



Universidade do Minho

Mestrado Integrado em Engenharia Informática
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Letivo de 2017/2018

Sistema de gestão de eventos sociais

**David Kramer A77849, Joaquim Simões A77653,
José Menezes A79187, Rafael Costa A61799**

Novembro, 2017

BD

| | |
|-----------------|--|
| Data de Receção | |
| Responsável | |
| Avaliação | |
| Observações | |

Sistema de gestão de eventos sociais

**David Kramer A77849, Joaquim Simões A77653,
José Menezes A79187, Rafael Costa A61799**

Novembro, 2017

Resumo

Durante este relatório fomos demonstrando os vários passos percorridos para a realização da base de dados.

Inicialmente apresentamos a contextualização e o porquê de este projeto ser viável perante a atual conjuntura, tendo sido apresentados certos requisitos e condições a serem cumpridas. De seguida, construímos o modelo concetual analisando vários aspetos importantes tais como o Domínio dos Atributos, Verificação das Redundâncias no modelo, etc. Construímos ainda os modelos lógico e físico e procedemos à sua verificação de forma a termos um projeto mais completo e eficiente. Após estes passos apresentamos as três *queries* que consideramos mais importantes para a base de dados em questão, assim como uma vista e transações mais relevantes.

Concluimos depois o trabalho com uma breve apreciação crítica do projeto nos diferentes temas abordados.

Área de Aplicação: Conceção de Sistemas de Bases de Dados

Palavras-Chave: MySQL, Modelo Conceptual, Modelo Lógico, Modelo Físico, Bases de Dados Relacionais

Índice

| | |
|---|----|
| 1.2. Apresentação do Caso de Estudo | 1 |
| 1.3. Fundamentação da implementação da base de dados | 2 |
| 1.4. Análise da Viabilidade do Projeto | 3 |
| 2. Levantamento de Requisitos | 4 |
| 2.1. Método de levantamento e de análise de requisitos adotado | 4 |
| 2.2. Requisitos Levantados | 4 |
| 2.2.2 Requisitos de Exploração | 5 |
| 2.2.3 Requisitos de Controlo | 5 |
| 2.3. Perfis de Utilização | 5 |
| 3. Metodologia para a construção do Modelo Conceptual | 6 |
| 3.1. Identificação das Entidades | 6 |
| 3.2. Identificação dos Relacionamentos | 7 |
| 3.3. Identificação e associação dos Atributos | 8 |
| 3.3.1 Pessoa | 8 |
| 3.3.2 Evento | 8 |
| 3.3.3 Edição | 8 |
| 3.3.4 Organização | 9 |
| 3.4. Domínio dos Atributos | 9 |
| 3.5. Definição das Chaves Candidatas, Primárias e Alternativas | 13 |
| 3.6. Verificação das Redundâncias no Modelo | 13 |
| 3.6.1 Reexaminar Relacionamentos (1:1) | 13 |
| 3.6.2 Remover Relacionamentos Redundantes | 13 |
| 3.6.3 Considerar Dimensão Temporal | 14 |
| 3.7. Apresentação e Explicação do Diagrama ER | 14 |
| 3.8. Validação do Modelo Conceptual contra Transações do Utilizador | 15 |
| 3.8.1 Adicionar uma nova Pessoa | 15 |
| 3.8.2 Adicionar uma nova Organização | 15 |
| 3.8.3 Indicar que uma Pessoa vai a uma Edição | 15 |
| 3.8.4 Adicionar uma Edição | 15 |
| 3.9. Validação do Modelo Conceptual | 16 |
| 4. Metodologia para a construção do Modelo Lógico | 17 |

| | |
|--|----|
| 4.1. Derivar Relações a partir do Modelo Conceptual | 17 |
| 4.2. Desenho do modelo lógico | 19 |
| 4.3. Validar Relações usando a Normalização | 19 |
| 4.3.1 Unnormalized Form (UNF) | 21 |
| 4.3.2 First Normal Form (1NF) | 22 |
| 4.3.3 Second Normal Form (2NF) | 22 |
| 4.3.4 Third Normal Form (3NF) | 22 |
| 4.4. Validação do modelo com interrogações do utilizador | 22 |
| 4.4.1 Permitir a consulta das informações de uma organização | 22 |
| 4.4.2 Permitir a consulta das informações de uma edição de um evento | 23 |
| 4.4.3 Listar os eventos, e suas edições, que irão decorrer numa dada data ou intervalo de datas | 23 |
| 4.4.4 Listar os eventos, e suas edições, a decorrer numa dada região (país/cidade) | 23 |
| 4.4.5 Informar quais as organizações envolvidas num dado evento | 23 |
| 4.4.6 Informar qual a edição ou edições associadas a um evento | 23 |
| 4.4.7 Informar quais os eventos organizados por uma organização | 24 |
| 4.4.8 Listar as diferentes pessoas que tencionam ir a um evento | 24 |
| 4.4.9 Filtrar eventos e edições de acordo com uma certa estatística (rácio entre homens e mulheres ou quantidade de pessoas de uma faixa etária) que irão decorrer no futuro | 24 |
| 4.4.10 Listar eventos e edições de acordo com o seu preço, que irão decorrer no futuro | 24 |
| 4.4.11 Listar os eventos que não estejam lotados, que irão decorrer no futuro | 24 |
| 4.5. Validar Relações contra Transações do Utilizador | 25 |
| 4.5.1 Adicionar uma nova Pessoa | 25 |
| 4.5.2 Adicionar uma nova Organização | 25 |
| 4.5.3 Indicar que uma Pessoa vai a uma Edição | 25 |
| 4.5.4 Adicionar uma Edição | 26 |
| 4.6. Verificar Integridade das Restrições | 26 |
| 4.6.1 Dados Necessários | 26 |
| 4.6.2 Domínio das Restrições dos Atributos | 26 |
| 4.6.3 Cardinalidade | 26 |
| 4.6.4 Integridade das Entidades | 27 |
| 4.6.5 Integridade Referencial | 27 |
| 4.6.6 Restrições Gerais | 27 |
| 4.7. Revisão do Modelo Lógico | 28 |
| 4.8. Verificar Futuro Crescimento | 28 |
| 5. Metodologia para a construção do Modelo Físico | 29 |
| 5.1. Seleção do Sistema de gestão de Base de Dados | 29 |

| | |
|---|----|
| 5.2. Relações | 29 |
| 5.3. Representação dos Atributos Derivados | 34 |
| 5.4. Tradução das interrogações do utilizador para SQL | 34 |
| 5.4.1 Pesquisar Eventos por Intervalo de Datas | 34 |
| 5.4.2 Pesquisar Eventos por Maioria de Género | 35 |
| 5.4.3 Pesquisar Eventos por Organização | 36 |
| 5.5. Tradução das transações estabelecidas para SQL | 37 |
| 5.5.1 Indicar que uma Pessoa a uma Edição | 37 |
| 5.5.2 Inserir uma Edição | 40 |
| 5.5.3 Inserir uma Pessoa | 41 |
| 5.6. Escolha, definição e caracterização de Índices em SQL | 41 |
| 5.7. Estimativa do espaço em disco da base de dados e taxa de crescimento anual | 43 |
| 5.7.1 Estimativa do espaço em disco de dados | 43 |
| 5.7.2 Taxa de crescimento anual | 45 |
| 5.8. Definição e caraterização das vistas de utilização em SQL | 46 |
| 5.8.1 Pesquisar Eventos Não Lotados | 46 |
| 5.9. Definição e caracterização dos Mecanismos de Segurança em SQL | 46 |
| 5.10. Validação do Modelo Físico | 47 |
| 6. Conclusões e Trabalho Futuro | 48 |

Índice de Figuras

| | |
|---|----|
| Figura 1 - Modelo concetual | 14 |
| Figura 2 - Modelo lógico | 19 |
| Figura 3 - Interrogação EventosPorIntervalo | 35 |
| Figura 4 - Interrogação EventosPorGenero | 36 |
| Figura 5 - Interrogação EventosOrganizacao | 37 |
| Figura 6 - Transação insertPessoaVaiEdicao (verificação e atualização das estatísticas relativas ao género) | 38 |
| Figura 7 - Transação insertPessoaVaiEdicao (atualização das estatísticas das faixas etárias) | 39 |
| Figura 8 - Transação insertPessoaVaiEdicao (inserção) | 39 |
| Figura 9 - Transação insertEdicao | 40 |
| Figura 10 - Vista EventosNaoLotados | 46 |

Índice de Tabelas

| | |
|--|----|
| Tabela 1 - Caracterização das entidades | 7 |
| Tabela 2 - Relacionamentos | 8 |
| Tabela 3 - Domínios dos Atributos | 12 |
| Tabela 4 - Descrição das Relações do modelo lógico | 18 |
| Tabela 5 - Pessoa | 20 |
| Tabela 6 – Foto | 20 |
| Tabela 7 - Contacto | 20 |
| Tabela 8 - PessoaVaiEdicao | 20 |
| Tabela 9 - Evento | 20 |
| Tabela 10 – OrganizacaoEvento | 20 |
| Tabela 11 – Edicao | 21 |
| Tabela 12 - Organizacao | 21 |
| Tabela 13 - Pessoa | 30 |
| Tabela 14 – Evento | 30 |
| Tabela 15 – Edicao | 31 |
| Tabela 16 – Organizacao | 32 |
| Tabela 17 – Foto | 32 |
| Tabela 18 - PessoaVaiEdicao | 33 |
| Tabela 19 - OrganizacaoEvento | 33 |
| Tabela 20 – Contacto | 34 |
| Tabela 21 - Espaço por entrada da tabela Pessoa | 43 |
| Tabela 22 - Espaço por entrada da tabela Foto | 43 |
| Tabela 23 - Espaço por entrada da tabela Edicao | 44 |
| Tabela 24 - Espaço por entrada da tabela PessoaVaiEdicao | 44 |
| Tabela 25 - Espaço por entrada da tabela Evento | 44 |
| Tabela 26 - Espaço por entrada da tabela Organizacao | 44 |
| Tabela 27 - Espaço por entrada da tabela Contacto | 45 |
| Tabela 28 - Espaço por entrada da tabela OrganizacaoEvento | 45 |

1. Introdução

1.1. Contextualização

Atualmente, tendo em conta o estado de desenvolvimento cultural das cidades no nosso país (e fora deste), assim como o crescente interesse social na cultura de entretenimento, qualquer cidade apresenta uma vasta rede de estabelecimentos de âmbito social, nomeadamente, cafés, bares, clubes ou discotecas e outros de âmbito cultural, como galerias, cinemas, etc. os quais organizam frequentemente diversos eventos. Existem ainda organizações públicas e privadas responsáveis por eventos de maior escala, tal como festivais de música. Consequentemente surge uma vasta rede de pessoas que, em diversas ocasiões, pretendem atender a eventos tendo em conta a sua descrição e especificações, a sua popularidade e ainda quem são as pessoas interessadas no mesmo.

Neste contexto é benéfico organizar os dados relativos aos eventos que decorrem numa dada região, tal como dados acerca das pessoas que os frequentam, de modo a fornecer a cada utilizador as informações relativas aos eventos decorrentes e como estes se relacionam com as suas possibilidades e interesses.

1.2. Apresentação do Caso de Estudo

Decidiu-se construir uma base de dados que permita gerir e consultar eventos sociais realizados numa qualquer região. Esta ideia surgiu com o objetivo de permitir obter, de uma maneira cómoda, a informação detalhada de quaisquer eventos organizados numa dada data, verificar quem os organizou e as pessoas que, por sua vez, irão atender a esses eventos.

Os eventos são caracterizados por um nome, e possuem uma ou mais edições. Por sua vez uma edição tem a si associada, um preço, um título, uma data de início e final do evento, uma lotação máxima, uma breve descrição do que se irá realizar, uma morada (composta por país, cidade e rua). Além disso, uma edição de um evento fornece estatísticas que informam a quantidade de mulheres *versus* homens que a irão frequentar, bem como a quantidade de pessoas que pertencem às diferentes faixas etárias. De notar que as faixas etárias consideradas consistem em: pessoas com menos de 18 anos, entre 18 a 25 anos, entre 25 a 30 anos e com mais de 30 anos.

Os eventos são criados por uma qualquer entidade organizadora que pode ser um qualquer tipo de estabelecimento ou uma câmara municipal. Uma organização tem a si associada uma morada (composta por país, cidade e rua), um nome, um *e-mail*, um ou mais contactos (quer sejam eles, números de telefone, telemóveis ou *websites*) e mantém as estatísticas gerais de todos os eventos organizados.

Finalmente, as pessoas (que podem ser vistas como clientes) vão a uma ou mais edições de eventos influenciando as estatísticas destas, assim como as estatísticas das organizações envolvidas na produção dos eventos. Uma pessoa é definida através de um nome, *e-mail*, data de nascimento, sexo, e tem a si associada uma ou mais fotos.

1.3. Fundamentação da implementação da base de dados

Este sistema foi pensado com o intuito de satisfazer dois diferentes tipos de utilizadores: as pessoas que vão constantemente frequentando diversos tipos de eventos sociais e as organizações de eventos.

Como sabemos, as pessoas frequentam, cada vez mais, diversos tipos de eventos de modo a escaparem às suas diversas rotinas de trabalho, procurarem fontes de inspiração, socializarem entre si ou conhecerem novas pessoas. Estas, por sua vez, nem sempre conseguem obter facilmente informações dos diversos eventos que vão constantemente ocorrendo nas suas regiões por falta de meios de divulgação dos mesmos. Muitas das vezes quando tomam conhecimento de um dado evento, já este terminou. Por outro lado, muitas delas quando vão a um evento desejam saber se este terá ou não aderência e se este irá ser frequentado pelas pessoas com que mais se identificam.

Colocando-nos no ponto de vista de uma qualquer organização de eventos, pretendemos que a informação atinja a maior quantidade possível do público alvo de uma maneira cómoda e pouco dispendiosa. Métodos como *flyers* ou cartazes que necessitam de ser distribuídos por colaboradores da organização são bastante dispendiosos e não permitem uma informação detalhada do evento organizado. Além disso, a maior parte do público alvo não é atingido pois nem sempre os cartazes são avistados e os *flyers* são constantemente descartados pelas pessoas.

Este sistema de base de dados visa auxiliar um qualquer tipo de aplicação a ser usada pelos dois tipos de utilizadores descritos acima e satisfazer as suas necessidades neste contexto.

1.4. Análise da Viabilidade do Projeto

Face ao que foi escrito nos pontos anteriores, torna-se claro que, para este projeto ser viável, é necessário que um enorme número de Organizações de Eventos e de pessoas estejam registadas na aplicação. Isto é, como um dos principais objetivos é dar a conhecer às pessoas estatísticas de um evento, quanto maior for o número de Pessoas que interagem com o sistema, mais precisas são as estatísticas de um dado evento. De notar que a veracidade de um evento é da responsabilidade de todas as Organizações de Eventos envolvidas no mesmo.

Do ponto de vista financeiro, para além do número de horas necessário na criação do projeto, há que considerar os custos relativos à compra/manutenção de um servidor com capacidade para alojar a base de dados do sistema e alojar o *website* que dará suporte à aplicação. Por outro lado, para a obtenção de lucro, as Organizações de Eventos envolvidas pagam para publicitar um evento. O dinheiro que uma Organização investe na publicidade de um evento irá determinar o número de pessoas do público alvo a atingir. Relativamente às pessoas, estas podem interagir com a aplicação sem quaisquer custos associados.

De maneira a obter fundos para a produção do projeto, pensou-se em algumas Organizações chave como parcerias. As melhores opções consistem em alguns estabelecimentos que já são para nós familiares na nossa região, ou certas Organizações públicas como a Câmara de Braga.

2. Levantamento de Requisitos

2.1. Método de levantamento e de análise de requisitos adotado

Dado os dois tipos de utilizadores (referidos no ponto 1.3), tentamos recolher todos os requisitos para que estes utilizadores pudessem ser satisfeitos. Assim, tivemos em conta que informações dos objetos presentes nesta base de dados seriam mais relevantes, assim como o tipo de ações que seriam necessárias.

2.2. Requisitos Levantados

Enumeram-se, de seguida, os requisitos que deverão ser suportados pelo sistema de base de dados a desenvolver para cada um dos diferentes tipos de requisitos (caracterização, exploração e administração).

2.2.1 Requisitos de Descrição

1. Uma Pessoa é definida através de um nome, *e-mail*, sexo e data de nascimento e possui um conjunto de fotos.
2. Um Evento é caracterizado por um título.
3. Uma Organização tem um nome, um *e-mail*, uma morada, um conjunto de contactos, estatísticas relativas à quantidade de homens *versus* mulheres. Possui também a quantidade de pessoas com menos de 18 anos, a quantidade de pessoas com idades entre os 18 e os 25 anos e a quantidade de pessoas entre os 25 e os 30 anos que frequentaram os eventos organizados.
4. Uma Edição tem um número de edição, um preço e uma lotação. Além disso possui, os mesmos valores para quantificar as diferentes faixas etárias enunciadas para uma Organização e possui valores que registam o número de mulheres e o número de homens.
5. Um contacto pode ser de 3 tipos (telefone, telemóvel ou *website*).
6. Uma morada é composta por um país, uma cidade e uma rua.

2.2.2 Requisitos de Exploração

1. Permitir a consulta das informações de uma organização.
2. Permitir a consulta das informações de uma edição de um evento.
3. Listar os eventos, e suas edições, que irão decorrer numa dada data ou intervalo de datas.
4. Listar os eventos, e suas edições, a decorrer numa dada região (país/cidade).
5. Informar quais as organizações envolvidas num dado evento.
6. Informar qual a edição ou edições associadas a um evento.
7. Informar quais os eventos organizados por uma organização de eventos.
8. Listar as diferentes pessoas que tencionam ir a uma edição de um evento.
9. Filtrar eventos e edições de acordo com uma certa estatística (rácio entre homens e mulheres ou quantidade de pessoas de uma faixa etária) que irão decorrer no futuro.
10. Listar eventos e edições de acordo com o seu preço que irão decorrer no futuro.
11. Listar os eventos e edições que não estejam lotados que irão decorrer no futuro.

2.2.3 Requisitos de Controlo

1. Permitir guardar informações relativas a uma pessoa, organização, evento e edição.
2. Permitir que uma pessoa indique que vai a um ou mais eventos (a uma edição deste).
3. Atualizar estatísticas associadas a uma edição sempre que uma nova pessoa informa que pretende ir à mesma (estatísticas de rácio entre sexo feminino e masculino e diferentes faixas etárias).
4. Atualizar as estatísticas associadas a uma organização de acordo com as estatísticas dos seus eventos.
5. Impedir que se crie uma edição com uma data de início anterior a outras edições já criadas do mesmo evento.

2.3. Perfis de Utilização

O sistema de base de dados em questão deve ser utilizado como auxílio a uma qualquer aplicação *web* ou *mobile* desenvolvida neste contexto.

As organizações devem poder conseguir registar-se e criar eventos (bem como várias edições do mesmo). Já as pessoas, ou clientes, devem conseguir registar-se e consultar as informações das entidades organizadoras de eventos, bem como todos os eventos criados e respetivas edições.

3. Metodologia para a construção do Modelo Conceptual

Nesta secção apresentamos todos os passos que efetuamos na construção do modelo conceptual para o sistema de base de dados a desenvolver. Começamos por descrever as diferentes entidades envolvidas no modelo e identificar como é que estas se relacionam entre si. De seguida identificamos os diferentes atributos que caracterizam as diferentes entidades. Para cada atributo indicamos se este pode, ou não, identificar unicamente uma entidade, pode ser nulo, se é derivado, composto ou multivalorado. No final, ilustramos o modelo conceptual desenvolvido e descrevemos os diferentes passos que efetuamos para a sua validação. O modelo conceptual foi desenvolvido com recurso à ferramenta *TerraER*.

3.1. Identificação das Entidades

De forma a construir o modelo conceptual, devemos, num primeiro momento, identificar os principais objetos nos quais os utilizadores estão interessados. Estes objetos, que são usados como entidades do modelo, podem ser identificados através de uma análise dos requisitos. Assim, devemos analisar quais os substantivos que se destacam, tendo em conta que em alguns casos estes substantivos são apenas qualidades dos objetos e não entidades. Por exemplo, “*Listar os **eventos** a decorrer numa dada **região***”, região é apenas uma qualidade do objeto evento.

Tendo em conta o que foi referido a cima, ficamos com as seguintes entidades: Pessoa, Organização, Evento e Edição, como se pode verificar na seguinte tabela:

| Nome da Entidade | Descrição |
|------------------|---|
| Pessoa | Termo geral para representar uma pessoa que vai a eventos |
| Organização | Termo geral que identifica a entidade responsável pela organização de eventos |
| Evento | Termo geral que identifica um evento |
| Edição | Termo geral que identifica a edição de um evento |

Tabela 1 - Caracterização das entidades

3.2. Identificação dos Relacionamentos

A identificação dos relacionamentos pode ser dividida em duas partes: identificar os relacionamentos em si e depois determinar a sua multiplicidade e obrigatoriedade.

De forma a identificar os relacionamentos, podemos mais uma vez analisar os requisitos e verificar quais os verbos que expressam relacionamentos entre entidades. Por exemplo, é possível identificar os relacionamentos através da análise dos requisitos:

- Permitir que uma pessoa indique que **vai** a um ou mais eventos (a uma edição deste).
- Informar quais as organizações **envolvidas** num dado evento.
- Informar qual a edição ou edições **associadas** a um evento.

Concluimos que existem três relacionamentos diferentes:

- Relacionamento entre as entidades Pessoa e Edição.
- Relacionamento entre as entidades Organização e Evento.
- Relacionamento entre as entidades Edição e Evento.

Necessitamos agora de determinar a multiplicidade e a obrigatoriedade destes relacionamentos. Relativamente ao relacionamento entre a Pessoa e Edição, através dos requisitos sabemos que uma Pessoa pode ir a um ou mais Edições e uma Edição pode ser frequentada por 1 ou mais pessoas, não havendo qualquer obrigatoriedade em ambos os lados. Ficamos assim com um relacionamento de N para N.

De seguida, temos que um Evento é criado por uma ou mais Organizações e que uma Organização pode criar 1 ou mais Eventos. Ficamos com um relacionamento de N para N, sendo que um Evento está obrigatoriamente associado a uma ou mais Organizações.

Por fim, o relacionamento entre Evento e Edição é de 1 para N pois um Evento tem uma ou mais edições e uma Edição tem apenas a si associada um Evento, havendo obrigatoriedade em ambos os sentidos do relacionamento.

Ficamos assim com os seguintes relacionamentos:

| Entidade 1 | Entidade 2 | Relacionamento | Multiplicidade | Obrigatoriedade |
|--------------------|---------------|----------------|----------------|-------------------|
| <i>Pessoa</i> | <i>Edição</i> | Vai | N-N | -- |
| <i>Organização</i> | <i>Evento</i> | Organiza | N-N | Num sentido |
| <i>Evento</i> | <i>Edição</i> | Tem | 1-N | Nos dois sentidos |

Tabela 2 - Relacionamentos

3.3. Identificação e associação dos Atributos

Apresentam-se, de seguida todos os atributos que caracterizam as entidades descritas na secção 3.1.

3.3.1 Pessoa

Uma Pessoa deve poder registar-se no sistema. Este registo requer algumas informações básicas, tais como: o nome, *e-mail*, data de nascimento e sexo. A cada registo de uma pessoa é atribuído um número identificativo (*ID*). Todas estas propriedades de uma Pessoa representam atributos simples, já que estes não podem ser decompostos em várias partes, apresentar mais do que um valor, e são independentes de quaisquer outros atributos.

Além dos atributos mínimos para efetuar um registo com sucesso, uma Pessoa pode, também, fornecer um conjunto de fotos associadas a si. Este atributo (foto) é multivalorado.

3.3.2 Evento

Um Evento é caracterizado por um título. De modo a evitar redundâncias, sempre que um Evento é criado, é atribuído a este um identificador único (*ID*). Ambos estes atributos são simples.

3.3.3 Edição

Uma Edição é caracterizada pelo seu número de edição, descrição, data de início e de fim da edição e um preço. Uma Edição possui também uma lotação que define o número máximo de pessoas que poderão atender a essa edição. Além disso, uma Edição possui um identificador único. Estas propriedades correspondem a atributos simples desta entidade.

Uma Edição é, também, caracterizada por uma morada. Esta é composta por uma rua, cidade e país. Este atributo é composto, já que é dividido em atributos simples.

Uma Edição tem a si associada um conjunto de estatísticas relativas à quantidade de homens e mulheres e número de pessoas de uma determinada faixa etária que a vão frequentar. Estas estatísticas representam um conjunto de atributos derivados, a saber, número total de mulheres, número total de homens, número total de pessoas com menos de 18 anos, número total de pessoas com idades entre os 18 e os 25 anos e número total de pessoas com idades entre os 25 e os 30 anos.

3.3.4 Organização

Tal como uma Pessoa, uma Organização pode, também, efetuar um registo, sendo depois permitido que esta crie um conjunto de eventos associados a si. Para o registo ser efetuado com sucesso, deve ser fornecido o nome da Organização e *e-mail*. Novamente, um identificador único (*ID*) é automaticamente atribuído a uma Organização que efetua um registo com sucesso. Estes atributos são simples.

Uma Organização deve também fornecer uma morada (sendo esta composta por: rua, cidade e país), assim como um conjunto de contactos. A morada representa um atributo composto e o contacto um atributo multivalorado.

Tal como uma Edição, uma Organização possui um conjunto de estatísticas que representam atributos derivados. Estes atributos são os mesmos que foram especificados para a Edição, mas representam as estatísticas gerais associadas a todas as Edições associadas a uma Organização.

3.4. Domínio dos Atributos

Na seguinte tabela são apresentados os diferentes atributos, e respetivos domínios, que caracterizam todas as entidades envolvidas no contexto do nosso problema. Para cada atributo é indicado se este pode ou não possuir um valor nulo, se é derivado ou se é multivalorado. É também ilustrado quais dos atributos representam chaves primárias e chaves alternativas. Falaremos melhor desse aspeto na secção 3.5.

| Entidade | Atributo | Descrição | Domínio | Nulo | Derivado | Multivalorado |
|----------|-----------------------------------|---|---------------------------------|------|----------|---------------|
| Pessoa | <i>ID</i> (Chave Primária) | Número Identificativo de cada Pessoa registada no sistema | Inteiro | Não | Não | Não |
| | Nome | Nome da Pessoa | <i>String</i> até 45 caracteres | Não | Não | Não |
| | <i>E-mail</i> (Chave Alternativa) | <i>Email</i> da Pessoa | <i>String</i> até 45 caracteres | Não | Não | Não |

| | | | | | | |
|--------|---|--|---|-----|-----|-----|
| | Sexo | Sexo de uma Pessoa | String com apenas um caracter ('M' para masculino ou 'F' para feminino) | Não | Não | Não |
| | Data de nascimento | Data de nascimento da Pessoa | Data | Não | Não | Não |
| | Foto | Fotografia da Pessoa | String até 1000 caracteres | Não | Não | Sim |
| Evento | ID (Chave Primária) | Número Identificativo de cada Evento criado no sistema | Inteiro | Não | Não | Não |
| | Título | Título do Evento | String até 100 caracteres | Não | Não | Não |
| Edição | Número da Edição (Chave Primária Parcial) | Número da Edição associada a um Evento | Inteiro | Não | Não | Não |
| | Preço | Preço da Edição | Real | Não | Não | Não |
| | Descrição | Descrição dos detalhes de uma Edição | String até 1000 caracteres | Não | Não | Não |
| | Início | Data e hora de início de uma Edição | Data e Tempo | Não | Não | Não |
| | Fim | Data e hora de fim de uma Edição de um Evento | Data e Tempo | Não | Não | Não |
| | Lotação | Número máximo de Pessoas que podem ir a um Edição | Inteiro | Não | Não | Não |
| | Número de mulheres | Número total de mulheres que vão a uma Edição | Inteiro | Não | Sim | Não |
| | Número de homens | Número total de homens que vão a uma Edição | Inteiro | Não | Sim | Não |
| | Pessoas com menos de 18 anos | Número total de pessoas com menos de 18 anos que vão a | Inteiro | Não | Sim | Não |

| | | | | | | |
|-------------|----------------------------------|--|------------------------------------|-----|-----|-----|
| | | uma Edição | | | | |
| | Pessoas entre os 18 e os 25 anos | Número total de pessoas com idades entre os 18 e os 25 anos que vão a uma Edição | Inteiro | Não | Sim | Não |
| | Pessoas entre os 25 e os 30 anos | Número total de pessoas com idades entre os 25 e os 30 anos que vão a uma Edição | Inteiro | Não | Sim | Não |
| | Rua | Rua correspondente à morada onde irá decorrer uma Edição | String com máximo de 50 caracteres | Não | Não | Não |
| | Cidade | Cidade correspondente à morada onde irá decorrer uma Edição | String com máximo de 30 caracteres | Não | Não | Não |
| | País | País correspondente à morada onde irá decorrer uma Edição | String com máximo de 30 caracteres | Não | Não | Não |
| Organização | ID (Chave Primária) | Número identificativo da de uma Organização | Inteiro | Não | Não | Não |
| | E-mail (Chave Alternativa) | E-mail de uma Organização | String com máximo de 45 caracteres | Não | Não | Não |
| | Nome | Nome de uma Organização | String com máximo de 45 caracteres | Não | Não | Não |
| | Rua | Rua correspondente à morada onde irá decorrer uma Edição | String com máximo de 50 caracteres | Não | Não | Não |
| | Cidade | Cidade correspondente à morada onde irá decorrer uma Edição | String com máximo de 30 caracteres | Não | Não | Não |
| | País | País correspondente | String com | Não | Não | Não |

| | | | | | | |
|--|---|---|---|-----|-----|-----|
| | | à morada onde irá decorrer uma Edição | máximo de 30 caracteres | | | |
| | Número de mulheres | Número total acumulado de mulheres que foram a um ou mais Eventos de uma Organização | Inteiro | Não | Sim | Não |
| | Número de homens | Número total acumulado de homens que foram a um ou mais Eventos de uma Organização | Inteiro | Não | Sim | Não |
| | Pessoas com menos de 18 anos | Número total acumulado de pessoas com menos de 18 anos que foram a um ou mais Eventos de uma Organização | Inteiro | Não | Sim | Não |
| | Pessoas com idades entre 18 e 25 anos | Número total acumulado de pessoas com idades entre 18 e 25 anos que foram a um ou mais Eventos de uma Organização | Inteiro | Não | Sim | Não |
| | Pessoas com idades entre os 25 e os 30 anos | Número total acumulado de pessoas com idades entre os 25 e os 30 anos que foram a um ou mais Eventos de uma Organização | Inteiro | Não | Sim | Não |
| | Contacto | Contacto de uma Organização | <i>String</i> com máximo de 45 caracteres | Não | Não | Sim |

Tabela 3 - Domínios dos Atributos

3.5. Definição das Chaves Candidatas, Primárias e Alternativas

Para o caso da entidade Pessoa podemos definir os atributos *ID* e *e-mail* como chaves candidatas, visto que são únicos para cada pessoa. Destes dois atributos o *e-mail* tem a possibilidade de ser alterado, e possui mais caracteres do que o número *ID*, pelo que este último é utilizado como chave primária da entidade Pessoa, e o *e-mail* permanece como chave alternativa.

Para a entidade Organização, temos como chaves candidatas os seguintes atributos: *e-mail* de registo e *ID*. Analogamente ao caso anterior o *ID* será considerado a chave primária, e o *e-mail* mantém-se como chave alternativa.

A entidade Evento só possui dois atributos, *ID* e nome, sendo que nada exclui a possibilidade de existirem dois eventos com o mesmo nome. Temos então *ID* como única chave candidata.

Para o caso da entidade Edição, o atributo descrição aparenta a possibilidade de ser considerado único visto ser muito improvável haver descrições inteiramente repetidas, no entanto nada exclui essa possibilidade. A morada também não toma um valor único já que as edições de eventos tomam lugar apenas durante um específico intervalo de tempo naquela morada, o que não impossibilita o usufruto da mesma morada por parte de subsequentes edições do mesmo evento ou de outros eventos. Neste caso o atributo escolhido como chave primária foi o atributo que caracteriza o número da edição. De notar que a entidade Edição é uma *WEAK ENTITY*, ou seja, o número da edição é apenas uma chave primária parcial, sendo que a chave primária desta entidade consiste numa chave composta pelos atributos número da edição e identificador único do evento a que a edição está associada.

3.6. Verificação das Redundâncias no Modelo

3.6.1 Reexaminar Relacionamentos (1:1)

Tendo em conta que o modelo não apresenta qualquer relacionamento do tipo um para um podemos ignorar este primeiro passo.

3.6.2 Remover Relacionamentos Redundantes

Este passo não é necessário visto que o nosso modelo não apresenta ciclos, apenas existe um sentido para a obtenção de informação.

3.6.3 Considerar Dimensão Temporal

Este passo também não obriga que o nosso modelo seja alterado. Como foi mencionado no segundo passo, não foi necessário retirar relacionamentos visto que o nosso modelo não contém ciclos.

3.7. Apresentação e Explicação do Diagrama ER

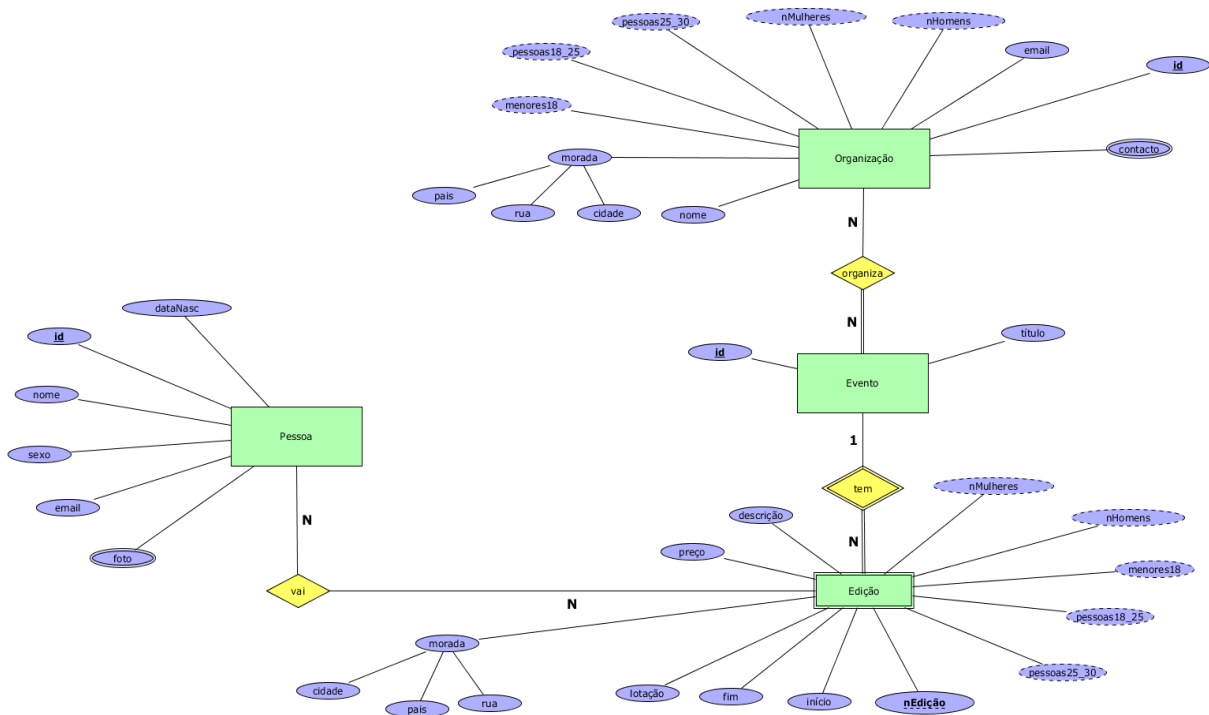


Figura 1 - Modelo conceitual

A figura acima ilustra o nosso Modelo Conceitual proposto para o sistema de base de dados a desenvolver, seguindo a notação *Chen*. Podem-se, facilmente, identificar as diferentes entidades envolvidas (Pessoa, Organização, Evento e Edição). De notar que a entidade Edição é uma *WEAK ENTITY*. Isto significa que esta entidade não pode ser unicamente identificada pelos seus atributos (o atributo chave desta entidade, sendo este o numero da edição, é apenas parcial). Esta entidade será unicamente identificada através do seu atributo chave parcial e através do atributo chave da entidade Evento. Como a existência desta entidade é dependente da entidade Evento o relacionamento Evento **tem** Edição é do tipo *WEAK RELATIONSHIP*, havendo participação total. Por sua vez um Evento é sempre organizado por uma ou mais Organizações, havendo uma participação total da entidade Evento neste relacionamento.

Finalmente, podem-se identificar dois atributos multivalorados (foto e contacto), bem como um atributo composto, sendo este a morada. Este último atributo está presente nas entidades Edição e Organização.

3.8. Validação do Modelo Conceptual contra Transações do Utilizador

Tendo em conta o sistema de base de dados a desenvolver e, tendo em conta os requisitos descritos na secção 2, consideremos que o Modelo Conceptual deve estar preparado para suportar pelo menos três operações fundamentais (adicionar uma Pessoa, adicionar uma Organização e permitir que uma Pessoa indique que vai a uma Edição).

3.8.1 Adicionar uma nova Pessoa

Através do atributo *e-mail* pertencente à entidade Pessoa é possível que, ao se adicionar uma nova Pessoa, se verifique que esta já se encontre registada.

3.8.2 Adicionar uma nova Organização

Da mesma maneira que se pode verificar se uma Pessoa já se encontra registada ou não, também é possível verificar se uma Organização se encontra, ou não, registada no sistema.

3.8.3 Indicar que uma Pessoa vai a uma Edição

Como é exemplificado no modelo conceptual, esta associação é representada por um relacionamento (N:N), indicando que uma Pessoa *vai* a muitas Edições e, por sua vez, uma Edição recebe múltiplas Pessoas. Através desta associação pode-se influenciar as estatísticas relativas a uma Edição. Por sua vez, como uma Edição está relacionada com um Evento e este a várias Organizações pode-se, também, influenciar as estatísticas de todas as Organizações envolvidas num Edição.

3.8.4 Adicionar uma Edição

Para se adicionar uma nova Edição esta tem que, obrigatoriamente, estar associada a um Evento. Para além disso a data de início da Edição deve ser menor que a data de fim desta. É importante considerar que não é permitido adicionar uma nova Edição de um Evento com uma data anterior a qualquer outra Edição anterior do Evento em questão. O modelo

desenvolvido consegue validar este processo através da associação entre um Evento **tem** muitas Edições.

3.9. Validação do Modelo Conceptual

Tendo em conta o que foi descrito na secção anterior e, tendo em conta os requisitos mínimos que foram impostos para o desenvolvimento duma primeira fase deste sistema, verifica-se que o modelo conceptual construído está preparado para o sistema a desenvolver.

4. Metodologia para a construção do Modelo Lógico

Apresentamos de seguida os detalhes de todos os passos que efetuamos para o desenvolvimento do modelo lógico. Começamos por descrever o processo que efetuamos para a derivação de relações a partir do modelo conceptual apresentado na secção anterior. De seguida apresentamos o desenho do modelo lógico em si, com recurso ao programa *MySQL Workbench*. No processo de validação do modelo em questão, verificamos se este cumpria as regras da normalização e, se este era capaz de responder às interrogações e transações por parte do utilizador. Terminamos esta secção com a verificação da integridade dos dados envolvidos, descrevendo as restrições consideradas e domínios dos mesmos.

4.1. Derivar Relações a partir do Modelo Conceptual

Começa-se por derivar as relações de forma a apresentar as entidades, relações e atributos presentes no modelo conceptual. Especifica-se o nome da relação, seguida pela lista dos seus atributos, identificando as chaves primárias, estrangeiras e atributos derivados (caso existam). Na transição do modelo conceptual para o modelo lógico começa-se por escolher as chaves primárias para cada entidade. Tendo definido as chaves primárias para cada entidade, podemos construir o modelo lógico.

Em relacionamentos do tipo 1:N a chave primária da entidade “pai” (1), é utilizada como chave estrangeira na entidade “filha” (N). Por exemplo, na entidade Evento a chave primária é o ID e na entidade Edição de Evento, esse ID é utilizado como chave estrangeira. Os relacionamentos N:N originam uma nova tabela que inclui as chaves primárias de ambas as entidades. Finalmente, os atributos multivalorados (Foto e Contacto) tratam-se como uma relação do tipo 1:N.

| |
|--|
| Pessoa (id, email, nome, sexo, dataNasc) |
| Chave primária id |
| Organização (id, email, nome, pais, cidade, rua, nMulheres, nHomens, menores18, pessoas18_25, pessoas25_30) |
| Chave primária id |

| |
|---|
| Atributos derivados: nMulheres, nHomens, menores18, pessoas18_25, pessoas25_30 |
| Edicao (nEdicao, eventoID, descricao, preco, lotacao, inicio, fim, pais, cidade, rua, nMulheres, nHomens, menores18, pessoas18_25, pessoas25_30) Chave primária composta eventoID, nEdicao Chave estrangeira eventoID referencia Evento (id) Atributos Derivados: nMulheres, nHomens, menores18, pessoas18_25, pessoas25_30 |
| Evento (id, titulo) Chave primária id |
| Foto (id, pessoalID, foto) Chave primária id Chave estrangeira pessoalID referencia Pessoa(id) |
| Contacto (id, orgID, contacto, tipo) Chave primária id Chave estrangeira orgID referencia Organizacao (id) |
| OrganizacaoEvento (orgID, eventoID) Chave primária composta (orgID, eventoID) Chave estrangeira orgID referencia Organizacao (ID) Chave estrangeira eventoID referencia Evento (ID) |
| PessoaVaiEdição (pessoalID, eventoID, nEdicao) Chave primária composta (pessoalID, eventoID, nEdicao) Chave estrangeira pessoalID referencia Pessoa (id) Chave estrangeira (eventoID, nEdicao) referencia Edicao (eventoID, nEdicao) |

Tabela 4 - Descrição das Relações do modelo lógico

4.2. Desenho do modelo lógico

Obtidas todas as relações e respectivos relacionamentos, basta criar o modelo lógico. Para este propósito recorreremos ao *MySQL Workbench* para desenhar o modelo. Obtivemos o seguinte:

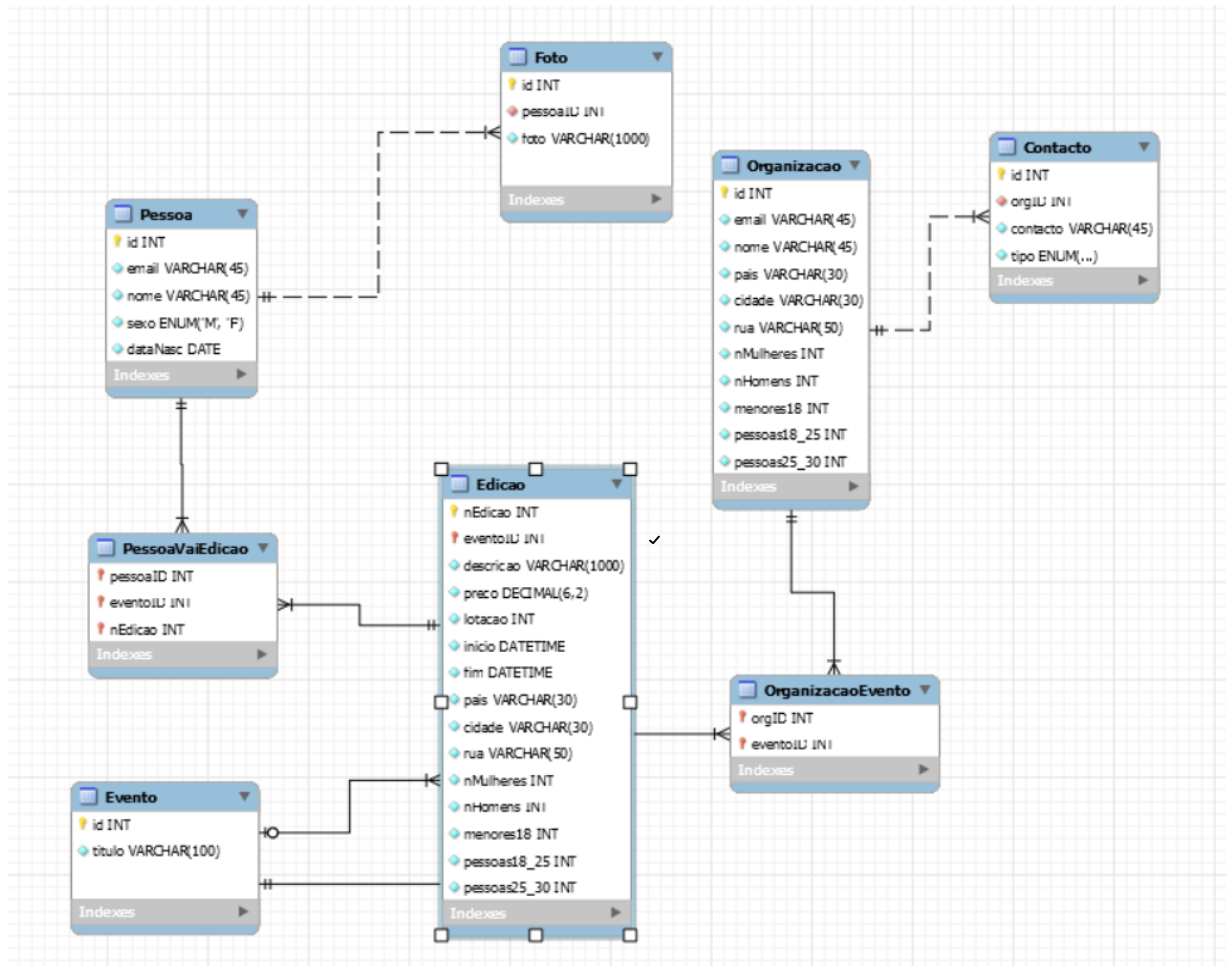


Figura 2 - Modelo lógico

4.3. Validar Relações usando a Normalização

O processo de normalização consiste em produzir relações com as propriedades corretas, dados os requisitos de uma base de dados com o intuito de aumentar a eficiência e a integridade dos dados. Assim, são usadas várias formas de normalização.

De forma a aplicar estas formas de normalização, fizemos povoamento *ad hoc* de um primeiro esboço do modelo lógico, ficando com o seguinte esquema:

| <u>ID</u> | NOME | SEXO | EMAIL | DATANASC |
|-----------|-----------|------|-----------------------|------------|
| 1 | Sifo-Dias | M | sifo_melonz@gmail.com | 2001-09-11 |

| | | | | |
|---|-------------|---|-----------------------|------------|
| 2 | Leia Organa | F | Leia_organa@gmail.com | 1956-10-21 |
| 3 | Han Solo | M | Han_fallonz@gmail.com | 1942-07-13 |

Tabela 5 - Pessoa

| <u>ID</u> | PESSOAID (FK) | FOTO |
|-----------|---------------|------------------|
| 1 | 2 | \fotos\foto1.jpg |
| 2 | 1 | \fotos\foto2.jpg |
| 3 | 3 | \fotos\foto3.jpg |

Tabela 6 – Foto

| <u>ID</u> | ORGID (FK) | CONTACTO | TIPO |
|-----------|------------|--|-----------|
| 1 | 1 | 913035697 | Telemóvel |
| 2 | 2 | 253456789 | Telefone |
| 3 | 3 | www.urban.com | Website |

Tabela 7 - Contacto

| <u>PESSOAID(FK)</u> | <u>EVENTOID(FK)</u> | <u>NEDICAO(FK)</u> |
|---------------------|---------------------|--------------------|
| 1 | 3 | 1 |
| 2 | 2 | 1 |
| 3 | 1 | 2 |

Tabela 8 - PessoaVaiEdicao

| <u>ID</u> | TITULO |
|-----------|---------------------------|
| 1 | Sábado à noite |
| 2 | Festival Paredes de Moura |
| 3 | Passagem de ano |

Tabela 9 - Evento

| <u>ORGID(FK)</u> | <u>EVENTOID(FK)</u> |
|------------------|---------------------|
| 1 | 2 |
| 2 | 3 |
| 3 | 1 |

Tabela 10 – OrganizacaoEvento

Para efeitos de visualização, não foram ilustrados todos os atributos pertencentes ao esquema das próximas tabelas.

| <u>NEDICAO</u> | PRECO | LOTACAO | FIM | INICIO | N MULHERES | MENORES 18 | <u>EVENTOID</u> (FK) |
|----------------|-------|---------|-----------------------|------------------------|---------------|---------------|-------------------------|
| 1 | 5€ | 1000 | 2017-03-05 6:00:00 | 2017-03-05 23:00:00 | 400 | 0 | 2 |
| 2 | 20€ | 20000 | 2018-08-02 5:00:00 | 2018-08-01 14:00:00 | 11000 | 4000 | 1 |
| 1 | 10€ | 5000 | 2018-01-01 8:00:00 | 2017-12-31 23:00:00 | 3000 | 600 | 3 |

Tabela 11 – Edicao

| <u>ID</u> | NOME | CIDADE | N MULHERES | PESSOAS 18 _25 | PESSOAS 25 _30 | MENORES 18 | N HOMENS |
|-----------|-----------------|--------|---------------|----------------------|----------------------|---------------|-------------|
| 1 | Plano B | Porto | 5006 | 3000 | 400 | 1600 | 2000 |
| 2 | Câmara de Braga | Braga | 10000 | 0 | 11000 | 4000 | 6000 |
| 3 | Urban | Lisboa | 5000 | 0 | 0 | 5500 | 500 |

Tabela 12 - Organizacao

Este pequeno povoamento *ad hoc* permite que se verifique, com muita mais facilidade, se o nosso modelo lógico segue as regras da normalização.

4.3.1 *Unnormalized Form (UNF)*

Facilmente se verifica no nosso modelo que nenhuma tabela possui um ou mais grupos repetidos.

4.3.2 *First Normal Form (1NF)*

Esta regra dita que, **a interseção entre cada linha e cada coluna deve conter um e um só valor.**

O nosso esquema obedece a esta regra, pelo que não há nada a alterar.

4.3.3 *Second Normal Form (2NF)*

Todos os atributos de uma tabela devem ser totalmente funcionalmente dependentes da chave primária dessa tabela.

Novamente, o nosso esquema obedece a esta regra já que todos os atributos são totalmente funcionalmente dependentes das chaves primárias das tabelas a que pertencem.

4.3.4 *Third Normal Form (3NF)*

Uma relação que cumpra a primeira e a segunda regras de normalização e em que todos os seus atributos, que não sejam chaves primárias, são transitivamente dependentes da chave primária.

Observando os esquemas de todas as relações presentes no Modelo Lógico, facilmente se verifica que todos os atributos são apenas determinados através da chave primária. Assim sendo, o nosso modelo cumpre esta regra de normalização.

4.4. Validação do modelo com interrogações do utilizador

As interrogações do utilizador correspondem aos requisitos de exploração. Como tal, para validar o modelo iremos provar que a partir do modelo lógico podemos responder com sucesso aos requisitos.

4.4.1 Permitir a consulta das informações de uma organização

As informações relativas a uma organização de eventos podem ser obtidas através da consulta da relação Organizacao e da relação Contacto. Através da Organizacao, somos capazes de obter informações como país, rua, número de homens, por exemplo. Relativamente a relação Contacto, podemos obter o(s) contacto(s) da organização em questão.

4.4.2 Permitir a consulta das informações de uma edição de um evento

As informações de uma edição de um evento são obtidas da mesma forma que as informações de uma organização de eventos, diferindo nas relações a consultar. Assim, basta consultar a relação Edicao para obter informações relativas à edição do evento, e a relação Evento para obter o nome do Evento.

4.4.3 Listar os eventos, e suas edições, que irão decorrer numa dada data ou intervalo de datas

A relação Edicao tem informações relativas a data de início e de fim de uma dada edição de um evento. Assim, sabendo o intervalo de datas basta filtrar as edições ficando com aquelas cujas datas de início ou de fim pertençam ao intervalo de datas.

4.4.4 Listar os eventos, e suas edições, a decorrer numa dada região (país/cidade)

A relação Edicao tem informações onde uma edição de evento ocorre, como a cidade, país e rua. Como tal, basta o utilizador indicar o país ou cidade para obter a edição que decorre nessa região.

4.4.5 Informar quais as organizações envolvidas num dado evento

De forma a obter as organizações envolvidas num dado evento, basta consultar a relação OrganizacaoEvento, nomeadamente a chave primária eventoID, uma vez que esta associa um evento a uma organização através das suas chaves primárias.

4.4.6 Informar qual a edição ou edições associadas a um evento

A relação Edicao é identificada através da chave primária evento e nEdicao, de tal forma que através de uma consulta desta relação podemos obter todas/as edição/edições associadas a um evento.

4.4.7 Informar quais os eventos organizados por uma organização

Tal como mencionado no ponto **4.4.5**, a relação `OrganizacaoEvento` associa um evento a uma organização através das suas chaves primárias, bastando assim consultar essa relação, nomeadamente a chave primária `orgID`, para obter o evento ou eventos organizados por organização.

4.4.8 Listar as diferentes pessoas que tencionam ir a um evento

A relação `PessoaVaiEdicao` indica todas as pessoas que vão a um evento. Assim, através da consulta desta relação conseguimos obter as pessoas que vão a uma edição de um evento.

4.4.9 Filtrar eventos e edições de acordo com uma certa estatística (rácio entre homens e mulheres ou quantidade de pessoas de uma faixa etária) que irão decorrer no futuro

Todas as estatísticas de uma edição de evento podem ser obtidas através da relação `Edicao`, com atributos como `personas25_30` que indicam o número de pessoas entre os 25 e 30 anos, ou o atributo `nMulheres` que indica o número de mulheres, por exemplo. Assim, no caso do atributo `nMulheres`, podemos obter o rácio entre homens e mulheres fazendo o quociente entre o número de mulheres e o número total de pessoas (soma entre os atributos `nMulheres` e `nHomens`).

4.4.10 Listar eventos e edições de acordo com o seu preço, que irão decorrer no futuro

O preço de uma edição de evento pode ser obtido através da relação `Edicao`, bastando assim indicar o preço ou intervalo de preços para que possamos filtrar a edição de evento segundo esse preço ou intervalo de preços.

4.4.11 Listar os eventos que não estejam lotados, que irão decorrer no futuro

A relação `Edicao` indica-nos a lotação de uma edição de evento e o número total de pessoas nessa edição, através dos atributos `lotacao` e soma dos atributos `nHomens` e

nMulheres, respetivamente. Assim se a lotação for igual ao número total de pessoas, essa edição passa a estar lotada e por isso passa a ser não listada (no caso desta interrogação).

4.5. Validar Relações contra Transações do Utilizador

4.5.1 Adicionar uma nova Pessoa

Insere-se (caso o *e-mail* já não exista) um *e-mail*, nome, sexo e data de nascimento na tabela Pessoa. Como a chave primária (*ID*) da Pessoa está presente na tabela intermédia PessoaVaiEdicao como chave estrangeira, consegue-se relacionar Pessoas com Edições de Eventos.

4.5.2 Adicionar uma nova Organização

Insere-se (caso o *e-mail* já não exista) um *e-mail*, nome e morada na tabela Organizacao. Como a chave primária (*ID*) da Organizacao está presente na tabela intermédia OrganizacaoEvento como chave estrangeira, consegue-se relacionar Organizações com os seus Eventos.

4.5.3 Indicar que uma Pessoa vai a uma Edição

As tabelas Pessoa e Edicao estão associadas entre si através de uma tabela intermédia chamada PessoaVaiEdicao. O esquema desta tabela intermédia consiste em três chaves estrangeiras (*pessoaID*, *eventoID*, *nEdicao*) que referenciam as chaves primárias das tabelas Pessoa e Edicao. Através desta associação é possível atualizar as estatísticas da tabela Edicao sempre que uma Pessoa informa que pretende ir a uma Edição. Quando esta informação é indicada ao sistema, é criada uma nova entrada na tabela intermédia com os *IDs* da Pessoa e da Edição em questão. Através dos dados sexo e data de nascimento, presentes no esquema da tabela Pessoa, podem-se incrementar os valores de nMulheres ou nHomens e o valor da faixa etária a que a Pessoa em questão corresponde.

As estatísticas das Organizações envolvidas na criação da Edição de um Evento que uma Pessoa vai são, também, facilmente atualizadas. A tabela Edicao possui uma chave estrangeira chamada evento que referencia o atributo *eventoID* da tabela Evento. Através deste *eventoID* e, com recurso à tabela intermédia OrganizacaoEvento, podemos descobrir os identificadores únicos das Organizações envolvidas no evento em questão. Com estes identificadores pode-se então proceder à atualização das estatísticas das Organizações envolvidas.

4.5.4 Adicionar uma Edição

Insere-se, caso já exista um evento associado, os valores relativos ao número de edição, evento a que a edição está associada, descrição, preço, lotação e datas de início e de fim na tabela *Edicao*. De notar que, a data de início deve ser menor que data de fim e que a data de início não pode ser menor que as datas de Edições anteriores. Através do identificador (*eventoID*), consegue-se saber se a edição em questão tem um evento associado.

4.6. Verificar Integridade das Restrições

4.6.1 Dados Necessários

Como foi descrito na secção 3.4 todos os atributos não podem ser nulos. Assim sendo, todas as colunas de todas as tabelas devem possuir um valor associado. Para este efeito foi ativada a opção *NOT NULL* em cada coluna de cada tabela.

4.6.2 Domínio das Restrições dos Atributos

O tipo de dados dos diferentes atributos foi escolhido de acordo com o que foi ilustrado na secção 3.4 e segundo o contexto de cada um dos atributos. Para valores inteiros foi selecionado o tipo de dados *INT*, para os preços das edições de um evento foi selecionado o tipo de dados *DECIMAL* com duas casas decimais e máximo de seis algarismos. Para datas foram escolhidos os tipos *DATE* e *DATETIME* (no caso de se querer guardar também a informação da hora). Para *Strings* foi utilizado o tipo *VARCHAR* com tamanho apropriado ao atributo em questão. Para as fotos optou-se por guardar o caminho destas num *VARCHAR* com tamanho considerável ao invés de as guardar num *BLOB*. Esta opção irá reduzir consideravelmente a memória ocupada pela base de dados. Finalmente, para os contactos foi criado um tipo enumerado (*ENUM*), de modo a distinguir os diferentes tipos de contactos (telefone, telemóvel e *website*).

4.6.3 Cardinalidade

Como já foi descrito, existem dois relacionamentos *many-to-many* no nosso modelo. Uma Pessoa pode ir a muitas Edições e por sua vez uma Edição recebe várias Pessoas. Por outro lado, uma Organização cria múltiplos Eventos e um Evento pode ser criado por várias Organizações. No modelo lógico, estes dois relacionamentos traduzem duas tabelas intermédias, a saber, *PessoaVaiEdicao* e *OrganizacaoEvento*.

4.6.4 Integridade das Entidades

Todas as chaves primárias não podem suportar valores nulos. Esta restrição foi imposta pela opção *NOT NULL*.

4.6.5 Integridade Referencial

Existem alguns casos que devemos considerar de modo a manter a integridade referencial, quando se alteram os dados que referenciam ou são referenciados por uma tabela.

Inserção de uma entrada nas seguintes tabelas (Foto, Contacto, PessoaVaiEdicao, Edicao e OrganizacaoEvento). Estas tabelas representam relações do tipo “filho” e não faz sentido existir uma entrada numa destas tabelas que referencia uma relação do tipo “pai” nula. Assim sendo, as inserções nestas tabelas só são válidas se já existirem entradas “pai” que lhes correspondem.

Ao se remover uma entrada da tabela Pessoa, deve-se também remover todas as entradas na tabela Foto que estão associadas à Pessoa em questão, visto que não faz sentido uma foto estar associada a uma pessoa com valor nulo. Ainda para este caso, deve-se remover qualquer entrada na tabela PessoaVaiEdicao relativa à Pessoa em questão. Ao se remover uma entrada da tabela Evento devem-se remover todas as Edições associadas. Ao se remover uma entrada da tabela Organizacao devem-se remover todos os seus Contactos, bem como todas as entradas pertencentes à tabela OrganizacaoEventos. Estes casos em que são removidas entradas de relações do tipo “pai” podem ser tratados à custa da ação *CASCADE*. Esta ação faz com que, sempre que haja uma atualização ou remoção de uma entrada de uma relação do tipo “pai”, todas as entradas de relações do tipo “filho” sejam também atualizadas ou removidas, mantendo a integridade referencial.

4.6.6 Restrições Gerais

Os *e-mails* das Pessoas e das Organizações, as fotos, os contactos e o título de um evento devem ser únicos. Para garantir esta restrição, foi ativada a opção *UNIQUE* nas colunas *e-mail*, *foto*, *contacto* e *titulo*.

Certos atributos como preço, lotação, número de homens ou de mulheres e número de pessoas pertencentes a uma determinada faixa etária não devem possuir valores negativos. Para resolver esta questão foi ativada a opção *UNSIGNED DATA TYPE* em todas as colunas que representam estes atributos.

Na tabela relativa a uma Edição a data e hora de fim deve ser posterior à data e hora de início de uma Edição, posterior às datas de início e de fim de Edições anteriores.

Nas tabelas Edicao e Organizacao, o resultado da soma dos atributos número de homens e número de mulheres deve ser maior ou igual que o resultado da soma dos atributos relativos às diferentes faixas etárias.

Na tabela relativa a uma Pessoa o sexo apenas pode possuir dois valores, a saber '*M*' que denomina o sexo Masculino e '*F*' que denomina o sexo Feminino.

Na tabela relativa a um Contacto o atributo tipo apenas pode ser um '*telefone*', '*telemovel*' ou um '*website*'.

4.7. Revisão do Modelo Lógico

Este modelo permite o registo de Pessoas e de Organizações. Além disso, permite que uma Organização crie múltiplos Eventos e respetivas Edições destes. Como foi descrito, é também possível que qualquer Pessoa indique que vai a uma Edição de um Evento. Por sua vez, esta indicação irá influenciar tanto as estatísticas que referenciam uma Edição, bem como todas as eventuais Organizações envolvidas. Concluimos, então, que o modelo apresentado cumpre os requisitos impostos.

4.8. Verificar Futuro Crescimento

É difícil prever quais poderão ser os novos requisitos adicionais no futuro e se estes são consistentes com o modelo apresentado. No entanto, o sistema encontra-se normalizado e sem redundâncias o que facilitará a manutenção e futuro desenvolvimento. Além disto, o modelo encontra-se preparado para suportar um qualquer aumento do número de Pessoas, Eventos ou Organizações. A maneira como as diferentes Relações estão associadas permite que uma alteração do esquema destas para suportar novos requisitos (controlo de estatísticas para mais faixas etárias por exemplo) não afete os requisitos mínimos impostos nesta primeira versão do sistema.

5. Metodologia para a construção do Modelo Físico

Nesta secção abordaremos os aspetos relativos à metodologia para a construção do modelo físico e alguns exemplos. Será detalhado o processo de transição do modelo lógico para o modelo físico, utilizando para esse efeito uma ferramenta de gestão de Base de Dados, nomeadamente o *MySQL*. Serão abordadas as traduções de interrogações do utilizador para SQL tal como as transações estabelecidas e as vistas de utilização.

5.1. Seleção do Sistema de gestão de Base de Dados

Para a gestão da Base de Dados decidimos optar pelo *MySQL*. Esta ferramenta apresenta várias vantagens entre as quais o reduzido custo de manutenção associado a um elevado desempenho para qualquer que seja a carga exigida (possuindo deste modo uma elevada escalabilidade). Para além destes aspetos tivemos em conta também a facilidade na utilização e administração comparativamente a outros sistemas sendo que esta é a ferramenta na qual os membros do grupo têm maior experiência.

5.2. Relações

Apresentam-se, de seguida, as descrições das diferentes relações pertencentes ao Modelo Físico. Para cada relação é listado o seu esquema (e descrição dos domínios dos diferentes atributos), bem como parte da instrução necessária, em *MySQL*, para a sua criação.

| | |
|----------|--|
| ID | Inteiro |
| Email | Variável conjunto de caracteres únicos com tamanho 45. |
| Nome | Variável conjunto de caracteres com tamanho 45. |
| Sexo | Caracter de valor 'F' para o sexo feminino e 'M' para o sexo masculino. Possui o valor 'F' como valor por defeito. |
| dataNasc | Data de nascimento da pessoa. |

```

Pessoa (
  `id` INT NOT NULL AUTO_INCREMENT,
  `email` VARCHAR(45) NOT NULL,
  `nome` VARCHAR(45) NOT NULL,
  `sexo` ENUM('M', 'F') NOT NULL DEFAULT 'F',
  `dataNasc` DATE NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `email_UNIQUE` (`email` ASC)
)

```

Tabela 13 - Pessoa

| | |
|--|--|
| ID | Inteiro |
| Título | Variável conjunto de caracteres com tamanho 100. |
| Evento` (`id` INT NOT NULL AUTO_INCREMENT, `titulo` VARCHAR(100) NOT NULL, PRIMARY KEY (`id`) UNIQUE INDEX `titulo_UNIQUE` (`titulo` ASC)) | |

Tabela 14 – Evento

| | |
|--------------|--|
| NEdicao | Inteiro positivo com valor 1 por defeito. |
| EventoID | Inteiro. |
| Descricao | Variável conjunto de caracteres com tamanho 1000. |
| Preco | Decimal positivo com 6 algarismos e duas casas decimais. Possui o valor 0 por defeito. |
| Lotacao | Inteiro positivo (valor 0 por defeito). |
| Fim | Data e hora de final da edição de evento. |
| Inicio | Data e hora de início da edição de evento. |
| NMulheres | Atributo derivado inteiro positivo. Valor 0 por defeito. |
| Pessoas25_30 | Atributo derivado inteiro positivo. Valor 0 por defeito. |
| Pessoas18_25 | Atributo derivado inteiro positivo. Valor 0 por defeito. |
| Menores18 | Atributo derivado inteiro positivo. Valor 0 por defeito. |
| NHomens | Atributo derivado inteiro positivo. Valor 0 por defeito. |
| Cidade | Variável conjunto de caracteres com tamanho 30. |
| Pais | Variável conjunto de caracteres com tamanho 30. |

| | |
|--|---|
| Rua | Variável conjunto de caracteres com tamanho 50. |
| Edicao (`nEdicao` INT NOT NULL, `eventoID` INT NOT NULL, `descricao` VARCHAR(1000) NOT NULL, `preco` DECIMAL(4,2) UNSIGNED NOT NULL, `lotacao` INT UNSIGNED NOT NULL, `inicio` DATETIME NOT NULL, `fim` DATETIME NOT NULL, `pais` VARCHAR(30) NOT NULL, `cidade` VARCHAR(30) NOT NULL, `rua` VARCHAR(50) NOT NULL, `nMulheres` INT NOT NULL DEFAULT 0, `nHomens` INT NOT NULL DEFAULT 0, `menores18` INT NOT NULL DEFAULT 0, `pessoas18_25` INT NOT NULL DEFAULT 0, `pessoas25_30` INT NOT NULL DEFAULT 0, CHECK (fim > inicio AND numero_de_homens + numero_de_mulheres >= pessoas_menores_18 + pessoas_18_25 + pessoas_25_30), INDEX `fk_Edicao_de_Evento_Evento1_idx` (`eventoID` ASC), PRIMARY KEY (`nEdicao`, `eventoID`), CONSTRAINT `fk_Edicao_de_Evento_Evento1` FOREIGN KEY (`eventoID`) REFERENCES `eventosdb`.`Evento` (`id`) ON DELETE CASCADE ON UPDATE CASCADE)) | |

Tabela 15 – Edicao

| | |
|--------------|--|
| ID | Inteiro. |
| Nome | Variável conjunto de caracteres com tamanho 45. |
| Email | Variável conjunto de caracteres únicos com tamanho 45. |
| NMulheres | Atributo derivado inteiro positivo. Valor 0 por defeito. |
| Pessoas25_30 | Atributo derivado inteiro positivo. Valor 0 por defeito. |
| Pessoas18_25 | Atributo derivado inteiro positivo. Valor 0 por defeito. |
| Menores18 | Atributo derivado inteiro positivo. Valor 0 por defeito. |
| NHomens | Atributo derivado inteiro positivo. Valor 0 por defeito. |
| Cidade | Variável conjunto de caracteres com tamanho 30. |
| Pais | Variável conjunto de caracteres com tamanho 30. |
| Rua | Variável conjunto de caracteres com tamanho 50. |

```

Organizacao (
`id` INT NOT NULL AUTO_INCREMENT,
`email` VARCHAR(45) NOT NULL,
`nome` VARCHAR(45) NOT NULL,
`pais` VARCHAR(30) NOT NULL,
`cidade` VARCHAR(30) NOT NULL,
`rua` VARCHAR(50) NOT NULL,
`nMulheres` INT NOT NULL DEFAULT 0,
`nHomens` INT NOT NULL DEFAULT 0,
`menores18` INT NOT NULL DEFAULT 0,
`pessoas18_25` INT NOT NULL DEFAULT 0,
`pessoas25_30` INT NOT NULL DEFAULT 0,
CHECK (nHomens + nMulheres >=
      menores18 + pessoas18_25 + pessoas25_30),
PRIMARY KEY (`id`),
UNIQUE INDEX `email_UNIQUE` (`email` ASC)
)

```

Tabela 16 – Organizacao

| | |
|---|---|
| ID | Inteiro. |
| Pessoa | Inteiro. |
| Foto | Variável conjunto de caracteres com tamanho 1000. |
| Foto (<pre> `id` INT NOT NULL AUTO_INCREMENT, `pessoaID` INT NOT NULL, `foto` VARCHAR(1000) NOT NULL, PRIMARY KEY (`id`), INDEX `fk_foto_Pessoa1_idx` (`pessoaID` ASC), UNIQUE INDEX `foto_UNIQUE` (`foto` ASC), CONSTRAINT `fk_foto_Pessoa1` FOREIGN KEY (`pessoaID`) REFERENCES `eventosdb`.`Pessoa` (`id`) ON DELETE CASCADE ON UPDATE CASCADE) </pre> | |

Tabela 17 – Foto

| | |
|--|-------------------|
| PessoaID | Inteiro. |
| EventoID | Inteiro. |
| NEdicao | Inteiro positivo. |
| PessoaVaiEdicao (<pre> `pessoaID` INT NOT NULL, `eventoID` INT NOT NULL, `nEdicao` INT NOT NULL, PRIMARY KEY (`pessoaID`, `eventoID`, `nEdicao`), INDEX `fk_Pessoa_has_Edicao_de_Evento_Pessoa1_idx` (`pessoaID` ASC), INDEX `fk_Pessoa_has_Edicao_de_Evento_EdicaoEvento1_idx` (`eventoID` ASC, `nEdicao` ASC), CONSTRAINT `fk_Pessoa_has_Edicao_de_Evento_Pessoa1` FOREIGN KEY (`pessoaID`) REFERENCES `eventosdb`.`Pessoa` (`id`) </pre> | |


```

ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `fk_Pessoa_has_Edicao_de_Evento_EdicaoEvento1`
FOREIGN KEY (`eventoID`, `nEdicao`)
REFERENCES `eventosdb`.`Edicao` (`eventoID`, `nEdicao`)
ON DELETE CASCADE
ON UPDATE CASCADE)
)

```

Tabela 18 - PessoaVaiEdicao

| | |
|---|----------|
| OrgID | Inteiro. |
| EventoID | Inteiro. |
| OrganizacaoEvento (`orgID` INT NOT NULL, `eventoID` INT NOT NULL, PRIMARY KEY (`orgID`, `eventoID`), INDEX `fk_Organizacao_de_Eventos_has_Evento_Evento1_idx` (`eventoID` ASC), INDEX `fk_Organizacao_de_Eventos_has_Evento_Organizacao_de_Eventos_idx` (`orgID` ASC), CONSTRAINT `fk_Organizacao_de_Eventos_has_Evento_Organizacao_de_Eventos1` FOREIGN KEY (`orgID`) REFERENCES `eventosdb`.`Organizacao` (`id`) ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT `fk_Organizacao_de_Eventos_has_Evento_Evento1` FOREIGN KEY (`eventoID`) REFERENCES `eventosdb`.`Evento` (`id`) ON DELETE CASCADE ON UPDATE CASCADE)) | |

Tabela 19 - OrganizacaoEvento

| | |
|---|---|
| ID | Inteiro. |
| OrgID | Inteiro. |
| Tipo | Tipo enumerado. Pode possuir os valores 'telefone', 'telemovel' e 'website'. Valor 'telemovel' por defeito. |
| Descricao | Variável conjunto de caracteres únicos com tamanho 45. |
| Contacto (`id` INT NOT NULL AUTO_INCREMENT, `orgID` INT NOT NULL, `contacto` VARCHAR(45) NOT NULL, `tipo` ENUM('telefone', 'telemovel', 'website') NOT NULL DEFAULT 'telemovel', PRIMARY KEY (`id`), INDEX `fk_contacto_Organizacao_de_Eventos1_idx` (`orgID` ASC), UNIQUE INDEX `descricao_UNIQUE` (`contacto` ASC), CONSTRAINT `fk_contacto_Organizacao_de_Eventos1` FOREIGN KEY (`orgID`) REFERENCES `eventosdb`.`Organizacao` (`id`)) | |

| |
|--|
| ON DELETE CASCADE ON UPDATE CASCADE)) |
|--|

Tabela 20 – Contacto

5.3. Representação dos Atributos Derivados

No nosso sistema de base de dados, os atributos derivados são os seguintes: nMulheres, nHomens, menores18, pessoas18_25 e pessoas25_30. Ambos estes atributos estão presentes nas tabelas Edicao e Organizacao. Todos estes atributos são inteiros positivos e devem ser incrementados em uma unidade sempre que uma Pessoa indica que vai a uma Edição. Obviamente, estes devem ser incrementados de acordo com os dados da Pessoa em questão. Por exemplo, se a “Maria” que nasceu no dia 27 de fevereiro de 1999 indica que vai à primeira edição do evento “Reabertura do *Urban Beach*”, então os atributos da Edição em questão, número de mulheres e o número de pessoas com idades entre os 18 e os 25 anos devem ser incrementados. Por sua vez, os mesmos atributos de todas as Organizações envolvidas na criação do evento “X” devem, também, ser incrementados em uma unidade.

Estes atributos derivados devem ser atualizados sempre que se insere uma entrada na tabela PessoaVaiEdicao. Este aspeto é importante já que, sempre que há uma consulta na base de dados que envolva qualquer um destes atributos, é necessário que estes estejam disponíveis e devidamente atualizados.

5.4. Tradução das interrogações do utilizador para SQL

Nesta secção analisamos algumas das interrogações do utilizador (*queries*) mais relevantes deste projeto.

Alguns aspetos importantes comuns em todas as *queries* são, por exemplo, o facto de serem omitidos nos resultados os *ID*'s das entidades, já que essa informação não tem qualquer interesse para o utilizador. Outro aspeto comum a todas as *queries* são as informações que são devolvidas nas pesquisas por eventos. Estas incluem sempre o título do evento, o número da edição, a descrição da edição do evento, a lotação, o início, o fim, o país, cidade e rua, a percentagem de homens e de mulheres e a percentagem de pessoas nas faixas etárias seguintes: menores de 18, entre os 18 e os 25, entre os 25 e os 30 e maiores de 30.

5.4.1 Pesquisar Eventos por Intervalo de Datas

Esta *query* é utilizada quando um utilizador pretende pesquisar eventos que se realizem num certo intervalo de datas.

```
$$
create procedure EventosPorIntervalo (in inicio date, fim date)
begin
  if (inicio >= fim) then
    select ('Datas invalidas') as msg;
  else
    select e.titulo,
           o.nEdicao, o.descricao, o.lotacao, o.inicio, o.fim, o.pais, o.cidade, o.rua,
           concat(round(o.nMulheres / (o.nMulheres + o.nHomens) * 100, 2), '%') as mulheres,
           concat(round(o.nHomens / (o.nMulheres + o.nHomens) * 100, 2), '%') as homens,
           concat(round(o.menores18 / (o.nMulheres + o.nHomens) * 100, 2), '%') as menores18,
           concat(round(o.pessoas18_25 / (o.nMulheres + o.nHomens) * 100, 2), '%') as entre18_25,
           concat(round(o.pessoas25_30 / (o.nMulheres + o.nHomens) * 100, 2), '%') as entre25_30
    from evento e
    join edicao o
    on e.id = o.eventoID
    where (date(o.fim) >= inicio and date(o.fim) <= fim) or
           (date(o.inicio) >= inicio and date(o.inicio) <= fim);
  end if;
end; $$
```

Figura 3 - Interrogação EventosPorIntervalo

Um utilizador fornece os dados de início e final do intervalo de tempo em que pretende pesquisar eventos e são apresentados todos as edições de eventos que decorram nesse intervalo. Para eventos que decorram durante vários dias, serão devolvidos aqueles cujos, independentemente de começarem antes ou acabarem depois do dado intervalo, decorrem total ou parcialmente durante esse intervalo de tempo.

Se um utilizador inserir na pesquisa uma data de início posterior à data de fim, a pesquisa não ocorre e devolve uma mensagem de erro.

5.4.2 Pesquisar Eventos por Maioria de Género

Esta *query* possibilita a pesquisa de edições de eventos futuros ordenados por ordem decrescente de percentagem de participantes do género (sexo) pretendido.

```

$$
create procedure EventosPorGenero (in sexo enum('M', 'F'))
begin
  if sexo = 'M' then
    select e.titulo,
           o.nEdicao, o.descricao, o.lotacao, o.inicio, o.fim, o.pais, o.cidade, o.rua,
           concat(round(o.nMulheres / (o.nMulheres + o.nHomens) * 100, 2), '%') as mulheres,
           concat(round(o.nHomens / (o.nMulheres + o.nHomens) * 100, 2), '%') as homens,
           concat(round(o.menores18 / (o.nMulheres + o.nHomens) * 100, 2), '%') as menores18,
           concat(round(o.pessoas18_25 / (o.nMulheres + o.nHomens) * 100, 2), '%') as entre18_25,
           concat(round(o.pessoas25_30 / (o.nMulheres + o.nHomens) * 100, 2), '%') as entre25_30
    from edicao o
    join evento e
    on o.eventoID = e.id
    where date(o.inicio) >= date(now())
    order by o.inicio asc, o.nHomens desc;
  else
    select e.titulo,
           o.nEdicao, o.descricao, o.lotacao, o.inicio, o.fim, o.pais, o.cidade, o.rua,
           concat(round(o.nMulheres / (o.nMulheres + o.nHomens) * 100, 2), '%') as mulheres,
           concat(round(o.nHomens / (o.nMulheres + o.nHomens) * 100, 2), '%') as homens,
           concat(round(o.menores18 / (o.nMulheres + o.nHomens) * 100, 2), '%') as menores18,
           concat(round(o.pessoas18_25 / (o.nMulheres + o.nHomens) * 100, 2), '%') as entre18_25,
           concat(round(o.pessoas25_30 / (o.nMulheres + o.nHomens) * 100, 2), '%') as entre25_30
    from edicao o
    join evento e
    on o.eventoID = e.id
    where date(o.inicio) >= date(now())
    order by o.inicio asc, o.nMulheres desc;
  end if;
end; $$

```

Figura 4 - Interrogação EventosPorGenero

Um utilizador escolhe o sexo que pretende, por exemplo, se pretender ir a um evento que seja maioritariamente atendido por mulheres, pesquisa 'F,' e é-lhe devolvida a lista de eventos futuros ordenados decrescentemente em função da percentagem de mulheres que tencionam participar.

Através da utilização da função *date(now())*, e comparando esta com a data de início dos eventos, garantimos que a pesquisa apenas devolve edições futuras de eventos, o que faz mais sentido para o utilizador, pois este apenas quer saber quais os eventos futuros correspondem às suas preferências (já que seria um pouco complicado para um utilizador participar num evento que já se realizou).

5.4.3 Pesquisar Eventos por Organização

Esta *query* devolve todas as edições de eventos organizados por uma dada organização, por ordem decrescente de data de realização do evento.

```

$$
create procedure EventosOrganizacao (in nome varchar(45))
begin
    select org.nome, e.titulo,
           o.nEdicao, o.descricao, o.lotacao, o.inicio, o.fim, o.pais, o.cidade, o.rua,
           concat(round(o.nMulheres / (o.nMulheres + o.nHomens) * 100, 2), '%') as mulheres,
           concat(round(o.nHomens / (o.nMulheres + o.nHomens) * 100, 2), '%') as homens,
           concat(round(o.menores18 / (o.nMulheres + o.nHomens) * 100, 2), '%') as menores18,
           concat(round(o.pessoas18_25 / (o.nMulheres + o.nHomens) * 100, 2), '%') as entre18_25,
           concat(round(o.pessoas25_30 / (o.nMulheres + o.nHomens) * 100, 2), '%') as entre25_30
    from edicao o
    join evento e
    on o.eventoID = e.id
    join organizacao org
    on org.nome like concat ('%', nome, '%')
    where e.id in (
        select orgE.eventoID from organizacaoevento orgE
        where orgE.orgID in (
            select org.id from organizacao org
            where org.nome like concat ('%', nome, '%'))
        )
    order by o.inicio desc;
end; $$

```

Figura 5 - Interrogação EventosOrganizacao

Para a criação desta *query* foi necessário fazer dois *join*'s, visto serem necessários dados de 3 tabelas distintas: da organização, do evento e da respetiva edição.

É de notar que a pesquisa pode ser efetuada através de uma *string* que corresponda apenas a parte do título da organização pretendida. Isto deve-se à utilização da função *like concat*. Por exemplo, para pesquisar a organização *Urban Beach*, poderíamos apenas inserir a *string* 'urb'.

5.5. Tradução das transações estabelecidas para SQL

5.5.1 Indicar que uma Pessoa a uma Edição

Esta transação é usada quando uma pessoa vai a uma edição de um evento, sendo preciso atualizar todas as estatísticas da edição e organizações do evento em questão assim como inserir uma entrada na relação *PessoavaiEdicao*. A transação é a seguinte:

```

create procedure insertPessoaVaiEdicao(in p int, e int, ne int)
begin
    declare erro bool default 0;
    declare existePessoa bool default 1;
    declare existeE bool default 1;
    declare s enum ('M', 'F');
    declare idade int;
    declare d date;
    declare organizacao int;
    declare continue handler for sqlexception set erro = 1;

    start transaction;

    set existePessoa = (select id from pessoa where id = p);
    set existeE = (select nEdicao, eventoID from edicao where nEdicao = ne and eventoID = e);

    -- Verifica validade da transação
    if existePessoa and existeE then
        set d = (select dataNasc from pessoa where id = p);
        set idade = (select extract(year from (from_days(datediff(now(), d)))) + 0);
        set s = (select sexo from pessoa where id = p);

        if (s = 'M') then
            update edicao
            set nHomens = nHomens + 1
            where eventoID = e and nEdicao = ne;

            update organizacao as og
            set og.nHomens = og.nHomens + 1
            where og.id in (
                select orgID from organizacaoevento
                where eventoID = e
            );
        else
            update edicao
            set nMulheres = nMulheres + 1
            where eventoID = e and nEdicao = ne;

            update organizacao as og
            set og.nMulheres = og.nMulheres + 1
            where og.id in (
                select orgID from organizacaoevento
                where eventoID = e
            );
        end if;
    end if;
end if;

```

Figura 6 - Transação *insertPessoaVaiEdicao* (verificação e atualização das estatísticas relativas ao género)

Num primeiro momento, através das variáveis *existePessoa* e *existeE*, verificamos se a pessoa que introduzimos na transação existe, como também verificamos se a edição de evento e evento correspondem a uma entrada na relação *Edicao*. No caso destas interrogações terem resposta afirmativa, começamos por atualizar as estatísticas da edição e da organização (tabelas *Edicao* e *Organizacao*, respetivamente) relativas ao número de pessoas de cada sexo.


```

end if;

if (idade >= 18 and idade <= 25) then
    update edicao
        set pessoas18_25 = pessoas18_25 + 1
        where eventoID = e and nEdicao = ne;

    update organizacao as og
        set og.pessoas18_25 = og.pessoas18_25 + 1
        where og.id in (
            select orgID from organizacaoevento
            where eventoID = e
        );
elseif (idade >= 25 and idade <= 30) then
    update edicao
        set pessoas25_30 = pessoas25_30 + 1
        where eventoID = e and nEdicao = ne;

    update organizacao as og
        set og.pessoas25_30 = og.pessoas25_30 + 1
        where og.id in (
            select orgID from organizacaoevento
            where eventoID = e
        );
elseif (idade < 18) then
    update edicao
        set menores18 = menores18 + 1
        where eventoID = e and nEdicao = ne;

    update organizacao as og
        set og.menores18 = og.menores18 + 1
        where og.id in (
            select orgID from organizacaoevento
            where eventoID = e
        );
end if;

```

Figura 7 - Transação *insertPessoaVaiEdicao* (atualização das estatísticas das faixas etárias)

De seguida, atualizamos as estatísticas relativas às faixas etárias.

```

end if;

insert
    into pessoavaiedicao (pessoaID, eventoID, nEdicao)
    values (p, e, ne);
else
    select ('Pessoa ou edição inválidas') as msg;
    set erro = 1;
end if;

if erro then
    select ('Erro ao indicar que uma pessoa vai a uma edição de evento') as msg;
    rollback;
else
    select ('Operação efetuada com sucesso') as msg;
    commit;
end if;
end;

```

Figura 8 - Transação *insertPessoaVaiEdicao* (inserção)

Por fim, adicionamos a nova entrada à tabela PessoaVaiEdicao.

5.5.2 Inserir uma Edição

Transação usada para inserir uma edição:

```
create procedure insertEdicao(in ne int, e int, d varchar(1000), p decimal(6,2),
                           l int, i datetime, f datetime, pa varchar(30), c varchar(30), r varchar(50))
begin
    declare erro bool default 0;
    declare fimUltimaEdicao datetime;
    declare existe bool default 1;
    declare continue handler for sqlexception set erro = 1;

    start transaction;

    set existe = (select id from evento where id = e);

    if existe then
        if i >= f then
            select ('Data de inicio é incompatível com a data de fim do evento') as msg;
            set erro = 1;
        else
            set fimUltimaEdicao = (select fim from edicao
                                where e = eventoID
                                order by nEdicao desc
                                limit 1);

            if fimUltimaEdicao >= i then
                select ('Data de inicio da edicao atual é incompatível com as edições anteriores') as msg;
                set erro = 1;
            else
                insert
                into edicao (nEdicao, eventoID, descricao, preco, lotacao, inicio, fim, pais, cidade, rua)
                values (ne, e, d, p, l, i, f, pa, c, r);
            end if;
        end if;
    else
        select ('O evento associado à edição não existe') as msg;
        set erro = 1;
    end if;

    if erro then
        select ('Erro ao adicionar uma entrada na tabela EdicaoEvento') as msg;
        rollback;
    else
        select ('Operação efetuada com sucesso') as msg;
        commit;
    end if;
end;
```

Figura 9 - Transação insertEdicao

Ao inserir uma edição é necessário ter atenção a três pontos. Em primeiro lugar, a inserção de uma edição só é válida se o evento já existir. Através da variável **existe**, conseguimos verificar isso mesmo. Em segundo lugar, a data de início não pode ser depois da data de fim, naturalmente. Isto garante-se facilmente através da comparação direta entre as variáveis (**if i >= f**). Por fim, é preciso garantir que a edição que estamos a inserir vai ocorrer em datas posteriores a edições anteriores do mesmo evento (se existirem). Assim, selecionamos a data de fim da edição anterior à que queremos inserir, através da variável **fimUltimaEdicao**. De seguida, basta comparar essa data à data de início da edição que vamos inserir.

Depois de verificar se a inserção é válida, basta inserir a edição através dos valores fornecidos.

5.5.3 Inserir uma Pessoa

Esta transação tem como objetivo inserir uma pessoa:

```
create procedure insertPessoa(in e varchar(45), n varchar(45), s enum('M', 'F'), d date)
begin
    declare erro bool default 0;
    declare continue handler for sqlexception set erro = 1;

    start transaction;

    insert
        into pessoa (email, nome, sexo, dataNasc)
        values (e, n, s, d);

    if erro then
        select ('Erro ao adicionar uma entrada na tabela Pessoa') as msg;
        rollback;
    else
        select ('Operação efetuada com sucesso') as msg;
        commit;
    end if;
end;
```

Ao contrário das transações anteriores, esta não necessita de qualquer verificação antes da inserção, de modo que apenas podemos fazer *insert* dos valores fornecidos.

5.6. Escolha, definição e caracterização de Índices em SQL

Os índices permitem uma otimização relevante no que toca a consulta de tabelas que contenham uma enorme quantidade de registos. No entanto, não se deve abusar do recurso dos índices, já que estes apresentam uma enorme penalidade no que toca à quantidade de memória gasta.

Relativamente aos índices, o *MySQL* atribui índices às chaves primárias, pelo que todas as chaves primárias envolvidas nas tabelas são índices. Tal como foi falado, certos atributos como os *e-mails*, tanto de uma Pessoa como de uma Organização, são *UNIQUE*. Esta restrição faz com que estes atributos sejam, também, indexados. Face às consultas que foram consideradas como relevantes para o sistema em consideração, verifica-se que muitas das consultas são efetuadas através do nome de uma Organização e do título de um Evento. Como foi descrito, o título de um Evento é *UNIQUE*, pelo que este atributo é indexado. Considerou-se indexar, também o atributo nome pertencente à tabela Organização. No entanto, após uma pesquisa, verificou-se que pesquisas efetuadas com a sintaxe *LIKE '%campo a procurar%'* não usam índices. Através desta pesquisa abandonou-se a ideia da atribuição de um índice ao atributo nome, já que é interessante que nas pesquisas sejam

filtradas todas as Organizações que contenham, no seu nome, uma certa sequência de caracteres.

5.7. Estimativa do espaço em disco da base de dados e taxa de crescimento anual

5.7.1 Estimativa do espaço em disco de dados

Cada em entrada em cada relação vai ocupar um certo número de *bytes*. Começemos por calcular esse espaço:

Relação Pessoa

| Atributo | Tipo | Espaço ocupado(bytes) |
|-------------------|----------------|-----------------------|
| id | INT | 4 |
| email | VARCHAR (45) | 45 |
| nome | VARCHAR (45) | 45 |
| sexo | ENUM ('M','F') | 1 |
| dataNasc | DATE | 3 |
| Total por entrada | -- | 98 |

Tabela 21 - Espaço por entrada da tabela Pessoa

Relação Foto

| Atributo | Tipo | Espaço ocupado (bytes) |
|-------------------|----------------|------------------------|
| id | INT | 4 |
| pessoalID | INT | 4 |
| foto | VARCHAR (1000) | 1000 |
| Total por entrada | -- | 1008 |

Tabela 22 - Espaço por entrada da tabela Foto

Relação Edicao

| Atributo | Tipo | Espaço ocupado (bytes) |
|-----------|----------------|------------------------|
| nEdicao | INT | 4 |
| eventoID | INT | 4 |
| descricao | VARCHAR (1000) | 1000 |
| preco | DECIMAL (6,2) | 5 |
| lotacao | INT | 4 |
| inicio | DATETIME | 8 |
| fim | DATETIME | 8 |
| pais | VARCHAR (30) | 30 |
| cidade | VARCHAR (30) | 30 |

| | | |
|-------------------|--------------|------|
| rua | VARCHAR (50) | 50 |
| nMulheres | INT | 4 |
| nHomens | INT | 4 |
| menores18 | INT | 4 |
| peessoas18_25 | INT | 4 |
| peessoas25_30 | INT | 4 |
| Total por entrada | -- | 1163 |

Tabela 23 - Espaço por entrada da tabela Edicao

Relação PessoaVaiEdicao

| Atributo | Tipo | Espaço ocupado (bytes) |
|-------------------|------|------------------------|
| peessoalID | INT | 4 |
| eventoID | INT | 4 |
| nEdicao | INT | 4 |
| Total por entrada | -- | 12 |

Tabela 24 - Espaço por entrada da tabela PessoaVaiEdicao

Relação Evento

| Atributo | Tipo | Espaço ocupado (bytes) |
|-------------------|---------------|------------------------|
| id | INT | 4 |
| titulo | VARCHAR (100) | 100 |
| Total por entrada | -- | 104 |

Tabela 25 - Espaço por entrada da tabela Evento

Relação Organizacao

| Atributo | Tipo | Espaço ocupado (bytes) |
|-------------------|--------------|------------------------|
| id | INT | 4 |
| email | VARCHAR (45) | 45 |
| nome | VARCHAR (45) | 45 |
| pais | VARCHAR (30) | 30 |
| cidade | VARCHAR (30) | 30 |
| rua | VARCHAR (50) | 50 |
| nMulheres | INT | 4 |
| nHomens | INT | 4 |
| menores18 | INT | 4 |
| peessoas18_25 | INT | 4 |
| peessoas25_30 | INT | 4 |
| Total por entrada | -- | 448 |

Tabela 26 - Espaço por entrada da tabela Organizacao

Relação Contacto

| Atributo | Tipo | Espaço ocupado (bytes) |
|-------------------|---|------------------------|
| id | INT | 4 |
| orgID | INT | 4 |
| contacto | VARCHAR (45) | 45 |
| tipo | ENUM ('telefone', 'telemovel', 'website') | 1 |
| Total por entrada | -- | 54 |

Tabela 27 - Espaço por entrada da tabela Contacto

Relação OrganizacaoEvento

| Atributo | Tipo | Espaço ocupado (bytes) |
|-------------------|------|------------------------|
| orgID | INT | 4 |
| eventoID | INT | 4 |
| Total por entrada | -- | 8 |

Tabela 28 - Espaço por entrada da tabela OrganizacaoEvento

Necessitamos agora de um povoamento tipo para estimar o espaço total ocupado. O povoamento será do tipo: 500 pessoas das quais 400 foram a eventos e 200 tem foto, 100 eventos, 200 edições, 105 organizações todas com contacto e com 100 eventos criados.

Assim, para este tipo de povoamento, o espaço ocupado seria cerca de:

$500 * 98 + 1008 * 200 + 200 * 1163 + 12 * 400 + 104 * 200 + 448 * 105 + 54 * 105 + 100 * 8 = 562310 \text{ bytes} = 0.536 \text{ megabytes}.$

5.7.2 Taxa de crescimento anual

Para determinar a taxa de crescimento anual, vamos supor que diariamente se registam 15 pessoas, 5 delas com fotos, 5 organizações todas elas com contacto, 1 evento e 1 edição. Vamos supor também que 70 pessoas indicam que vão a uma edição por dia, e que 1 organização cria 1 evento, por dia. Assim, tendo em conta que um ano tem 365 dias, o espaço ocupado aumenta anualmente da seguinte forma:

$365 * (15 * 98 + 5 * 1008 + 5 * (448 + 54) + 104 + 1163 + 70 * 12 + 8) = 4064275 \text{ bytes} = 3.87599 \text{ MB}$

Para este exemplo, o crescimento anual seria de 3.87599 MB.

5.8. Definição e caracterização das vistas de utilização em SQL

Nesta secção analisamos uma vista de utilização (*view*), que foi a única que tivemos necessidade aplicar neste projeto.

5.8.1 Pesquisar Eventos Não Lotados

Esta *view* serve simplesmente para listar as edições de eventos futuros que não se encontram ainda lotados, ou seja, cuja capacidade máxima indicada ainda não foi atingida.

```
create view EventosNaoLotados as
select e.titulo,
       o.nEdicao, o.descricao, o.lotacao, o.inicio, o.fim, o.pais, o.cidade, o.rua,
       concat(round(o.nMulheres / (o.nMulheres + o.nHomens) * 100, 2), '%') as mulheres,
       concat(round(o.nHomens / (o.nMulheres + o.nHomens) * 100, 2), '%') as homens,
       concat(round(o.menores18 / (o.nMulheres + o.nHomens) * 100, 2), '%') as menores18,
       concat(round(o.pessoas18_25 / (o.nMulheres + o.nHomens) * 100, 2), '%') as entre18_25,
       concat(round(o.pessoas25_30 / (o.nMulheres + o.nHomens) * 100, 2), '%') as entre25_30
from evento e
join edicao o
on e.id = o.eventoID
where o.nMulheres + o.nHomens < o.lotacao and date(o.inicio) >= date(now())
order by inicio;
```

Figura 10 - Vista EventosNaoLotados

O funcionamento desta *view* não apresenta diferenças funcionais relevantes relativamente ao que foi analisado nas secções anteriores.

5.9. Definição e caracterização dos Mecanismos de Segurança em SQL

Como foi especificado na secção 2.3, este sistema deve ser utilizado como suporte a uma aplicação, quer seja esta *web* ou *mobile*. Embora a aplicação possua dois tipos de utilizadores (Pessoa e Organização de eventos), estes devem ser distinguidos pela lógica de negócio e não pelo sistema de base dados. Assim sendo, o sistema de base de dados apenas possui um tipo de utilizador, sendo este a própria aplicação. Este aspeto é importante, pois deverá ser a própria aplicação que deverá impedir que um utilizador registado como uma Pessoa crie eventos. O objetivo desta arquitetura foi pensado com o intuito de permitir que aplicação seja fácil de utilizar, já que o objetivo é permitir a criação e pesquisa de eventos de uma maneira que seja transparente ao utilizador, sem que este manipule diretamente a base de dados.

5.10. Validação do Modelo Físico

Através de uma análise detalhada de todos os esquemas de todas as tabelas desenvolvidas, bem como de todas as transações, vistas e diferentes consultas que foram propostas para o desenvolvimento em questão, verificamos que o modelo físico desenvolvido cumpre todos os requisitos que foram estipulados.

6. Conclusões e Trabalho Futuro

Iniciamos este projeto com o âmbito de aplicar os conhecimentos adquiridos nas aulas de Bases de Dados a um caso real e atual em que uma base de dados poderia ser benéfica. Neste contexto decidimos criar uma base de dados que conectasse eventos sociais com pessoas, de modo a resolver um dos principais paradoxos dos eventos sociais: a sua popularidade depende da quantidade e do tipo de pessoas que participam – ao mesmo tempo que as pessoas decidem a que evento ir consoante a sua popularidade. Muitas vezes a falta de informação resulta em que certos eventos com potencial não tenham a devida aderência e também que certas pessoas acabem por participar em eventos que não sejam da sua preferência por ignorarem as alternativas.

Nesta base de dados, relacionamos informações acerca de pessoas, organizações responsáveis por eventos, os próprios eventos e as suas respetivas edições. Ligando estas quatro entidades, e através de uma diversidade de *queries*, obtemos uma variedade de pesquisas filtradas que fornecem ao utilizador as informações específicas que pretende obter acerca de uma edição de um certo evento, ou pesquisar por eventos conforme as estatísticas deste, por exemplo, pesquisar um evento que seja atendido maioritariamente por jovens entre os 18 e os 25 anos. Para mais, a cada evento estão associados dados acerca da sua organização, as suas edições anteriores e futuras e estatísticas acerca das pessoas que atenderam ou tencionam atender a essas edições.

É de notar que decidimos dividir o que poderia ser considerado como uma entidade única Evento em duas componentes, Evento e Edição. Esta decisão melhora o desempenho da base de dados, visto que um mesmo evento pode ter várias edições e os dados relativos ao evento variam de edição para edição.

Na fase de análise de transações foram apresentados dois casos de inserções, sendo estes a inserção de uma pessoa na base de dados e a inserção de uma edição de um evento. Estas transações exemplificam o funcionamento das restantes inserções. Uma transação mais complexa foi a de indicar que uma pessoa vai a um evento, e por esse motivo foi também analisada neste relatório.

Na secção relativa à análise de *queries* foram apresentados três casos diferentes de *queries*. Embora para este projeto tenham sido criadas várias *queries* com vista a possibilitar uma grande diversidade de pesquisas para o benefício do utilizador, escolhemos apenas exemplificar três casos que, pela sua divergência, são já bastante representativos das *queries* criadas para a base de dados.

Relativamente a vistas de utilização, apenas tivemos necessidade de criar uma, já que as pesquisas são maioritariamente feitas através de *queries*.

Esperamos que esta base de dados possa servir como base de algo que poderá realmente ser utilizado por verdadeiros estabelecimentos que organizem eventos e que, num futuro breve, toda a informação que quisermos saber acerca do que se vai passar “logo à noite”, esteja à distância de um clique.

Referências

Connolly, T.M. and Begg, C.E., 2015. *Database Systems, A Pratical Approach to Design, Implementation, and Management*. 6th ed. Pearson.

Referências WWW

<http://www.vertabelo.com/blog/technical-articles/chen-erd-notation>

<https://dev.mysql.com/doc/refman/5.7/en/tutorial.html>

Lista de Siglas e Acrónimos

| | |
|-----------|--------------------------------------|
| BD | Base de Dados |
| DW | Data Warehouse |
| OLTP | <i>On-Line Analytical Processing</i> |
| SQL | Structured Query Language |
| MB | MegaByte |
| ER | Entity Relationship |