

2ª SUBMISSÃO: DESENVOLVIMENTO DO MODELO RELACIONAL, CRIAÇÃO E POPULAÇÃO DA BASE DE DADOS

Unidade Curricular: Bases de Dados (LEIC012)

Ano Letivo: 2024/2025

Professor da Componente Prática: Michel Celestino Paiva
Ferreira

Turma: 2LEIC13

Grupo: 1307

Estudantes e Autores:

Gonçalo França up202305533@fe.up.pt

Sofia Cruz up202208135@fe.up.pt

Tomás Sousa up202303524@fe.up.pt

1. Índice

1. Índice.....	2
2. Introdução.....	3
3. Definição de Domínio.....	4
4. Modelo Conceitual Inicial.....	5
5. Modelo Conceitual Final com auxílio de IA.....	5
6. Refinar o Modelo Conceitual.....	7
7. Modelo Relacional Inicial.....	7
8. Modelo Relacional Final após introdução de IA geradora.....	8
9. Análise Inicial das Dependências Funcionais e Formas Normais.....	10
10. Análise Final com base em ferramentas de IA.....	12
11. Criação da Base de Dados em SQLite.....	13
12. População da Base de Dados.....	14
13. Avaliação da ferramenta de IA utilizada e membros do grupo.....	15
14. Conclusão.....	16

2. Introdução

No âmbito da disciplina de bases de dados, foi-nos proposta a realização de um trabalho que consistia no desenvolvimento de uma base de dados sobre um tema à nossa escolha, seguindo os passos necessários para a criação da mesma. Na primeira submissão tínhamos como objetivo a criação de um modelo conceitual enquanto que, na segunda tínhamos que, partindo do feedback dado pelo professor sobre o nosso modelo conceitual, corrigir este mesmo modelo, transformá-lo num modelo relacional e, por fim, criar e popular uma base de dados a partir dos dois modelos referidos anteriormente.

Também é importante frisar que, em todos os momentos que nos foi requisitada a utilização de uma ferramenta de IA para auxílio em determinadas tarefas, fizemos uso de “Chat GPT”, que consideramos como sendo a mais capaz dentro das que temos conhecimento.

3. Definição de Domínio

O domínio deste projeto envolve uma empresa de gestão de eventos, responsável pela organização de diferentes tipos de eventos, como casamentos, batizados, festivais e bailes. A empresa atua como uma intermediária entre clientes e fornecedores, assegurando que todos os elementos necessários para o sucesso de um evento estejam devidamente coordenados. O objetivo da base de dados desenvolvida neste projeto é armazenar e organizar informações relacionadas aos clientes, eventos, fornecedores, locais e serviços, permitindo à empresa gerir com eficiência o processo de planeamento e execução de eventos.

Os eventos são a entidade central do sistema, e cada um pode ser descrito por dados como data e hora, e número de convidados. Os eventos podem ser classificados, em casamentos, batizados, festivais, bailes, ou de outro tipo, permitindo que o sistema trate cada tipo de forma diferente, conforme as características associadas. O espaço está diretamente associado ao evento, pois cada evento precisa de um local onde será realizado, que vá de encontro às necessidades, como área e localização. A época também está vinculada ao evento, pois os preços podem ser ajustados conforme a época do ano que são realizados.

As empresas desempenham um papel essencial no sistema. Cada empresa é identificada pelo nome, localização, data de fundação, NIF, e meios de contacto (número de telefone e email). São as empresas, as principais responsáveis pelo planeamento e execução dos eventos, garantindo a entrega dos serviços de acordo com os requisitos dos clientes. Estes requisitos correspondem ao preço e ao horário de contratação do serviço. O sistema gerencia também os fornecedores, um tipo de empresas, responsáveis pelo fornecimento de materiais e serviços que complementam o evento. Estas empresas são compostas por funcionários, responsáveis por executar os serviços contratados. Sobre eles, o sistema armazena informações essenciais como nome, data de nascimento, morada, NIF e número de telefone.

Os clientes são aqueles para os quais os eventos são organizados. Para cada cliente, são registadas informações básicas como nome, morada, NIF e meios de contacto (número de telefone e email).

O pagamento, quer seja o pagamento aos fornecedores quer o pagamento por parte do cliente, é essencial para acompanhar os custos associados ao evento. Este pode ser feito de diversos meios, que pode incluir cheque, dinheiro, transferência bancária, entre outros, permitindo assim a flexibilidade na gestão financeira de cada evento. Além do método de pagamento, o sistema ainda armazena o montante a pagar, e a data de quando o mesmo foi feito, e no caso do cliente o prazo de liquidação.

4. Modelo Conceitual Inicial

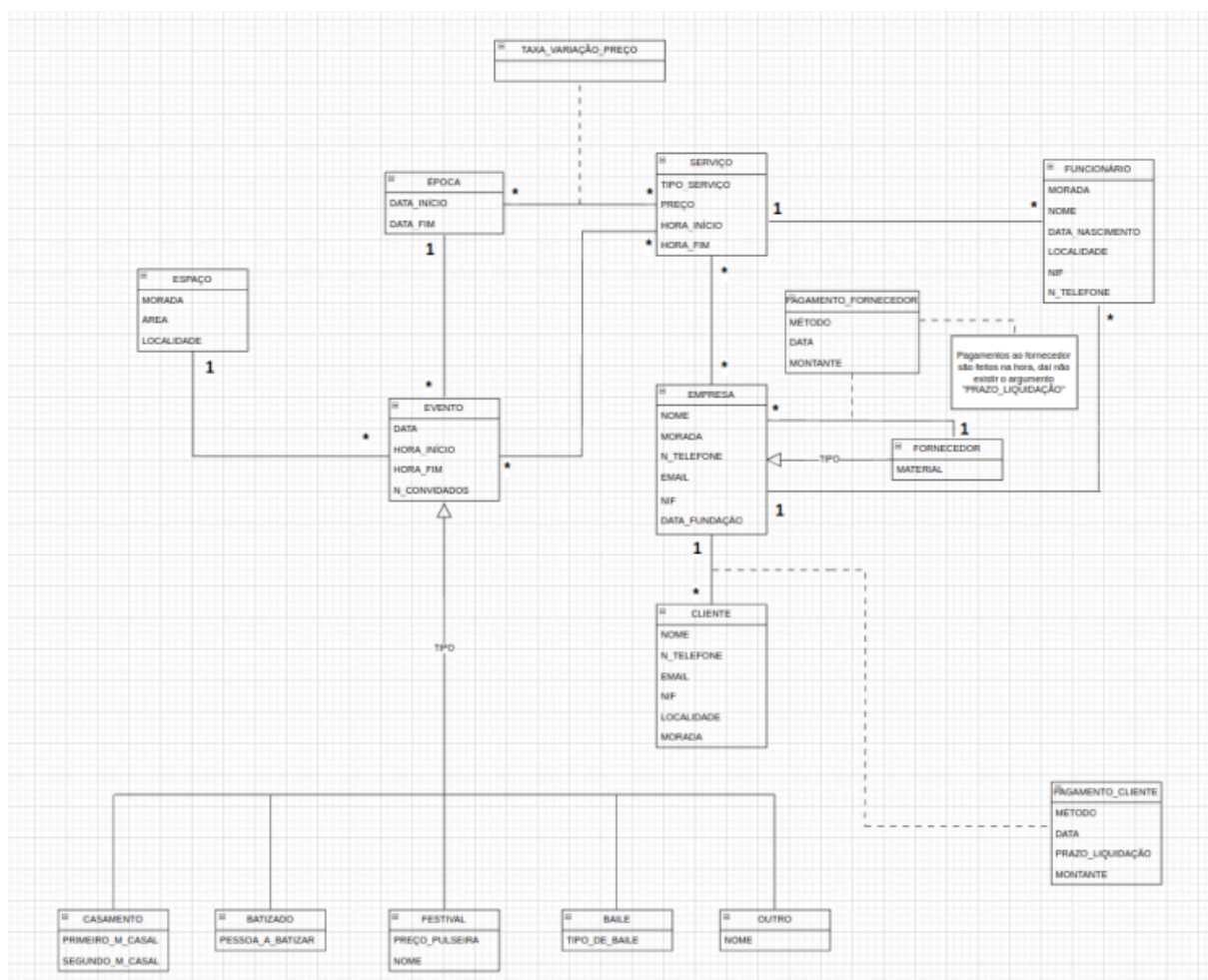


Fig. 1 - Modelo Conceitual Inicial

5. Modelo Conceitual Final com auxílio de IA

A fim de melhorar o UML previamente proposto, implementámos uma ligação direta entre “EVENTO” e “CLIENTE”, garantindo que cada evento só pode ter um

cliente associado. Esta alteração torna o modelo mais claro e facilita a gestão de quem organiza ou participa nos eventos.

Adicionamos também uma nota a indicar que os eventos não podem coincidir no mesmo espaço e horário, evitando conflitos e assegurando uma melhor gestão dos recursos.

Atualizamos a nomenclatura dos atributos para “DATA_REGISTO” e “DATA_OCORRENCIA” para maior clareza. Esta distinção permite identificar, respetivamente, quando uma entidade é inserida no sistema e quando um evento ocorre, evitando ambiguidades.

Optámos por não criar uma classe separada de “ENDEREÇO” com os atributos “LOCALIDADE” e “MORADA”, por considerarmos que essa abstração complicaria o modelo sem trazer grandes vantagens. Manter estes atributos diretamente nas entidades envolvidas simplifica a estrutura.

Por fim, decidimos não incluir uma restrição a indicar que os atributos da classe “CASAMENTO” não podem ser nulos, por ser uma regra implícita. A identificação obrigatória dos membros do casal faz parte da própria lógica deste tipo de evento, tornando essa validação desnecessária no modelo.

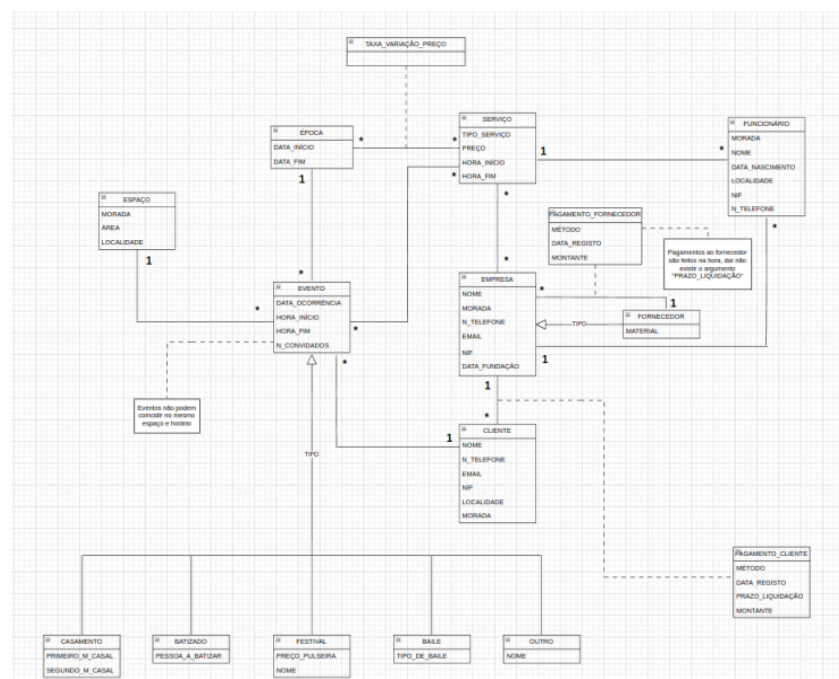


Fig. 2 - Segundo Modelo Conceitual

6. Refinar o Modelo Conceitual

A partir do feedback dado pelo professor após correção da nossa primeira submissão, foi possível destacar um problema principal no nosso modelo conceitual: a existência de uma classe empresa, que se referia à empresa que utilizamos como tema para este trabalho. Foi-nos então sugerido que esta classe fosse removida e que o modelo fosse depois adaptado para que fizesse sentido sem ela.

Após correção, o nosso modelo conceitual final ficou assim:

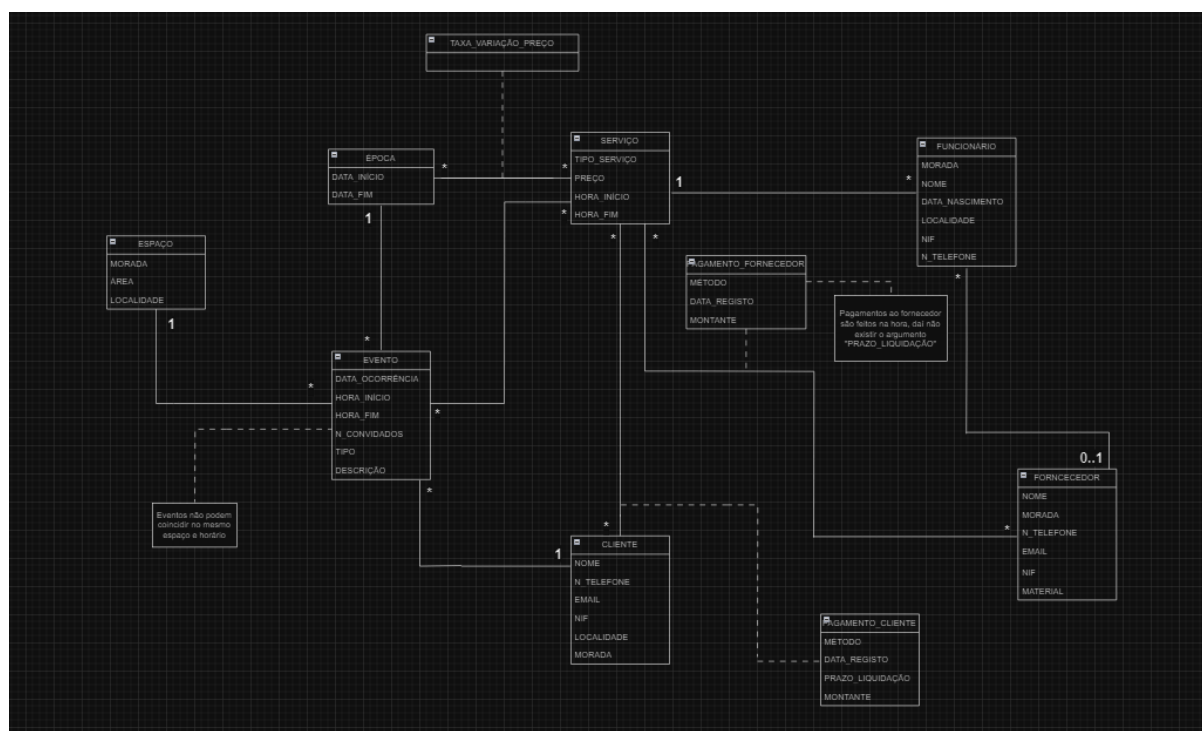


Fig. 3 - Modelo Conceitual Final

7. Modelo Relacional Inicial

Com o modelo conceitual corrigido em mãos, precisávamos agora de o converter para um modelo relacional. Limitamo-nos a traduzir o modelo conceitual e a adicionar atributos “ID” a todas as classes, para que servissem como chave primária, resultando no seguinte modelo conceitual:

Cliente(Cid, NIF, Nome, N_Telefone, Email, Localidade, Morada)

Serviço(Sid, Tipo_Serviço, Preço, Hora_Início, Hora_Fim)

Época(Epid, Data_Início, Data_Fim)

Espaço(Espid, Morada, Área, Localidade)

Evento(Evid, Data_Ocorrência, Hora_Início, Hora_Fim, N_Convidados, Tipo, Descrição, Cid->Cliente, Espid->Espaço, Epid->Época)

Funcionário(Funcid, NIF, Nome, Morada, Data_Nascimento, Localidade, N_Telefone, Sid->Serviço, Fornid->Fornecedor)

Fornecedor(Fornid, NIF, Nome, Morada, N_Telefone, Email, Material)

Pagamento_Fornecedor(Sid->Serviço, Fornid->Fornecedor, Método, Data_Registo, Montante)

Pagamento_Cliente(Sid->Serviço, Cid->Cliente, Método, Data_Registo, Prazo_Liquidação, Montante)

Taxa_Variação_Preço(Sid->Serviço, Epid->Época, Taxa)

Serviço/Evento(Sid->Serviço, Evid->Evento)

8. Modelo Relacional Final após introdução de IA geradora

Para esta parte do trabalho, com o objetivo de obter possíveis sugestões que pudessem melhorar o nosso modelo relacional e de explorar todas as funcionalidades da ferramenta de inteligência artificial decidimos, inicialmente, pedir-lhe que gerasse um esquema relacional com base no nosso esquema conceitual, a fim de verificar o quão parecido ficaria com o nosso e, depois de obter o resultado, introduzimos o nosso esquema relacional e perguntamos o que poderia ser alterado/melhorado no nosso modelo relacional.

Para alcançar os nossos objetivos, introduzimos os seguintes prompts:

- 1º Prompt: *foto do modelo conceitual corrigido*

“Boa tarde. Este foi o modelo conceitual para uma base de dados que eu e os meus colegas de grupo desenvolvemos no âmbito da disciplina de base de dados. Precisamos agora que o converta para um modelo relacional, para que possamos comparar com o modelo que desenvolvemos.”

- 2º Prompt: *Modelo Relacional desenvolvido com base no modelo conceitual*
“Este foi o modelo relacional que nós desenvolvemos. Em algumas classes achamos pertinente definir uma chave primária "ID" para identificar as classes. O que achas que devemos alterar no nosso modelo. Se achares pertinente realizar alguma mudança, realiza-as!”

Após introdução dos prompts, foram geradas diversas sugestões. Entre as que não levamos em conta estão a adição de descrições/atributos adicionais na relação “Serviço/Evento”, a criação de uma associação direta entre evento e funcionário e a correção da redundância do atributo localidade em Cliente e Espaço (que não corrigimos pois não consideramos que existisse qualquer tipo de redundância).

Apesar de não termos levado em conta nenhuma das sugestões anteriores, houve uma alteração que achamos pertinente, trocar o nome do atributo “Taxa” na classe de associação “Taxa_Variação_Preço” para “Percentagem_Variação”, a fim de tornar mais legível o atributo que criamos.

Depois de realizadas as alterações referidas, este foi o resultado do nosso modelo relacional:

Cliente(Cid, NIF, Nome, N_Telefone, Email, Localidade, Morada)

Serviço(Sid, Tipo_Serviço, Preço, Hora_Início, Hora_Fim)

Época(Epid, Data_Início, Data_Fim)

Espaço(Espid, Morada, Área, Localidade)

Evento(Evid, Data_Ocorrência, Hora_Início, Hora_Fim, N_Convidados, Tipo, Descrição, Cid->Cliente, Espid->Espaço, Epid->Época)

Funcionário(Funcid, NIF, Nome, Morada, Data_Nascimento, Localidade, N_Telefone, Sid->Serviço, Fornid->Fornecedor)

Fornecedor(Fornid, NIF, Nome, Morada, N_Telefone, Email, Material)

Pagamento_Fornecedor(Sid->Serviço, Fornid->Fornecedor, Método, Data_Registo, Montante)

Pagamento_Cliente(Sid->Serviço, Cid->Cliente, Método, Data_Registo, Prazo_Liquidação, Montante)

Taxa_Variação_Preço(Sid->Serviço, Epid->Época, Percentagem_Variação)

Serviço/Evento(Sid->Serviço, Evid->Evento)

9. Análise Inicial das Dependências Funcionais e Formas Normais

Nesta etapa, tínhamos como meta realizar a análise das dependências funcionais da nossa base de dados, analisar se estas respeitavam a terceira forma normal e a forma normal de Boyce Codd e, caso isso não acontecesse, efetuar as devidas correções.

Antes de fazer qualquer análise era necessário, em primeiro lugar, destacar as dependências funcionais que achamos que faziam sentido no contexto da nossa base de dados. Chegamos, então, às seguintes:

- Cliente:
Cid -> NIF, Nome, Morada, Localidade, N_Telefone, Email;
NIF -> Nome, Morada;
Morada -> Localidade;
Nome, Morada -> N_Telefone, Email;
- Serviço:
Sid -> Tipo_Serviço, Preço, Hora_Início, Hora_Fim;
- Época:
Epid -> Data_Início, Data_Fim;
- Espaço:
Espid -> Morada, Área, Localidade;
Morada -> Localidade;
- Evento:
Evid -> Data_Ocorrência, Hora_Inicio, Hora_Fim, N_Convidados, Tipo, Descrição, Cid, Espid, Epid;
- Funcionário:
Funcid -> NIF, Nome, Morada, Data_Nascimento, Localidade, N_Telefone, Sid, Fornid;
NIF -> Nome, Morada, Data_Nascimento;
Morada -> Localidade;
- Fornecedor:
Fornid -> NIF, Nome, Morada, N_Telefone, Email, Material;
NIF -> Nome, Morada;
Nome, Morada -> N_Telefone, Email, Material;

- Pagamento_Fornecedor:
Sid, Fornid -> Método, Data_Registo, Montante;
- Pagamento_Cliente:
Sid, Cid -> Método, Data_Registo, Prazo_Liquidação, Montante;
- Taxa_Variação_Preço:
Sid, Epid -> Percentagem_Variação;
- Serviço/Evento:
Não possui dependências funcionais, visto que é apenas uma associação sem atributos que conecta as classes Serviço e Evento;

Depois de observar estas dependências, concluímos que era necessário simplificá-las, devido à existência de dependências triviais que consideramos desnecessárias neste contexto, resultando no seguinte conjunto de dependências:

- Cliente:
Cid -> NIF, Nome, Morada, Localidade, N_Telefone, Email;
- Serviço:
Sid -> Tipo_Serviço, Preço, Hora_Início, Hora_Fim;
- Época:
Epid -> Data_Início, Data_Fim;
- Espaço:
Espid -> Morada, Área, Localidade;
- Evento:
Evid -> Data_Ocorrência, Hora_Inicio, Hora_Fim, N_Convidados, Tipo, Descrição, Cid, Espid, Epid;
- Funcionário:
Funcid -> NIF, Nome, Morada, Data_Nascimento, Localidade, N_Telefone, Sid, Fornid;
- Fornecedor:
Fornid -> NIF, Nome, Morada, N_Telefone, Email, Material;
- Pagamento_Fornecedor:
Sid, Fornid -> Método, Data_Registo, Montante;
- Pagamento_Cliente:

Sid, Cid -> Método, Data_Registo, Prazo_Liquidação, Montante;

- Taxa_Variação_Preço:

Sid, Epid - > Percentagem_Variação;

- Serviço_Evento:

Não possui dependências funcionais, visto que é apenas uma associação sem atributos que conecta as classes Serviço e Evento;

Após obter todas as dependências funcionais, e estando convictos de que estas já estavam simplificadas, fizemos a seguinte análise quanto à terceira forma normal e forma normal de Boyce Codd para todas as classes, visto que acreditamos que todas elas respeitam estas normas:

- Podemos afirmar que as dependências funcionais resultantes de cada uma das classes respeitam a terceira forma normal e a forma normal de Boyce Codd visto que não existem relações triviais e que possuem uma única relação cuja superchave é chave primária e determina todos os outros atributos da classe;

10. Análise Final com base em ferramentas de IA

A fim de verificar se as nossas análises estavam corretas introduzir, na ferramenta de inteligência artificial, o seguinte prompt:

- *dependências funcionais simplificadas*

Estas são as nossas dependências funcionais, já simplificadas, e abaixo tens a nossa justificação para o facto de todas as classes respeitarem a terceira forma normal e a forma normal de boyce codd:

justificação dada no tópico acima

“Achas que a nossa análise e a nossa justificação estão corretas? Se acreditares que existe algum equívoco, sugere algo que possa corrigir o nosso erro!”

A este prompt, a ferramenta respondeu que a nossa análise estava bastante sólida e que nenhuma das nossas dependências funcionais violava alguma das normas. Portanto, assumimos como a nossa análise e justificação finais aquelas que referimos no tópico acima.

11. Criação da Base de Dados em SQLite

Com base no modelo relacional desenvolvido e das dependências analisadas, elaboramos os scripts create1.sql e create2.sql para a criação das tabelas para a base de dados.

O primeiro script, create1.sql, foi construído a partir do modelo relacional final proposto. Este incluía relações com as chaves primárias e estrangeiras identificadas no modelo relacional, restrições de integridade como NOT NULL, CHECK e UNIQUE, aplicadas a vários atributos para garantir a consistência dos dados e tipos de dados básicos, como NUMERIC para IDs e valores numéricos, e VARCHAR para textos.

Com o objetivo de melhorar o modelo, recorremos a uma ferramenta IA para sugerir alterações ao script inicial. Enviando os seguintes prompts:

script inicial do create1.sql

"Nesta etapa do trabalho, o nosso objetivo é criar uma base de dados em sqlite com base no modelo relacional que te enviei na última mensagem. Nós já criamos um, mas gostaríamos de saber como farias. Podes criá-la e enviar-me o ficheiro, por favor?"

Com base nestes inputs, a ferramenta forneceu novas tabelas, que integraram melhorias nos tipos de dados, restrições de integridade, nomenclatura e organização geral.

Após integrar as sugestões da ferramenta de IA, criamos o script final, create2.sql. Neste script final foram corrigidos tipos de dados, em que os IDs foram alterados de NUMERIC para INTEGER, garantindo assim compatibilidade e precisão, valores monetários passaram a utilizar REAL em vez de NUMERIC, datas e horas passaram a utilizar tipos como DATE e TIME, e os atributos NIF e N_TELEFONE passaram de NUMERIC a TEXT. Na tabela EVENTO o CHECK(UNIQUE(DATA_OCCORENCIA, ESPID)) não combinava corretamente CHECK e UNIQUE, levando-nos a corrigir para uma cláusula UNIQUE isolada. Para a validação dos números de telefone foi adicionada também a condição CHECK(LENGTH(...)). A nível de nomenclatura de tabelas e atributos, algumas alterações também foram realizadas, a tabela SERVIÇO/EVENTO utilizava o carácter '/', que não vai de encontro com as convenções SQL, tendo sido renomeada para SERVIÇO_EVENTO. A ferramenta de IA também sugeriu a

renomeação do atributo TAXA na tabela TAXA_VARIAÇÃO_PREÇO, para PERCENTAGEM_VARIAÇÃO, tornando-o assim mais descritivo. Para a validação de datas foi adicionada a restrição CHECK(DATA_FIM > DATA_INICIO) na tabela EPOCA de forma a garantir a consistência nas datas, assim como na tabela FUNCIONARIO, onde foi adicionada a restrição CHECK(DATA_NASCIMENTO < CURRENT_DATE) para evitar datas de nascimento inválidas.

12. População da Base de Dados

Após produzir, à mão, alguns dados prévios para a base de dados, realizamos o seguinte prompt:

tuplos feitos manualmente

Estes foram os tuplos que construímos para popular a nossa base de dados. Será que consegues construir mais 10 tuplos para cada tabela, tendo em atenção horários e datas para que façam sentido?

** ... **

“Quase tudo o que fizeste está correto, exceto alguns tuplos que escreveste na ordem errada”

Identificação dos tuplos errados e como fazê los de forma correta

“Segue os exemplos que te dei acima e refaz os tuplos das tabelas que te pedi para corrigir, seguindo a estrutura que te indiquei”

** ... **

Numa primeira tentativa, a ferramenta de IA, produziu alguns dados que correspondiam ao esperado, mas também produziu outros que possuíam, não só tipo de dados errados como também o número errado de colunas. Num prompt seguinte alguns dos erros foram arranjados, mas ainda não havia correspondência entre o tipo de dados no contexto, havendo, por exemplo, dados que colocavam uma empresa de som responsável pela comida do evento. Após mais algumas tentativas e uma ligeira alteração manual de dados, retirando valores a mais,

obtivemos um resultado satisfatório que optamos por adicionar aos nossos dados iniciais, guardando o resultado no ficheiro “populate2.sql”.

13. Avaliação da ferramenta de IA utilizada e membros do grupo

Após a realização deste trabalho, pudemos concluir que é nítida e impressionante a capacidade de ferramentas como, no nosso caso, o “chat gpt” para analisar, produzir e corrigir dados. Todavia, é necessário ter bastante cautela e espírito crítico no que toca a analisar as sugestões dadas visto que, como fomos reforçando ao longo do trabalho, nem sempre as respostas dadas pela ferramenta seguiam o layout pedido ou respeitavam os limites que havíamos definido para a geração de dados.

No que diz respeito à participação individual de cada um dos membros do grupo, não verificamos qualquer situação anómala e tudo correu bem durante a realização do trabalho!

14. Conclusão

Em suma, para a realização deste trabalho foi requisitada a realização de uma base de dados baseada num tema à escolha do grupo, seguindo as indicações dadas no PDF de descrição do projeto. Respeitamos todas as indicações fornecidas e seguimos à risca todos os feedbacks e sugestões dados pelo professor da componente prática da Unidade Curricular.

Juntamente com este relatório serão entregues, tal como solicitado, mais quatro documentos denominados “create1.sql”, “create2.sql”, “populate1.sql” e “populate2.sql” correspondentes, respetivamente, à criação inicial da base de dados em sql, base de dados final com base em sugestões de ferramentas de IA, ficheiro de população de base de dados com tuplos criados manualmente e, por fim, ficheiro de população da base de dados com dados criados por inteligência artificial.

Este projeto foi benéfico para todos os membros do grupo, visto que nos auxiliou a perceber melhor os tópicos lecionados na unidade curricular e a colocar em prática esses conhecimentos numa situação que se aproximava mais de algo real.