

Professor : Doutor Nuno Miguel da Costa de Sousa Leite

18 – 3 – 2023

Resumo

No âmbito da primeira fase do trabalho prático da cadeira, este relatório tem como propósito a elaboração de uma base de dados para o problema do enunciado da primeira fase.

Tendo em vista numa fase posterior uma implementação da mesma plataforma proposta.

Através do trabalho produzido ao longo desta fase pretendemos demonstrar conhecimento sobre Transações, Níveis de Isolamento, Funções, Procedimentos e Gatilhos em *Postgres* através da forma como usamos estas ferramentas para resolver os problemas propostos no enunciado.

Este relatório parte do pressuposto que haverá acesso por parte do leitor ao código desenvolvido, não sendo portanto necessário enunciá-lo extensivamente, somente apenas referenciar porções do mesmo na medida do necessário.

Abstract

Within the scope of the first phase of the subject's practical work, this report aims to create a database for the problem of in the document for the first phase.

With the intention for a further implementation of the same proposed platform at a later stage.

Through the work produced throughout this phase, we intend to demonstrate knowledge about transactions and isolation levels, through the way we use these tools to solve the proposed problems.

This report assumes that the reader will have access to the developed code, therefore it is not necessary to enunciate it extensively, only to reference portions of it as necessary.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

SQL Structured Query Language. 1, i
PK Primary Key. 3
FK Foreign Key. 3
EA Modelo Entidade-Associação. 4, i
ER Modelo Relacional. 4

Glossário

NULL Valor nulo, ausência de valor. i

1

Introdução

1.1 Procedimentos

[1] Procedimentos armazenados são blocos de código Structured Query Language (SQL) que são armazenados na base de dados e que podem ser executados sempre que for necessário. Eles são úteis para realizar tarefas complexas ou repetitivas que precisam de ser executadas múltiplas vezes.

1.2 Funções

Funções são blocos de código que executam uma ou mais instruções Structured Query Language (SQL) e podem receber argumentos de entrada. São semelhantes aos procedimentos armazenados, mas diferem na forma como são chamadas e retornam valores.

1.3 Vistas

Vistas são queries Structured Query Language (SQL) armazenadas como objetos na base de dados. Elas permitem que os utilizadores vejam os dados de uma tabela virtual, podendo esta ser uma representação de uma ou mais tabelas da base de dados.

1.4 Gatilhos

Gatilhos (ou triggers) são procedimentos armazenados que são executados automaticamente depois de um evento específico numa tabela, como a inserção, atualização ou a remoção de um tuplo.

1.5 Níveis de Isolamento

De forma a manter a integridade transacional, assim como a consistência da DB [3], foi necessário atribuir a cada função e procedimento o nível de isolamento adequado.

Assim, consoante as operações que queríamos realizar, foi atribuído um dos seguintes níveis de isolamento:

- Serializable - que garante uma execução em série das transações e previne anomalias como phantom-reads.
- Repeatable Read - que bloqueia tanto leituras como escritas de outras transações, sobre todos os tuplos referenciados, prevenindo assim a ocorrência da anomalia dos non-repeatable reads.
- Read Committed - que garante a leitura da informação anteriormente committed, prevenindo anomalias do tipo dirty read;

2

Análise do Enunciado

2.1 Caso em Estudo

No presente trabalho é nos proposto o desenvolvimento de um sistema de dados para uma empresa chamada "GameOn". O propósito deste sistema é registar múltiplos dados relacionados a utilizadores, jogos, partidas, e também estatísticas destes. Os utilizadores têm a possibilidade de comprar Jogos, e participar em partidas destes. As partidas podem ser "normais"(Single-Player) ou "multi-jogador"(Multiplayer). Estas partidas podem ter vários estados, e contém também uma tabela de pontuações indicando a pontuação que cada jogador participante obteve.

2.2 Modelo Entidade Associação

[2] Realizamos este processo seguindo as regras presentes no livro [2] *Fundamentals of Database System*. Para a identificação das chaves sublinhamos apenas para identificar que se trata de uma Primary Key (PK) e identificamos explicitamente as Foreign Key (FK).

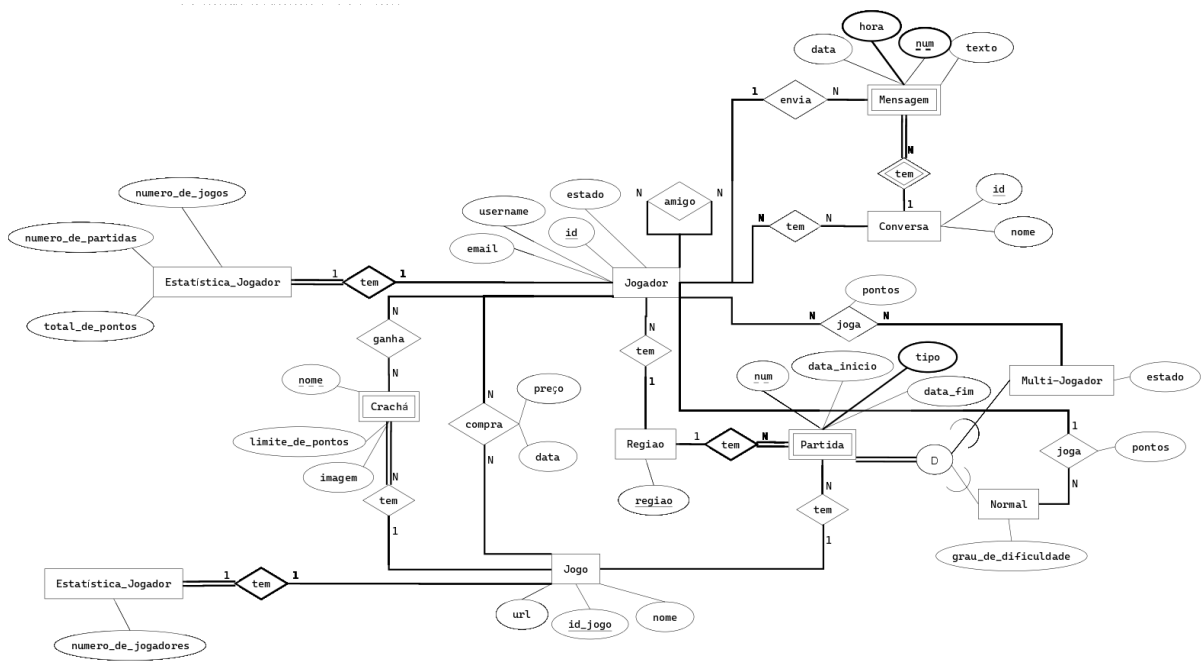

Figura 2.1: Diagrama EA

2.3 Modelo Entidade Relação

Realizamos nesta fase do trabalho a passagem do Modelo Entidade-Associação (EA) mostrado em cima para o Modelo Relacional (ER) seguindo as regras presentes no livro [2] *Fundamentals of Database System* e o resultado final desta fase pode ser consultado na secção dos anexos deste relatório.

3

Implementação

3.1 Construção do Modelo Físico

Nesta fase do trabalho o grupo passou para a fase de implementação em Postgres dos requisitos do enunciado.

3.1.1 Criar o Modelo Físico

Criámos um script (create.sql) que construísse o modelo físico da base de dados seguindo as tabelas do Modelo Relacional construído em cima aplicando então nos atributos as restrições de integridade que achámos necessárias e também para certos atributos criando domínios como para o url, email e a chave alfanumérica.

3.1.2 Remoção do Modelo Físico

Criámos um script que removesse o modelo físico (remove.sql), ou seja, executando o comando drop table para todas as tabelas criadas anteriormente.

3.1.3 Preenchimento Inicial da Base de Dados

Construímos também um script que preenchesse a base de dados com conteúdo inicial (insert.sql) que poderia ser eventualmente usado nos testes ou para simplesmente para visualizar o comportamento aplicacional das tabelas que tinham sido criadas.

3.2 Funções, Procedimentos, Vistas e Gatilhos

3.2.1 Procedimento Criar Jogador

Neste procedimento é feita uma averiguação inicial à integridade dos parâmetros e logo de seguida é feito o insert na tabela do jogador com os valores recebidos. Como estamos diante de uma inserção numa tabela a anomalia a ter em conta seria um dirty write podendo resultar num lost update portanto é recomendável que este procedimento seja executado dentro de uma transação com isolamento superior ou igual a repeatable read.

3.2.2 Procedimento Jogador Inativo

Neste procedimento é feita uma averiguação inicial que averigua se o jogador existe e logo de seguida é executado o update na tabela do jogador alterando o atributo estado para Inativo. Como estamos diante de um update numa tabela a anomalia a ter em conta seria um dirty write podendo resultar num lost update portanto é recomendável que este procedimento seja executado dentro de uma transação com isolamento superior ou igual a repeatable read.

3.2.3 Procedimento Jogador Banido

Neste procedimento é feita uma averiguação inicial que averigua se o jogador existe e logo de seguida é executado o update na tabela do jogador alterando o atributo estado para Banido. Como estamos diante de um update numa tabela a anomalia a ter em conta seria um dirty write podendo resultar num lost update portanto é recomendável que este procedimento seja executado dentro de uma transação com isolamento superior ou igual a repeatable read.

3.2.4 Função totalPontosJogador

Nesta função é feita a soma das pontuações de um dado jogador tanto em partidas normais como em multi-jogador e retornado o valor resultante na forma de um inteiro. Como estamos diante de uma leitura em múltiplas tabelas a anomalia a ter em conta seria um dirty read podendo assim ser retornado um valor que não seria o expectado portanto é recomendado que este procedimento seja executado dentro de uma transação com isolamento superior ou igual a read committed.

3.2.5 Função totalJogosJogador

Nesta função é feita a soma do número de pontuações de um dado jogador tanto em partidas normais como em multi-jogador e retornado o valor resultante na forma de um inteiro. Como estamos diante de uma leitura em múltiplas tabelas a anomalia a ter em conta seria um dirty read podendo assim ser retornado um valor que não seria o expectado portanto é recomendado que este procedimento seja executado dentro de uma transação com isolamento superior ou igual a read committed.

3.2.6 Função PontosJogoPorJogador

Nesta função é retornada a tabela com as pontuações dos jogadores numa dada partida. Como estamos diante de uma leitura numa tabela a anomalia a ter em conta seria um dirty read podendo assim ser retornado um valor que não seria o expectado portanto é recomendado que este procedimento seja executado dentro de uma transação com isolamento superior ou igual a read committed.

3.2.7 Procedimento associarCrachá

Neste procedimento é feita uma série de averiguações iniciais que validam se a operação pode ser executada ou não e no final é feito o insert na tabela crachaJogador com os valores recebidos. Como estamos diante de uma inserção numa tabela a anomalia a ter em conta seria um dirty write podendo resultar num lost update portanto é recomendável que este procedimento seja executado dentro de uma transação com isolamento superior ou igual a repeatable read.

3.2.8 Procedimento iniciarConversa

Neste procedimento é feita uma série de inserções para garantir o comportamento expectado pelo procedimento, desde a inserção de uma nova conversa, passando pela junção à mesma conversa e acabando com o envio de uma mensagem para a mesma conversa sinalizando que a conversa foi criada. Como estamos diante de uma inserção em múltiplas tabelas a anomalia a ter em conta seria um dirty write podendo resultar num lost update portanto é recomendável que este procedimento seja executado dentro de uma transação com isolamento superior ou igual a repeatable read.

3.2.9 Procedimento juntarConversa

Neste procedimento é feita uma série de inserções novamento para garantir o comportamento expectado pelo procedimento, começando pela junção à conversa e de seguida

o envio de uma mensagem para a mesma conversa sinalizando que o utilizador acabou de entrar. Como estamos diante de uma inserção em múltiplas tabelas a anomalia a ter em conta seria um dirty write podendo resultar num lost update portanto é recomendável que este procedimento seja executado dentro de uma transação com isolamento superior ou igual a repeatable read.

3.2.10 Procedimento enviarMensagem

Neste procedimento é feita uma inserção na tabela da Mensagem sinalizando o envio de uma mensagem para conversa passada como parâmetro. Como estamos diante de uma inserção numa tabela a anomalia a ter em conta seria um dirty write podendo resultar num lost update portanto é recomendável que este procedimento seja executado dentro de uma transação com isolamento superior ou igual a repeatable read.

3.2.11 Vista jogadorTotalInfo

Nesta vista é feita uma leitura e seleção do conteúdo de múltiplas tabelas com base no id dos jogadores da tabela Jogador.

3.2.12 Trigger de Atribuição de Crachás

Este trigger é acionado depois de um update na coluna data_fim da tabela Partida executando a função que é responsável por atribuir os crachás aos jogadores que participaram na mesma partida consoante a respetiva pontuação de cada jogador.

3.2.13 Trigger sobre o DELETE na vista jogadorTotalInfo

Este trigger é acionado quando há ocorre um delete sobre a vista jogadorTotalInfo executando a função que é responsável por banir os jogadores que estiverem presentes na mesma vista.

3.3 Testes

Para atestar a qualidade das alíneas anteriores realizámos uns testes. Dividimos os testes pelo nível de isolamento em que gostaríamos que corressem resultando assim três sequências de testes.

A sequência de testes que corre no nível de isolamento Read Committed têm como intuito averiguar que as leituras nos procedimentos ocorrem sem problemas, não sendo necessário portanto elevar o nível de isolamento para tal efeito.

Na sequência Repeatable Read queríamos que fosse dada a garantia que os tuplos das tabelas que estavam a ser alterados não fossem também alterados por outras procedures ou por outras transações a ocorrer ao mesmo tempo.

Por último, na sequência Serializable queríamos que fosse dada a garantia que os procedures executassem na ordem descrita para garantir a correta simulação do comportamento aplicacional pretendido com o teste, logo foi escolhido o nível do isolamento mais restrito, o serializable.

Referências

[1] S. Sudarshan Abraham Silberschatz Henry F. Korth, *Database System Concepts 7th ed.* McGrawHill Education, 2020.

[2] Shamkant B. Navathe R. Elmasri, *Fundamentals of Database System 7th ed.* Pearson Education, 2016.

[3] Andreas Reuter Jim Gray, *Transaction Processing: Concepts and Techniques 5th ed.* Morgan Kaifmann, 1993.

Anexo 1

No anexo a baixo irá encontrar o diagrama que representa o modelo EA e as respetivas tabelas que dão lugar na íntegra ao modelo relacional obtido acompanhado das restrições de integridade.

A.1 Modelo Relacional

As relações são organizadas de acordo com a ordem lógica do enunciado, com o objetivo de facilitar a compreensão do modelo relacional. Cada atributo é enumerado juntamente com o seu domínio e restrições, o que é importante para posteriormente criar uma base de dados em SQL.

Se um atributo tiver a restrição de NULL, significa que pode aceitar um valor nulo.

Relação REGIAO

Nome	Tipo	Restrições
nome	string	

Tabela A.1: Relação REGIAO

Relação JOGADOR

Nome	Tipo	Restrições
id	int	
username	string	
email	string	no formato válido
estado	string	"Ativo", "Inativo", "Banido"
regiao	string	

Tabela A.2: Relação JOGADOR

Relação AMIZADE

Nome	Tipo	Restrições
id_jogador1	int	
id_jogador2	int	

Tabela A.3: Relação AMIZADE

Relação CONVERSA

Nome	Tipo	Restrições
id	int	
nome	string	

Tabela A.4: Relação CONVERSA

Relação CONVERSAMEMBROS

Nome	Tipo	Restrições
id_conversa	int	
id_jogador	int	

Tabela A.5: Relação CONVERSAMEMBROS

Relação MENSAGEM

Nome	Tipo	Restrições
num	int	
id_conversa	int	
data	date	no formato "DD-MM-AAAA"
hora	time	
texto	string	
id_jogador	int	

Tabela A.6: Relação MENSAGEM

Relação JOGO

Nome	Tipo	Restrições
id_jogo	string	
nome	string	
url	string	no formato válido

Tabela A.7: Relação JOGO

Relação COMPRA

Nome	Tipo	Restrições
id_jogo	string	
id_jogador	int	
data	date	no formato "DD-MM-AAAA"
preco	int	preco > 0

Tabela A.8: Relação COMPRA

Relação PARTIDA

Nome	Tipo	Restrições
num	int	
ref_jogo	string	
data_inicio	date	no formato "DD-MM-AAAA"
data_fim	date	no formato "DD-MM-AAAA» data_inicio
tipo	string	"Normal"ou "Multi-Jogador"
regiao	string	

Tabela A.9: Relação PARTIDA

Relação PARTIDA_NORMAL

Nome	Tipo	Restrições
num	int	
ref_jogo	string	
grau_de_dificuldade	string	

Tabela A.10: Relação PARTIDA_NORMAL

Relação PARTIDA_MULTI_JOGADOR

Nome	Tipo	Restrições
num	int	
ref_jogo	string	
estado	string	"Por Iniciar", "A Aguardar Jogadores", "Em Curso", "Terminada"

Tabela A.11: Relação PARTIDA_MULTI_JOGADOR

Relação PONTUACOES_NORMAL

Nome	Tipo	Restrições
num_partida	int	
ref_jogo	string	
id_jogador	int	
pontos	int	pontos >= 0

Tabela A.12: Relação PONTUACOES_NORMAL

Relação PONTUACOES_MULTI_JOGADOR

Nome	Tipo	Restrições
num_partida	int	
ref_jogo	string	
id_jogador	int	
pontos	int	pontos >= 0

Tabela A.13: Relação PONTUACOES_MULTI_JOGADOR

Relação CRACHA

Nome	Tipo	Restrições
nome	string	
ref_jogo	string	
limite_de_pontos	int	limite_de_pontos > 0
imagem	string	

Tabela A.14: Relação CRACHA

Relação CRACHAJOGADOR

Nome	Tipo	Restrições
nome	string	
id_jogador	int	

Tabela A.15: Relação CRACHAJOGADOR

Relação ESTATISTICA_JOGADOR

Nome	Tipo	Restrições
id_jogador	int	
numero_de_jogos	int	numero_de_jogos >= 0
numero_de_partidas	int	numero_de_partidas >= 0
numero_total_de_pontos	int	numero_total_de_pontos >= 0

Tabela A.16: Relação ESTATISTICA_JOGADOR

Relação ESTATISTICA_JOGO

Nome	Tipo	Restrições
ref	string	
numero_de_partidas	int	numero_de_partidas >= 0
numero_de_jogadores	int	numero_de_jogadores >= 0
total_de_pontos	int	total_de_pontos >= 0

Tabela A.17: Relação ESTATISTICA_JOGO

A.2 Modelo Entidade-Associação

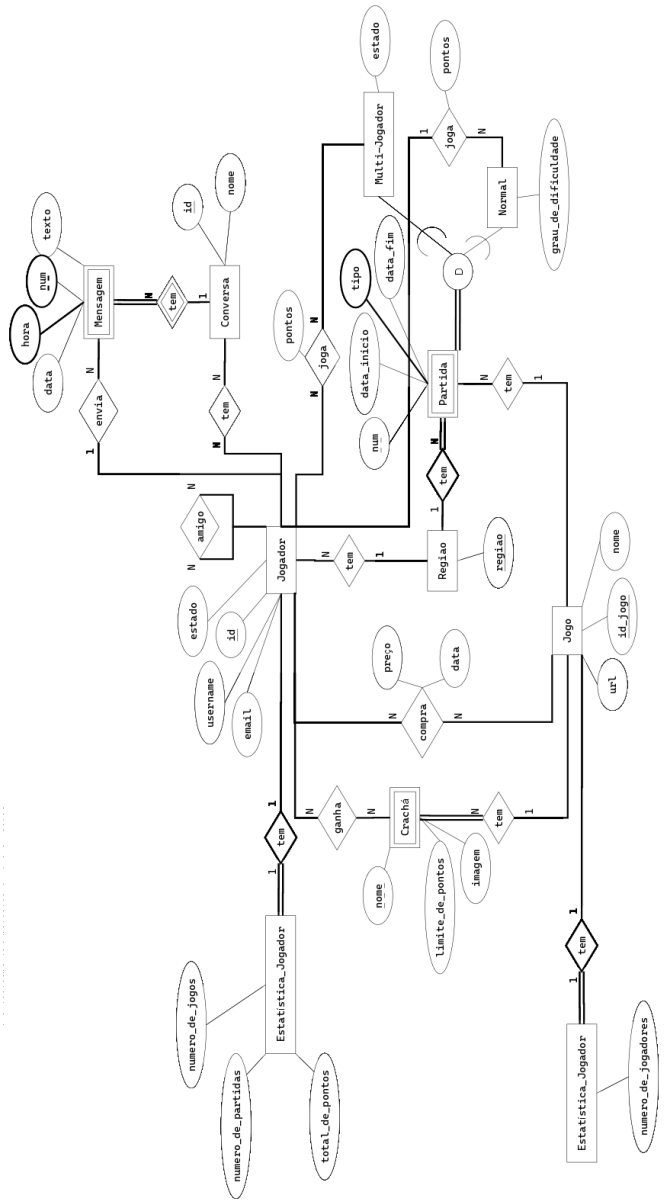

Figura A.1: Diagrama EA full page