Contextualização

Como sabemos o Streaming é uma forma de distribuição digital frequentemente usada para distribuir conteúdo multimédia através da Internet, retirando a necessidade de o utilizador armazenar todo este conteúdo no seu computador.

A Arpeggio Music, um novo serviço de streaming, foi criada com a finalidade de permitir que o mundo consiga ouvir música de forma legal por um preço mais acessível que o mercado oferecia.

Requisitos

De forma a ser possível perceber as funcionalidades que são necessárias e que tipo de informação a base de dados deveria conter, começamos por realizar uma reunião com o nosso cliente de forma a proceder ao levantamento dos requisitos que ele queria que tivéssemos em conta. Além desse levantamento, analisamos outro grande serviço de música e realizamos entrevistas com os seus diversos utilizadores.

<Exemplos>

Requisito de Descrição

Utilizador para se identificar para além do seu username e email necessita de indicar o seu nome, data nascimento, pais e número de telemóvel

Requisito de Exploração

Deve ser possível procurar um artista pelo seu nome ou id

Requisito de Controlo

Uma faixa tem como atributo um número de identificação único

Após análise dos requisitos procedeu-se à modelação do modelo conceptual da BD

Modelo Conceptual

Seguimos abordagem centralizada pois não se justifica devido a complexidade da BD escolher uma abordagem de integração de vistas.

Identificação das entidades

-Utilizador

-Artista

-Playlist

-Album

-Faixa

Identificação dos relacionamentos

Faixa->Playlist Relacionamento N:M

(Uma Playlist tem N faixas e N faixas estão em M Playlists)

Album -> Faixa Relacionamento 1:N

(Um Album possui N Faixas)

Artista -> Faixa Relacionamento 1:N

(Um Artista possui N Faixas)

Artista -> Album Relacionamento 1:N

(Artista possui N Albuns)

Utilizador->Artista Relacionamento N:M

(Um Utilizador pode (ou não) seguir vários artistas. Da mesma forma um Artista pode ser seguido (ou não) por vários utilizadores)

Utilizador->Playlist 2 Relacionamentos

1-(Um utilizador tem N playlists) Relacionamento 1: N

2- (Um utilizador segue (ou não) N playlists. Da mesma forma

Uma playlist é seguida (ou não) por N ) N:M

Chaves primárias Id\_\_\_\_

Modelo Lógico

Depois da validação do modelo com o cliente iniciamos o processo de transição do modelo Conceptual para o Modelo Lógico.

A transição foi bastante simples, no entanto houve um momento em que a tradução do Modelo Conceptual para o Lógico não foi direta

De forma a manter a integridade de dados o atributo país foi transformado em tabela de maneira a evitar diferentes formas de identificar um país. As entidades País e Utilizador têm agora um relacionamento com cardinalidade 1:N entre elas.

**Relacionamentos 1:N** (gera uma cópia da chave primária da Entidade com menor cardinalidade

(entidade Pai) e coloca-a na Entidade de maior cardinalidade (entidade Filho). Esta cópia

designa-se por chave estrangeira e garante a integridade dos dados que referenciam a entidade Pai)

Utilizador -> Pais

Faixa -> Album

Faixa -> Artista

Album -> Artista

Playlist -> Utilizador

**Relacionamentos N:M** (é criada uma tabela com as chaves estrangeiras de ambas as entidades.

Cada entidade liga-se a essa tabela com uma cardinalidade 1:N(gerando assim as chaves estrangeiras))

Utilizador -> Utilizador\_follows\_Artista -> Artista

Playlist -> Playlist\_has\_Faixa -> Faixa

Validação da normalização

1ª forma normal

Os valores de todos os atributos são atómicos, isto é,

se não for possível decompô-los

Relacionamento N:1 Utilizador-Pais

Vários Utilizadores podem fazer parte do mesmo país, a cardinalidade do relacionamento

Utilizador-Pais N:1

2ª forma normal

Atributo que não seja chave primária vai depender totalmente desta, ou seja,

não possui uma dependência parcial.

Nosso modelo não possui nenhuma dependência parcial

IdAlbum -> Nome

-> Data

-> DuraçãoTotal

3ª forma normal

Verifica se se o modelo tem dependências transitivas. Todos os atributos da tabela

devem ser independentes uns dos outros sendo exclusivamente dependentes da chave primária

O nosso modelo também verifica a terceira forma normal

Validação do modelo interrogações

É necessário realizar a validação com as interrogações do utilizador, de forma a verificar

se o modelo corresponde às suas expectativas

1-Número de utilizadores, de um determinado país que seguem um artista especifico

Para determinar o número de utilizadores de um determinado país que seguem específico é

necessário recolher dados das tabelas Pais, Utilizador e Artista.

Através da relação Utilizador->Pais é possivel retirar os utilizadores de um determinado pais

Relação entre tabelas Utilizador-> Artista é possível determinar quais os utilizadores

que seguem um artista específico.

2- Procura as faixas de uma playlist

Tirando a informação das tabelas Faixa->Playlist é possível descobrir as faixas

que constituem uma determinada playlist

3-Procura quais são os álbuns de um determinado artista

É possível determinar os nomes os álbuns através da relação entre a tabela

Album->Artista, isto pois, é necessário verificar quem é o artista que o

utilizador procura e os álbuns que por ele foram lançados

4-Procura as faixas, de um dado artista, que foram lançadas entre duas datas

Para ser possível determinar que faixas foram lançadas entre duas datas por um

dado artista é necessário usar a relação que as tabelas Artista e Faixa possuem

retirando a informação necessária de cada uma

Validação do modelo com as transações estabelecidas

A validação das transações estabelecidas foi realizada através de mapas de transações.

Assim é possível ver as interações entre as diferentes tabelas de forma a obter o resultado pretendido

--**Inserção de uma faixa numa playlist**

Para adicionar uma faixa à playlist é necessário realizar duas ações. A primeira ação que se vai realizar é inserir na tabela Playlist\_has\_Faixa o Id da faixa que se pretende inserir na playlist e o Id da playlist onde se quer inserir, por fim atualizamos o atributo nrFaixas na tabela playlist

--**Remoção de uma Faixa de uma Playlist**

Raciocionio é o mesmo da inserção

**Revisao do modelo logico**

Nova reunião com o cliente

Após rever as especificações modelo foi aprovado

Modelo Fisico

Foi feita a escolha do SGBD (Sistema de Gestão de Bases de Dados) MySql

Foi feita a modelação do esquema físico envolve a tradução dos relacionamentos previamente

definidos no Modelo Lógico para um código que possa ser implementado no SGBD SQL

Queries

Todas as queries se encontram dentro de stored procedures de maneira estarem otimizadas e melhorar a velocidade já que são cached no servidor

Se forem chamadas várias vezes com diferentes valores também faz sentido

Segurança o código que implementa a base de dados não toca em instruções SQL apenas chama o procedure

1-Número de utilizadores de um determinado país, que seguem um artista específico

Explicar query

2-Procura o número de faixas de uma playlist

Explicar query

Transações

Explicar transação

Trigger

Explicar trigger

Indices

Capazes de aumentar a velocidade de procura de resultados

Capazes de ajudar a ordenar os registos mais facilmente e rapidamente

View

View Playlist Mostra todas as playlist, e as suas faixas possuidas pelos utilizadores

Passamos agora para o segundo trabalho que trata da implementação da mesma base de Dados em NoSql

Segunda Parte

NoSql

--As bases de dados relacionais acentam nas propriedades ACID

-Atomicidade

-Consistencia

-Isolamento

-Durabilidade

De forma a facilitar o crescimento da Base de Dados, em algumas circunstâncias

estas propriedades são ignoradas

-- Vantagens NoSql

-custos mais reduzidos

-dados sempre disponiveis

-base de dados orientada a objetos flexiveis

-facilidade de introducao de dados

-facilidade de gestao de grandes quantidade de informacao

Novo Sistema de Base de Dados

Implementação da BD em Neo4j

- As entidades passam a ser nodos

- Relacionamentos passam a ser arestas entre os nodos

Migração de Dados

**Exportação**

-Exportação de dados presentes na base de dados em MySQL para ficheiros

com extensão .csv

- Cada tabela foi exportada para um ficheiro diferente

- Processo de migração foi necessário importar os ficheiros para uma diretoria especifica

\r\n Carriage Return + Line Feed passa de linha e mete cursor no inicio

**Criação de nodos**

Load do ficheiro .csv faz se o create de cada nodo com os fields respetivos

**Criação de Indexes**

Mesma razão do modelo relacional

**Criação de relacionamentos**

Load CSV from ficheiro

Match para descobrir as ligações

Merge para as criar

[:Pertence] é o nome do Relacionamento

**Queries**

* 1- selecionar todos os utilizadores da base de dados, retornando os seus usernames e os respetivos emails
* 2-selecionar o utilizador onde o UtilizadorID é 1 e vê quais os artistas que ele segue retornando o username do utilizador e o nome do artista
* 3- contar quantas faixas tem o álbum cujo nome é Reputation, retornando o nome do álbum e essa contagem

**Versão final que obtemos**

**Conclusão**