

Enunciado do trabalho

O trabalho prático da disciplina deve versar sobre o projeto e uso de uma base de dados para um sistema de informação (SI), a ser modelado e implantado em computador. O trabalho envolve a modelagem e o projeto da base de dados com o uso de ferramentas de modelagem, bem como criação, instanciação e manipulação em um modelo relacional.

Esse relatório se refere à primeira parte do trabalho, dizendo respeito à implementação de um projeto conceitual da Base de Dados e sua subsequente implementação em SGBD relacional.

Universo de Discurso

O universo de discurso é baseado no site/aplicativo Google Play[1], que é um serviço de distribuição digital de aplicativos, jogos, filmes, programas de televisão, músicas e livros, desenvolvido e operado pela Google[2]. Ela é a loja oficial de aplicativos para o sistema operacional Android, além de fornecer conteúdo digital.

O universo é composto por:

- **Usuário:** Deve possuir nome, senha, cpf, data de nascimento e email (único). O usuário é responsável pela maior parte das ações no aplicativo, sendo elas:
 - Comprar Itens (livros, álbuns de música, aplicativos e filmes): Pode-se fazer o download de quantos e quaisquer itens que quiser, porém nunca o mesmo item duas vezes. Existem itens que são pagos, e portanto, suas compras só podem ser feitas por cartão e aceitas se o usuário tiver um cartão cadastrado e pagar o preço referente ao item. Depois de comprado o item, o download do mesmo é feito, e a informação de que o download foi feito é salva. Fazendo com que o item seja eternamente do usuário, ou seja, se ele quiser excluir do celular, terá a possibilidade de baixar novamente quando quiser, sem precisar pagar novamente.
 - Adicionar Itens na Lista de Desejos: Quando o usuário encontra um item que gosta e tem intenção de comprar no futuro, adiciona na sua lista de desejos, que o possibilita voltar a qualquer momento e achar o item de forma rápida. Podem ser adicionados quantos e quaisquer itens que desejar, porém apenas uma vez cada um deles.
 - Revisão de Itens: Usuários podem dar sua opinião sobre itens, tanto os que já foram usados na plataforma, quanto vistos em outros lugares. Cabe ao bom senso do usuário comentar apenas em itens que já usou. Cada usuário só pode fazer uma revisão por item, mas pode revisar quantos itens quiser. Uma revisão é composta por uma nota (de 0 a 5), uma data e opcionalmente um breve comentário.
 - Adicionar cartões: Cartões são usados para realizar compras. O usuário pode adicionar quantos desejar. O cartão não necessariamente precisa estar em seu nome,

portanto é necessário que guardemos o nome impresso no cartão. Porém não pode adicionar novamente o mesmo cartão.

- Baixar Itens: O download é ligado com a compra (mesmo que seja de um item gratuito), assim cada vez que é feita uma compra, um download é realizado. O mesmo guarda a data e a informação de que o download foi feito, uma forma de demonstrar que o usuário possui o item baixado, para que futuramente por mais que o exclua, continue tendo permissão para baixar novamente, sem que seja necessário pagar pelo mesmo novamente.

- **Cartão:** Definido por número (único), data de vencimento e usuário que é responsável pelo mesmo (único), além do nome impresso em cima do cartão, já que, como descrito acima, o dono do cartão não necessariamente precisa ser o usuário dono da conta. Sendo o usuário com um ou mais cartões, responsável pela compra de itens (que possuem um valor maior que zero). Não existem cartões que não possuam um dono, e cada cartão pode pertencer a somente um usuário.

- **Item:** Pode ser livro, aplicativo, álbum de música ou filme, e é o objeto de consumo do usuário e base da plataforma.

Definido por preço (sendo o item considerado gratuito quando esse campo tiver o valor 0, ou não estiver preenchido, já que é opcional), nome, data de lançamento, resumo e código (único). Além disso o item pode entrar na promoção, sendo necessário guardar o valor dele com desconto e sua data limite de promoção (que pode ser nula caso o item não esteja em promoção). Caso a data atual seja menor do que a data limite da promoção, entende-se que o item está em promoção, portanto o valor do item, é o valor com desconto. Não existe interesse do sistema em armazenar um histórico de promoções

O item é ligado em quase todas as ações disponíveis do usuário (menos adicionar cartões), sendo dividido em quatro tipos:

- Aplicativo: Além de todos os atributos do item possui versão, tamanho (que pode ser, no máximo 1TB) e um desenvolvedor.

- * Desenvolvedor: Possui endereço, e-mail, senha e uma quantidade n de aplicativos criados. É um tipo de conta, feita para criadores de aplicativos, onde é considerado um desenvolvedor qualquer pessoa que criou uma conta deste tipo, ou seja, não é necessário ter um aplicativo para ser um desenvolvedor.

- Livro: Além de todos os atributos do item possui ISBN[5] (único), número de páginas, um único autor e uma única linguagem. Diferentes linguagens do mesmo livro, correspondem a diferentes itens.

- * Autor: Autores são pessoas que produziram algum conteúdo que está disponibilizado no sistema. Autores de livros também podem ter produzido algum álbum, ou ter participado de algum filme. Sempre que um livro é cadastrado, é necessário autor já existente, ou, caso não exista criar um novo.

- * Linguagem: Linguagens são pré-definidas, para que possam, além de controlar as linguagens de livros existentes, ajudar em buscas específicas.

Como linguagens são definidas antes de livros, existem linguagens sem livros cadastrados com elas, mas as mesmas podem ter até n livros. Livros sempre tem

apenas uma linguagem. Inicialmente, o sistema possuirá as as linguagens descritas na Tabela 1, porém, futuramente poderão ser adicionadas novas linguagens ao sistema, de acordo com a expansão do mesmo.

- Filme: Além de todos os atributos do item possui duração, linguagens e integrantes.
 - * Linguagem: Possui o mesmo objetivo da linguagem do livro, com a única diferença de que quando se relaciona com o filme pode ser de dois tipos, áudio ou legenda. Pode existir várias linguagens de áudio e legenda para um mesmo filme, porém não podem se repetir no mesmo filme (ex: duas vezes áudio em português).
 - * Integrantes: Compostas por nome, são pessoas que participaram da produção do filme, com certas funções. Uma mesma pessoa pode ter diferentes funções em diferentes filmes, porém não pode desempenhar a mesma função duas vezes no mesmo filme. Esses integrantes também podem ter escrito algum livro, ou lançado um álbum.
- Álbum: Além de todos os atributos do item possui faixas de música e artista.
 - * Artista: É o cantor ou grupo que criou o álbum, que pode também ter escrito algum livro, ou participado de algum filme. Tem um nome único e pode possuir quantos álbuns quiser, porém um álbum só tem um artista que o compôs.
 - * Faixa de Música: São as músicas que compõem o álbum, elas tem um nome único e uma duração. Cada música só está presente em um álbum, porém um álbum tem várias músicas. Entende-se que o artista responsável por essa música, é o mesmo responsável pelo álbum ao qual essa música pertence.
- Além desta divisão de tipos, os itens também tem categorias e anexos, que são:
 - Categoria: Nada mais que os gêneros de um item (ex: ação para filmes, pop para músicas...). Gêneros são pré-definidos e um item só pode ser cadastrado com gêneros que já existem. Além disso, itens só podem ter gêneros que fazem parte do seu tipo, por exemplo, um álbum não pode ter gênero ação. Dado isso, um item pode ter quantos gêneros quiser, desde que não repita o mesmo, seja do seu tipo e esteja contido na tabela de gêneros, inicialmente definida pelos valores da tabela 2, porém podendo ser adicionadas novas categorias no futuro.
 - Anexos: Itens podem ter fotos e vídeos anexados para demonstrar seu conteúdo. Sendo necessário que para cada anexo adicionado, tenha caminho em disco de onde o arquivo foi armazenado (único), tipo de arquivo, nome e extensão.

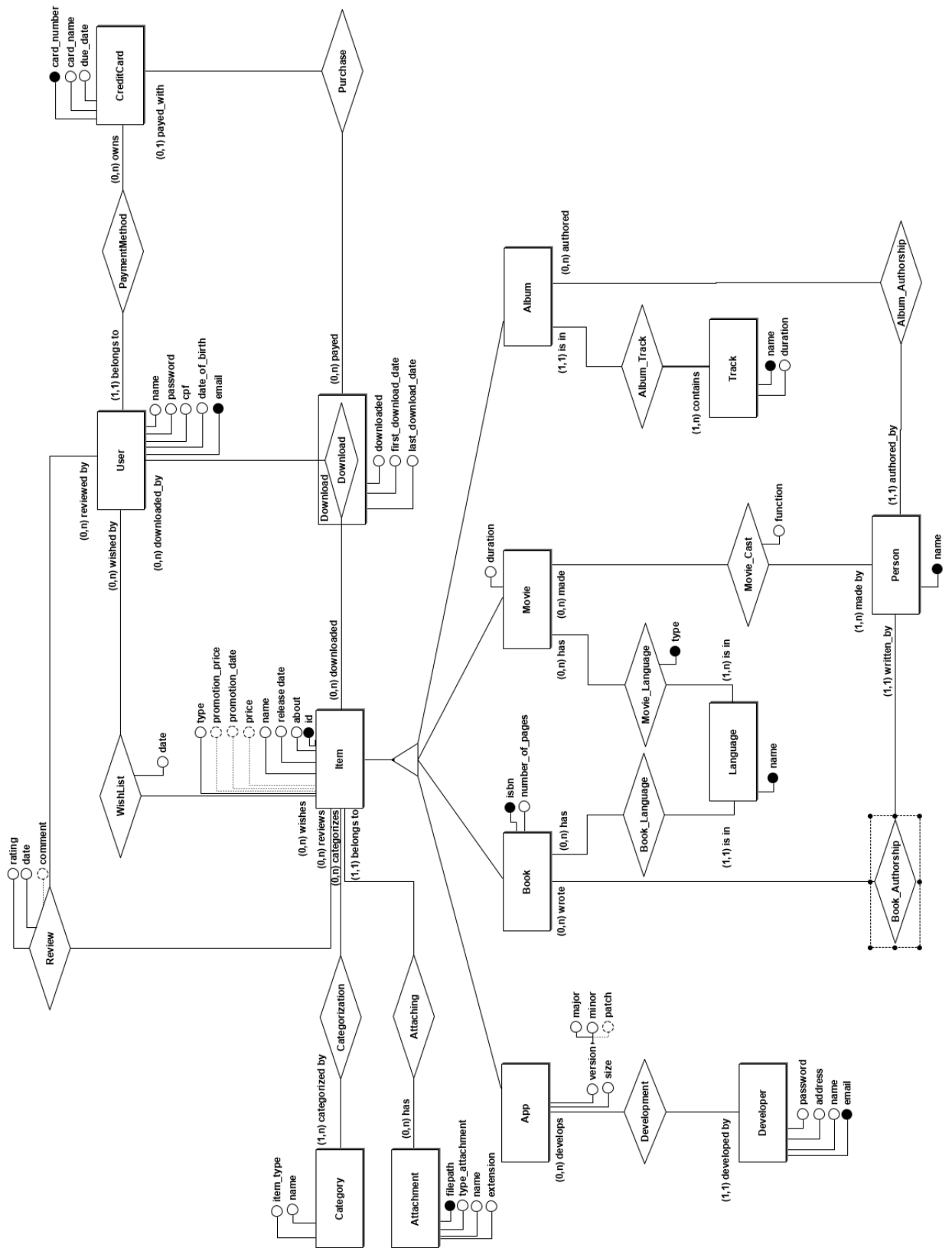
Linguagens			
Português	Inglês	Espanhol	Alemão
Italiano	Francês	Russo	Japones
Islândes	Norueguês	Sueco	Coreano

Tabela 1: Linguagens possíveis no sistema

Categorias			
<i>App</i>	<i>Filme</i>	<i>Livro</i>	<i>Álbum</i>
Arte e Design	Ação e Aventura	Arte	Alternativa
Carro e Veículos	Animação	Terror	Blues
Beleza	Comédia	Biografias	MPB
Esportes	Crime	Finanças	Infantil
Negócio	Documentário	Cozinha	Classica
Tempo	Drama	Educação	Country
Comunicação	Família	Ficção	Eletrônica
Namoro	Terror	História	Folk
Educação	Mistério	Humor	Rap
Entretenimento	Suspense	Psicologia	Jazz
Eventos	Ficção Científica	Religião	Metal
Família	Fantasia	Romance	Pop
Finanças	Esportes	Fantasia	Reggae
Social		Auto-ajuda	Rock
Viagens			Sertanejo
Saúde e Ginástica			Samba
Mapas e Navegação			Rock Nacional
Fotografia			Indie

Tabela 2: Categorias (com seus respectivos tipos) possíveis no sistema

Modelo Conceitual



Dicionário de Dados

O dicionário de dados será itemizado abaixo, com cada tabela representando uma entidade (ou relacionamento com atributos) presente na modelagem conceitual.

Atributos em **negrito** definem que eles representam chaves candidatas das entidades. Atributos com um asterisco indicam que eles são opcionais. Todos os tipos descritos abaixo são referentes aos tipos que o PostgreSQL implementa em seu SGBD.

Item			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
id	uuid	Identificador único para o item	704ef290-...
price*	numeric(5,2)	Valor a ser cobrado por um item, em reais, sendo o valor limitado em R\$ 999,99. Caso esse campo esteja nulo, entendemos que o produto está de graça.	9.99
name	varchar(50)	Nome do item, com no máximo 50 caracteres	Super SGBD
release_date	date	Data de lançamento do item	31/12/2018
about	varchar(1000)	Descrição sobre o item, podendo conter até no máximo 1000 caracteres	Lorem ipsum...
promotion_date*	date	Informa se o item está ou não em promoção. Se campo for diferente de null, e possuir uma data nele, sabemos que o item estará em promoção ATÉ aquela data. Se a data atual for maior do que a descrita no campo, ou ele for NULL, então o produto NÃO está em promoção.	19/07/2019
promotion_price*	numeric(5,2)	Valor a ser cobrado por um item, em reais, caso o atributo is_promotion seja true, sendo o valor limitado em R\$ 999,99. Caso esse campo esteja nulo, enquanto is_promotion esteja em true, entendemos que o produto está de graça.	
type	char(5)	Referencia qual tipo de Item esse Item realmente é. É utilizado para sabermos em qual outra tabela da especificação se encontram o resto dos dados (já que a generalização/especificação da tabela Item é total). Pode assumir os valores { app, movie, book, album}	app

Tabela 3: Tabela Item e a descrição textual de seus atributos

Attachment			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
filepath	varchar(100)	Caminho para onde o anexo está guardado localmente	./downloads/attach_001.jpg
type_attachment	char(6)	Tipo de anexo, sendo que os possíveis valores são {imagem, video}	imagem
name	varchar(50)	Nome dado ao anexo, para ser utilizado na hora de mostrarmos o mesmo na UI.	Imagem 01
extension	char(5)	Extensão dada para o arquivo	jpg
app	uuid	Referência à tabela Item, representando o relacionamento Attaching. Cada um dos Attachments pertence a somente um Item.	704ef290-...

Tabela 4: Tabela Attachment e a descrição textual de seus atributos

User			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
email	varchar(50)	E-mail do usuário, válido de acordo com a RFC 5322[3]	example@example.com
name	varchar(80)	Nome do usuário, com no máximo 80 caracteres	John Doe
password	varchar	Senha do usuário, guardada com criptografia SHA512[4]	1253AB...
cpf	char(11)	CPF do usuário, armazenado sem os caracteres separadores	99999999999
birthdate	date	Data de nascimento do usuário	02/01/1998

Tabela 5: Tabela User e a descrição textual de seus atributos

Credit Card			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
card_number	char(16)	Identificador do cartão de crédito	1111 1111 1111 1111
card_name	varchar(80)	Nome do dono do cartão, que pode ser diferente do User relacionado a esse Credit Card	John Doe
due_date	date	Data de vencimento do cartão	31/12/2025
user	varchar(50)	Referência à tabela User, representando o relacionamento PaymentMethod. Cada CreditCard pertence a apenas um usuário.	example@example.com

Tabela 6: Tabela CreditCard e a descrição textual de seus atributos

WishList			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
user	varchar(50)	Referência à tabela User, representando o relacionamento dele com o Relacionamento WishList (representado por essa tabela). Cada entrada na WishList pertence a um e somente um usuário.	example@example.com
app	uuid	Referência à tabela Item, representando o relacionamento dele com o Relacionamento WishList (representado por essa tabela). Cada entrada na WishList pertence a um e somente um item.	704ef290-...
date	date	Data na qual o Item foi adicionado à WishList do User	12/01/2019

Tabela 7: Tabela WishList e a descrição textual de seus atributos

Review			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
user	varchar(50)	Referência à tabela User, representando o relacionamento dele com o Relacionamento Review (representado por essa tabela). Cada entrada em Review pertence a um e somente um usuário.	example@example.com
item	uuid	Referência à tabela Item, representando o relacionamento dele com o Relacionamento Review (representado por essa tabela). Cada entrada em Review pertence a um e somente um item.	704ef290-...
rating	smallint	Nota atribuída pelo User ao Item. Seus valores possíveis são {0, 1, 2, 3, 4, 5}	4
date	date	Data que a review aconteceu	22/02/2019
comment*	varchar(256)	Comentário opcional do User referente ao Item	Top

Tabela 8: Tabela Review e a descrição textual de seus atributos

Download			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
item	uuid	Referência à tabela Item, representando o relacionamento dele com a Entidade Associativa Download (representado por essa tabela). Cada entrada em download pertence a um e somente um item.	704ef290-...
user	varchar(50)	Referência à tabela User, representando o relacionamento dele com a Entidade Associativa Download (representado por essa tabela). Cada entrada em download pertence a um e somente um user.	example@example.com
credit_card*	char(16)	Referência à tabela CreditCard, representando o relacionamento Purchase, representando qual CreditCard o User utilizou para realizar esse download. Esse campo só está preenchido se quando o User realizou o download do Item, ele possuía price ou promotion_price (caso em promoção) diferente de null ou 0.	1111 1111 1111 1111
first_download_date	date	Data na qual o item foi baixado pela primeira vez	18/01/2019
last_download_date	date	Data na qual o item foi baixado pela última vez. É sempre maior ou igual a first_download_date.	18/01/2019
downloaded	boolean	Informa se o item está instalado (true) ou desinstalado (false) no momento	false

Tabela 9: Tabela Download e a descrição textual de seus atributos

Language			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
name	varchar(20)	Nome da linguagem, com no máximo 20 caracteres	italiano

Tabela 10: Tabela Language e a descrição textual de seus atributos

Developer			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
email	varchar(50)	E-mail do(a) desenvolvedor(a), validado de acordo com a RFC 5322[3]	developer@example.com
name	varchar(80)	Nome do(a) desenvolvedor(a), com no máximo 80 caracteres	Developer Doe
address	varchar	Endereço do(a) desenvolvedor(a) contendo rua, número, bairro, cidade e país, ou conforme for o padrão do país de origem do(a) desenvolvedor(a)	..., Havana, Cuba
password	varchar	Senha do developer, guardada com criptografia SHA512[4]	8269bf0a...

Tabela 11: Tabela Developer e a descrição textual de seus atributos

Person			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
name	varchar(80)	Representa o nome de uma pessoa que pode ter escrito algum Book, participado de algum Movie ou sido a criadora de algum Album. Nome pode possuir no máximo 80 caracteres	Person Doe

Tabela 12: Tabela Artist e a descrição textual de seus atributos

Category			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
name	varchar(25)	Nome da categoria, sendo único dentro do mesmo item_type	Ficção Científica
item_type	char(5)	Tipo de item à qual essa categoria se aplica. Os valores possíveis são {app, livro, filme, album}. Para o mesmo item_type, não podem haver 2 categorias com o mesmo nome. Isso significa que item_type e name juntos , identificam uma categoria.	filme

Tabela 13: Tabela Category e a descrição textual de seus atributos. Mais informações sobre ela na Tabela 2.

Categorization			
Atributo	Tipo	Descrição	Exemplo
category_name	varchar(25)	Referência à tabela Category, representando o relacionamento dele com o Relacionamento Categorization (representado por essa tabela). Cada entrada em Categorization pertence a um e somente uma categoria, que possui esse nome e o tipo descrito no campo abaixo.	Ficção Científica
category_item_type	char(5)	Referência à tabela Category, representando o relacionamento dele com o Relacionamento Categorization (representado por essa tabela). Cada entrada em Categorization pertence a um e somente uma categoria, que possui o nome descrito acima e esse tipo. O Item referenciado por essa Categorization precisa ser do mesmo tipo que o descrito nesse campo.	filme
item	uuid	Referência à tabela Item, representando o relacionamento dele com o Relacionamento Categorization (representado por essa tabela). Cada entrada em Categorization pertence a um e somente um Item.	704ef290-...

Tabela 14: Tabela Categorization e a descrição textual de seus atributos

App (que especifica a tabela Item)			
Atributo	Tipo	Descrição	Exemplo
item	uuid	Referência à tabela Item, representando o relacionamento de especialização com Item. Cada App pertence a apenas um item. Esse relacionamento é utilizado para termos acesso às outras informações de App.	704ef290-...
developer	varchar(50)	Referência à tabela Developer, representando o relacionamento Development. Cada App possui apenas um developer.	developer@example.com
size	bigint	Tamanho do aplicativo, em bytes, limitado a 1TB (1099511627800)	1048576
version	varchar	Texto demonstrando a versão do app no formato "MAJOR.MINOR.PATCH", sendo que MAJOR, MINOR e PATCH são números inteiros, sem vírgulas, e PATCH pode ser omitido juntamente com o ponto anterior à ele	2.1.12

Tabela 15: Tabela App (que especifica a tabela Item) e a descrição textual de seus atributos

Album (que especifica a tabela Item)			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
item	uuid	Referência à tabela Item, representando o relacionamento de especialização com Item. Cada Album pertence a apenas um item. Esse relacionamento é utilizado para termos acesso às outras informações de Album.	704ef290-...
artist	varchar(80)	Referência à tabela Person, representando o relacionamento Album_Authorship. Cada Album pertence a apenas um criador de álbum.	Artist Doe

Tabela 16: Tabela Album (que especifica a tabela Item) e a descrição textual de seus atributos

Track			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
item	uuid	Referência à tabela Album, representando o relacionamento Album_Track. Cada Track pertence a apenas um Album. O artista que criou essa música pode ser acessado através do álbum.	704ef290-...
name	varchar(50)	Nome da música, sendo única dentro de um álbum. Possui no máximo 50 caracteres	Juntos e Shallow Now
duration	int	Duração da música em segundos	205

Tabela 17: Tabela Track e a descrição textual de seus atributos

Book (que especifica a tabela Item)			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
item	uuid	Referência à tabela Item, representando o relacionamento de especialização com Item. Cada Book pertence a apenas um item. Esse relacionamento é utilizado para termos acesso às outras informações de Book.	704ef290-...
author	varchar(80)	Referência à tabela Person, representando o relacionamento Book_Authorship. Cada Book pertence a apenas um escritor.	Author Doe
language	varchar(80)	Referência à tabela Language, representando o relacionamento Book_Language. Cada Livro possui apenas uma linguagem. O mesmo livro, em diferentes languages, são representados como diferentes livros.	Italiano
number_of_pages	smallint	Quantidade de páginas no livro	425
<u>ISBN</u>	char(13)	ISBN (código identificador) do livro[5]	9788533302273

Tabela 18: Tabela Book (que especifica a tabela Item) e a descrição textual de seus atributos

Movie (que especifica a tabela Item)			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
item	uuid	Referência à tabela Item, representando o relacionamento de especialização com Item. Cada Movie pertence a apenas um item. Esse relacionamento é utilizado para termos acesso às outras informações de Movie.	704ef290-...
duration	int	Quantidade de minutos do filme	123

Tabela 19: Tabela Movie (que especifica a tabela Item) e a descrição textual de seus atributos

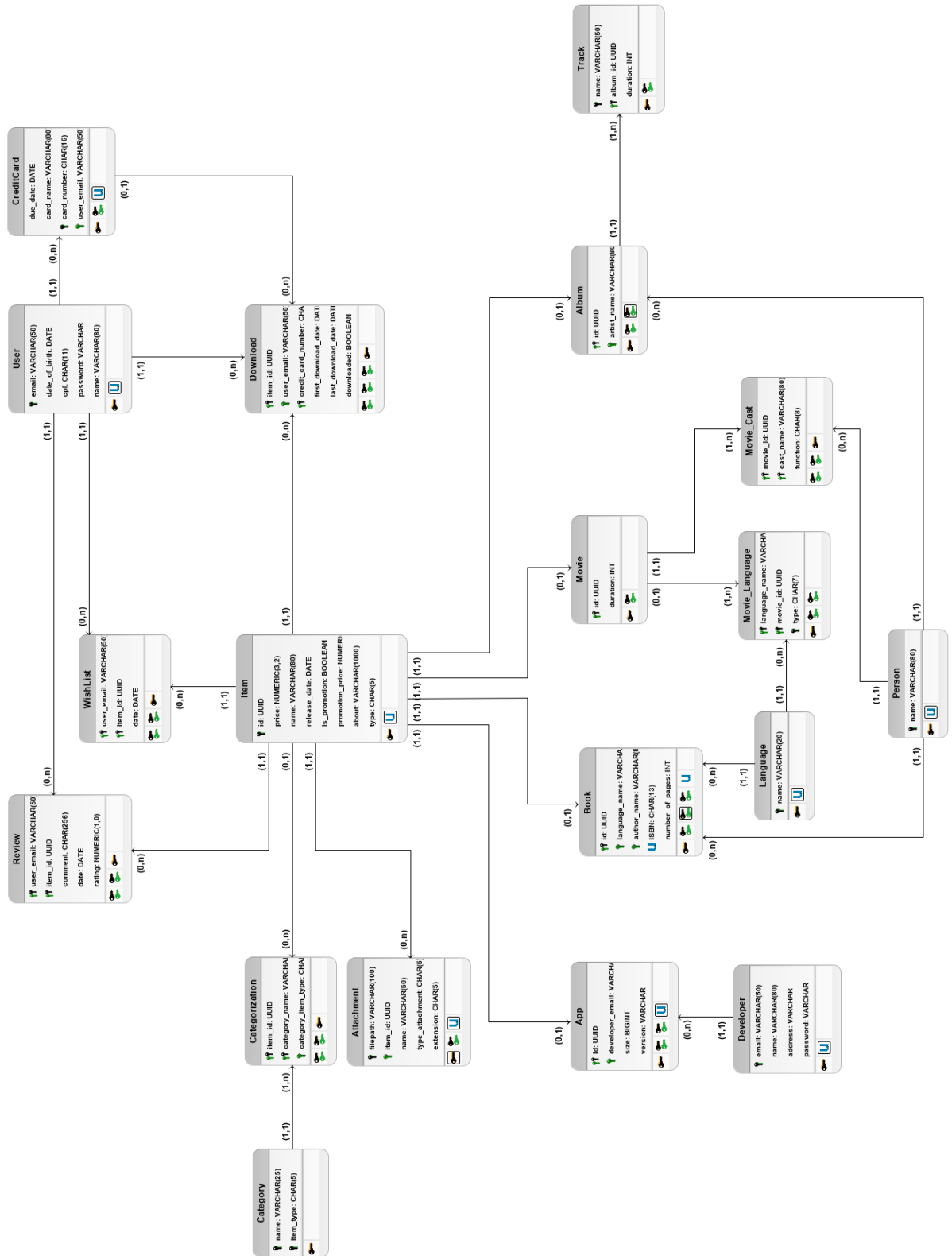
Movie _ Language			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
movie	uuid	Referência à tabela Movie, representando o relacionamento dele com o Relacionamento Movie _ Language (representado por essa tabela). Cada entrada em Movie _ Language se refere a um e somente um Movie. Cada Movie pode ter várias languages, desde que não seja descrito duas vezes a mesma linguagem com o mesmo tipo.	704ef290-...
language _ name	varchar(80)	Referência à tabela Language, representando o relacionamento dele com o Relacionamento Movie _ Language (representado por essa tabela). Cada entrada em Movie _ Language se refere a um e somente uma Linguagem.	Italiano
type	char(7)	Representa se essa linguagem para esse filme é relacionada à legenda ou ao áudio. Os únicos valores permitidos são {audio, legenda}	legenda

Tabela 20: Tabela Movie _ Language e a descrição textual de seus atributos

Movie _ Cast			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
movie _ id	uuid	Referência à tabela Movie, representando o relacionamento dele com o Relacionamento Movie _ Cast (representado por essa tabela). Cada entrada em Movie _ Cast se refere a um e somente um Movie.	704ef290-...
cast _ name	varchar(80)	Referência à tabela Person, representando o relacionamento dele com o Relacionamento Movie _ Language (representado por essa tabela). Cada entrada em Movie _ Language se refere a um e somente uma Person. Essas pessoas podem participar de filmes, ter escrito algum livro ou produzido algum álbum;	Cast Doe

Tabela 21: Tabela Movie _ Cast e a descrição textual de seus atributos

Modelo Lógico



Descrição do Mapeamento (Conceitual para Lógico)

Segue uma lista de mapeamentos implementados, onde os atributos, entidades e relacionamentos que estão escritos com *itálico*, são campos adicionais que não estavam no modelo conceitual e os que são definidos como "Mapeamento direto"querem dizer que estão da mesma maneira como foram definidos no conceitual.

Entidades:

- User: Mapeamento direto para **User**.
 - name: Mapeamento direto para **User.name**, obrigatório.
 - email: Mapeamento direto para **User.email**, obrigatório e chave primária.
 - date_of_birth: Mapeamento direto para **User.date_of_birth**, obrigatório.
 - password: Mapeamento direto para **User.password**, obrigatório.
- CreditCard: Mapeamento direto para **CreditCard**.
 - due_date: Mapeamento direto para **CreditCard.due_date**, obrigatório.
 - card_name: Mapeamento direto para **CreditCard.card_name**, obrigatório.
 - card_number: Mapeamento direto para **CreditCard.card_number**, obrigatório e chave primária.
 - user_email: Identificador do usuário, por conta do mapeamento de relacionamento por colunas adicionais no relacionamento PaymentMethod. Mapeado para **CreditCard.user_email**, obrigatório e chave estrangeira.
- Item: Como é uma generalização total e exclusiva, o mapeamento foi feito seguindo a Alternativa 1 dos slides de mapeamento(implementação por diferentes relações com uma relação para dados comuns). Foi usada como chave estrangeira (única) na entidade especializada a chave primária da entidade generalizada, fazendo com que sempre que ocorre uma entidade especializada, exista uma entidade generalizada que ainda não foi relacionada. Mapeada para **Item**.
 - id: Valor adicional criado para ser a chave primária do item, pois não existe nenhuma chave candidate que, sozinha, seja determinante para o Item. Como esse valor terá que ser usado como chave estrangeira na entidade especializada, foi decidido que seria lógico usar uma chave primária não-composta, por conta do menor espaço ocupado e facilidade. Mapeado para **Item.id**, obrigatório e chave primária, sendo do tipo UUID, automaticamente gerado pelo banco de dados.
 - name: Mapeamento direto para **Item.name**, obrigatório.
 - price: Mapeamento direto para **Item.price**, opcional.
 - release_date: Mapeamento direto para **Item.release_date**, obrigatório.
 - promotion_date: Mapeamento direto para **Item.promotion_date**, opcional.
 - promotion_price: Mapeamento direto para **Item.promotion_price**, opcional.
 - about: Mapeamento direto para **Item.about**, obrigatório.
- App: Entidade especializada de item, onde sua chave primária é uma chave estrangeira para item, já que, como dito anteriormente, foi utilizada a Alternativa 1 dos slides de mapeamento de generalização. Mapeada para **App**.

- *item_id*: É uma chave estrangeira para Item, ao mesmo tempo em que é chave primária, fazendo com que seja único o relacionamento com Item e não exista nenhum outro App relacionado com o mesmo. Criado Mapeamento direto para **App.item_id**, obrigatório e chave primária.
 - *developer_email*: É uma chave estrangeira para Developer. Mapeado para **App.developer_email**, obrigatório.
 - *size*: Mapeamento direto para **App.size**, obrigatório.
 - *version*: Mapeamento direto para **App.version**, obrigatório.
- Book: Entidade especializada de item, onde sua chave primária é uma chave estrangeira para item, já que, como dito anteriormente, foi utilizada a Alternativa 1 dos slides de mapeamento de generalização. Mapeada para **Book**.
 - *item_id*: É uma chave estrangeira para Item, ao mesmo tempo em que é chave primária, fazendo com que seja único o relacionamento com Item e não exista nenhum outro App relacionado com o mesmo. Mapeamento direto para **Book.item_id**, obrigatório e chave primária.
 - *author_name*: Chave estrangeira para a tabela Author. Criado por conta do mapeamento de relacionamento por colunas adicionais no relacionamento Book_Authorship. Mapeado para **Book.author_name**, obrigatório.
 - *language_name*: Chave estrangeira para a tabela Language. Criado por conta do mapeamento de relacionamento por colunas adicionais no relacionamento Book_Language. Mapeado para **Book.language_name**, obrigatório.
 - *isbn*: Mapeamento direto para **Book.isbn**, obrigatório e único.
 - *number_of_pages*: Mapeamento direto para **Book.number_of_pages**, obrigatório.
 - Movie: Entidade especializada de item, onde sua chave primária é uma chave estrangeira para item, já que, como dito anteriormente, foi utilizada a Alternativa 1 dos slides de mapeamento de generalização. Mapeada para **Movie**.
 - *item_id*: É uma chave estrangeira para Item, ao mesmo tempo em que é chave primária, fazendo com que seja único o relacionamento com Item e não exista nenhum outro App relacionado com o mesmo. Mapeamento direto para **Movie.item_id**, obrigatório e chave primária.
 - *duration*: Mapeamento direto para **Movie.duration**, obrigatório.
 - Album: Entidade especializada de item, onde sua chave primária é uma chave estrangeira para item, já que, como dito anteriormente, foi utilizada a Alternativa 1 dos slides de mapeamento de generalização. Mapeada para **Album**.
 - *item_id*: É uma chave estrangeira para Item, ao mesmo tempo em que é chave primária, fazendo com que seja único o relacionamento com Item e não exista nenhum outro App relacionado com o mesmo. Mapeamento direto para **Album.item_id**, obrigatório e chave primária.
 - *artist_name*: É uma chave estrangeira para Album. Criado por conta do mapeamento de relacionamento por colunas adicionais no relacionamento Album_Authorship. Mapeado para **Album.artist_name**, obrigatório.

- **Developer:** Mapeamento direto para **Developer**.
 - email: Mapeamento direto para **Developer.email**, obrigatório e chave primária, pois é a única chave candidata.
 - name: Mapeamento direto para **developer.name**, obrigatório.
 - address: Mapeamento direto para **Developer.address**, obrigatório.
 - password: Mapeamento direto para **Developer.password**, obrigatório.
- **Author:** Mapeamento direto para **Author**.
 - name: Mapeamento direto para **Author.name**, obrigatório e chave primária, pois é a única chave candidata, além de ser o único atributo.
- **Language:** Mapeamento direto para **Language**.
 - name: Mapeamento direto para **Language.name**, obrigatório e chave primária, pois é a única chave candidat, além de ser o único atributoa.
- **Cast:** Mapeamento direto para **Cast**.
 - name: Mapeamento direto para **Cast.name**, obrigatório e chave primária, pois é a única chave candidata, além de ser o único atributo.
- **Track:** Mapeamento direto para **Track**.
 - name: Mapeamento direto para **Track.name**, obrigatório e chave primária juntamente com **Track.album_id**, já que **Track** é uma entidade fraca.
 - *album_id*: Mapeamento direto para **Track.album_id**, obrigatório e chave primária juntamente com **Track.name**, já que **Track** é uma entidade fraca em relação a **Album**. Criado por conta do mapeamento de relacionamento por colunas adicionais no relacionamento Album_Track.
 - duration: Mapeamento direto para **Track.duration**, obrigatório e não nulo.
- **Attachment:** Mapeamento direto para **Attachment**
 - filepath: Mapeamento direto para **Attachment.filepath**, obrigatório e não nulo, é a chave primária da tabela por ser a única chave candidata.
 - *item_id*: Mapeamento direto para **Attachment.item_id**, obrigatório e não nulo. Criado por conta do mapeamento de relacionamento por colunas adicionais no relacionamento Attaching. É uma chave estrangeira para a tabela Item.
 - name: Mapeamento direto para **Attachment.name**, obrigatório e não nulo.
 - type_attachment: Mapeamento direto para **Attachment.type_attachment**, obrigatório e não nulo.
 - extension: Mapeamento direto para **Attachment.extension**, obrigatório e não nulo.
- **Category:** Mapeamento direto para **Category**
 - name: Mapeamento direto para **Category.name**, obrigatório e não nulo, é unico juntamente com **Category.item_type**, assim como chave primária juntamente com ele.
 - item_type: Mapeamento direto para **Category.item_type**, obrigatório e não nulo, é unico juntamente com **Category.name**, assim como chave primária, juntamente com ele.

Relacionamentos:

- **PaymentMethod**: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1: N . Com a chave estrangeira em **CreditCard**.
- **Review**: Mapeamento de relacionamento por tabela própria, pois é um relacionamento $N:M$.
 - *user_email*: Chave primária do usuário, por ser uma tabela de relação que liga usuário a itens. Mapeado para **Review.user_email**, obrigatório e chave estrangeira.
 - *item_id*: Chave primária do item, pelo mesmo motivo do atributo *user_email*. Mapeado para **Review.item_id**, obrigatório e chave estrangeira.
 - *comment*: Mapeamento direto para **Review.comment**, opcional.
 - *date*: Mapeamento direto para **Review.date**, obrigatório.
 - *rating*: Mapeamento direto para **Review.rating**, obrigatório.
- **WishList**: Mapeamento de relacionamento por tabela própria, pois é um relacionamento $N:M$.
 - *user_email*: Chave primária do usuário, por ser uma tabela de relação que liga usuário a itens. Mapeado para **WishList.user_email**, obrigatório e chave estrangeira.
 - *item_id*: Chave primária do item, pelo mesmo motivo do atributo *user_email*. Mapeado para **WishList.item_id**, obrigatório e chave estrangeira.
 - *date*: Mapeamento direto para **WishList.date**, obrigatório.
- **Download**: Mapeamento de relacionamento por tabela própria, pois é um relacionamento ternário. Foi implementada uma tabela com referência (chave estrangeira) às chaves primárias de **User**, **CreditCart** e **Item**.
 - *item_id*: Chave primária do item, que está fazendo parte da chave composta que identifica o próprio **Download**. Mapeado para **Download.item_id**, obrigatório e chave estrangeira.
 - *user_email*: Chave primária do usuário, mapeado para **WishList.user_email**, obrigatório e chave estrangeira.
 - *first_downloaded_date*: Mapeamento direto para **Download.first_downloaded_date**, obrigatório.
 - *last_downloaded_date*: Mapeamento direto para **Download.last_downloaded_date**, obrigatório.
 - *downloaded*: Mapeamento direto para **Download.downloaded**, obrigatório.
- **Development**: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1: N . Com a chave estrangeira em **App**.
- **Categorization**: Mapeamento de relacionamento por tabela própria, pois é um relacionamento $N:M$.
 - *item_id*: Chave estrangeira para **Item** e que está fazendo parte da chave primária composta. Mapeada para **Categorization.item_id**, obrigatória.

- *category_name* e *category_item_type*: Chave estrangeira composta de Category, onde *category_name* faz parte da chave composta junto com *item_id* em Categorization. Mapeados respectivamente para **Categorization.category_name** e **Category.category_item_type**
- Attaching: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1:*N*. Com a chave estrangeira em Attachment.
- Book_Authorship: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1:*N*. Com a chave estrangeira em Book.
- Book_Language: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1:*N*. Com a chave estrangeira em Book.
- Movie_Language: Mapeamento de relacionamento por tabela própria, pois é um relacionamento *N:M*.
 - *type*: Mapeamento direto para **Movie_Language.type**, obrigatório e parte da chave primária composta.
 - *language_name*: Chave estrangeira para Language e que está fazendo parte da chave primária composta. Mapeada para **Movie_Language.language_name**, obrigatória.
 - *movie_id*: Chave estrangeira para Movie e que está fazendo parte da chave primária composta. Mapeada para **Movie_Language.movie_id**, obrigatória.
- Movie_Cast: Mapeamento de relacionamento por tabela própria, pois é um relacionamento *N:M*.
 - *cast_name*: Chave estrangeira para Language e que está fazendo parte da chave primária composta. Mapeada para **Movie_Cast.cast_name**, obrigatória.
 - *movie_id*: Chave estrangeira para Movie e que está fazendo parte da chave primária composta. Mapeada para **Movie_Cast.movie_id**, obrigatória.
 - *function*: Mapeamento direto para **Movie_cast.function**, obrigatório.
- Album_Track: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1:*N*. Com a chave estrangeira em Track.
- Album_Authorship: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1:*N*. Com a chave estrangeira em Album.

Relatório do Programa

A linguagem de programação escolhida para desenvolver o programa foi Python[6], juntamente com seu web framework Flask[7]. Ele é também chamado de microframework porque mantém um núcleo simples mas extensível. Não há uma camada de abstração do banco de dados, validação de formulários, ou qualquer outro componente onde bibliotecas de terceiros existem para prover a funcionalidade. Fazendo com que não se tornem obscuros os detalhes de conexão com a base de dados e satisfazendo um dos tópicos do enunciado.

- Conexão com o banco de dados: Usando a biblioteca *psycopg2*, foi feita a conexão com o banco de dados, segundo o código abaixo:

```
1 def get_db():
2     if "db" not in g:
3         print("[INFO] Creating DB connection")
4         g.db = psycopg2.connect(database='GooglePlay', user="postgres",
5                                 password="postgres")
6
7         return g.db
```

A partir da conexão é criado um cursor:

```
1 def get_cursor():
2     if "cursor" not in g:
3         print("[INFO] Creating cursor")
4         g.cursor = get_db().cursor()
5
6     return g.cursor
7
```

Onde ambos são únicos para cada requisição e serão reusados se forem chamados novamente.

- Preparação e envio das consultas ao banco de dados e processamento de retorno das consultas: Todas as consultas no banco de dados são feitas conforme o código abaixo, onde é construída uma query SQL e feita uma chamada da função `run_query_parameter()`, que recebe como parametro uma query SQL e retorna um vetor de vetores. O código trata das exceções que acontecem durante a conexão com o banco de dados, retornando a exceção bem formatada, que pode ser tratada pela parte da aplicação que fez a requisição de consulta.

```
1 def promotion_film_members():
2     """
3     Retorna o nome e a funcao dos membros de filmes que estejam na
4     promocao.
5     """
6     try:
7         query = run_select_query('SELECT movie_cast.cast_name, "function
8
9         FROM movie_with_item item \
10        JOIN movie_cast ON (item.id =
11
12        movie_cast.movie_id) \
13
14        WHERE promotion_date is not null and
15        CURRENT_TIMESTAMP(0) <= promotion_date;')
16     except (Exception, psycopg2.Error) as error:
```

```

12         return {"status": "error", "error": error}
13

```

Cada vetor, dentro do vetor de vetores, representa uma linha de dados retornada da consulta feita. Por exemplo: O retorno recebido no código anterior é: `[["name1", "function1"], ["name2", "function2"], ..., ["nameX", "functionX"]]`.

- Detalhes da preparação, envio e processamento de retorno dos comandos de atualização: A inserção de novos elementos é feita segundo o código abaixo, onde é recebido como parametro os dados referente a tabela que deseja inserir e é chamado o cursor junto com a função `execute()`. Dentro da função `execute`, é passada como parametro a query SQL, junto com os dados recebidos pela função.

```

1  def add_categorization(item_uuid, category_name, category_type):
2      """
3      Tenta adicionar uma categorization de 'category_name' do tipo '
4      category_type'
5      no Item de uuid 'item_uuid'
6      """
7      try:
8          record_to_insert = (item_uuid, category_name, category_type)
9          cursor = db.get_cursor()
10         cursor.execute('INSERT INTO categorization VALUES (%s, %s, %s);'
11         , record_to_insert)
12         db.get_db().commit()
13
14         return {'row_count': cursor.rowcount, 'status': 'Record inserted
15         successfully into categorization table', 'error': ''}
16     except (Exception, psycopg2.Error) as error:
17         return {'row_count': 0, "status": "error", "error": error}

```

O código trata das exceções que acontecem durante a conexão com o banco de dados e a inserção da nova instância, retornando uma exceção, que pode ser tratada da mesma forma que mencionado no tópico anterior.

Referências

- [1] From Google owned by Alphabet Inc., all rights reserved, 2015-2019. Available from World Wide Web: (<https://play.google.com/store>)
- [2] Owned by Alphabet Inc., all rights reserved, 2015-2019. Available from World Wide Web (<https://abc.xyz/>)
- [3] Defined in Email address (Wikipedia). Accessed in 31/05/2019. Available from World Wide Web (https://en.wikipedia.org/wiki/Email_address)
- [4] SHA-2 cryptography, as defined at Wikipedia. Accessed in 31/05/2019. Available from World Wide Web (<https://en.wikipedia.org/wiki/SHA-2>)
- [5] International Standard Book Number. Accessed in 31/05/2019. Available from World Wide Web (<http://www.isbn.bn.br/website/>)
- [6] <https://www.python.org>
- [7] <http://flask.pocoo.org/>