



**Universidade
Europeia**

LAUREATE INTERNATIONAL UNIVERSITIES

Documentação de Base de Dados

IADE - Faculdade de Design, Tecnologia e Comunicação

Engenharia Informática — 3.º semestre

Felipe Campelo Sabbado (20191012); Leonardo Lage (20200859)

Bases de Dados (Miguel Boavida)

19 de novembro de 2021

Dicionário de Dados

Tabela tag_types

Nome	Descrição	Tipo de dados	Tamanho	Restrições
tt_id	Id do tipo de tag	serial		Pk
tt_name	Nome do tipo de tag (ex.: Sistema, User, ...)	varchar	30	not null

Tabela tags

Nome	Descrição	Tipo de dados	Tamanho	Restrições
tg_id	Id da tag	serial		Pk
tg_name	Nome da tag (ex.: Natureza, Animais, ...)	varchar	30	not null
tg_tt_id	id do tipo de tag	int		Fk

Tabela type_users

Nome	Descrição	Tipo de dados	Tamanho	Restrições
tu_id	Id do tipo de user	serial		Pk
tu_name	Nome do tipo (ex.: Novato, desbravador, especialista...)	varchar	30	not null

Tabela users

Nome	Descrição	Tipo de dados	Tamanho	Restrições
us_id	Id do user	serial		Pk
us_name	Nome do user	varchar	60	not null
us_bdate	Data de aniversário	date		not null
us_gender	Gênero do user	char	1	not null
us_email	Email do user	varchar	30	
us_country	País do user	varchar	30	
us_bio	Uma breve descrição de si mesmo	varchar	400	
us_dist	Distância que o user já percorreu	float		default 0
us_tu_id	Nível do utilizador	int		Fk, default 2

Tabela spots

Nome	Descrição	Tipo de dados	Tamanho	Restrições
sp_id	Id da atração	serial		Pk
sp_name	Nome da atração	varchar	60	not null
sp_lat	Latitude da atração	float		not null
sp_long	Longitude da atração	float		not null
sp_price	Saber se a atração é paga ou não	boolean		
us_bio	Uma breve descrição da atração	varchar	400	

Tabela spot_evaluations

Nome	Descrição	Tipo de dados	Tamanho	Restrições
se_id	Id da avaliação da atração	serial		Pk
se_rate	Rate de 1 a 5 da atração	int		not null
se_comment	Comentário de um user sobre a atração	varchar	400	
se_us_id	Que user comentou	int		Fk, not null
se_sp_id	Qual foi a atração	int		Fk, not null

Tabela users_spots_tags

Nome	Descrição	Tipo de dados	Tamanho	Restrições
ust_id	Id da do user spot tags	serial		Pk
ust_sp_id	qual o spot	int		not null
ust_tg_is	Qual a tag	int		not null
ust_us_id	Qual o user	int		not null

Tabela fav_spots

Nome	Descrição	Tipo de dados	Tamanho	Restrições
fs_id	Id de um spot favorito de um user	serial		Pk
fs_us_id	De que user	int		Fk, not null
fs_sp_id	Qual a atração	int		Fk, not null

Tabela done_spots

Nome	Descrição	Tipo de dados	Tamanho	Restrições
ds_id	id do spot feito	serial		Pk
ds_us_id	Qual o user	int		Fk, not null
ds_sp_id	Qual o spot	int		Fk, not null

Tabela routes

Nome	Descrição	Tipo de dados	Tamanho	Restrições
rt_id	Id da rota	serial		Pk
rt_name	Nome da rota	varchar	60	
rt_dist	Distância total da rota	float		
rt_bio	Uma breve descrição da atração	varchar	400	

Tabela fav_routes

Nome	Descrição	Tipo de dados	Tamanho	Restrições
fr_id	Id da rota favorita de um user	serial		Pk
fr_us_id	De que user	int		Fk, not null
fr_rt_id	Qual a rota	int		Fk, not null

Tabela route_evaluations

Nome	Descrição	Tipo de dados	Tamanho	Restrições
re_id	Id da avaliação da rota	serial		Pk
re_rate	Rate de 1 a 5 da rota	int		not null
re_comment	Comentário de um user sobre a rota	varchar	400	
re_us_id	Que user comentou	int		Fk, not null
re_rt_id	Qual foi a rota	int		Fk, not null

Tabela done_routes

Nome	Descrição	Tipo de dados	Tamanho	Restrições
dr_id	id da rota feita de um utilizador	serial		Pk
dr_us_id	De que user	int		Fk, not null
dr_rt_id	Qual a rota	int		Fk, not null

Tabela progress_routes

Nome	Descrição	Tipo de dados	Tamanho	Restrições
ua_id	Id da medalha de um user	int		Pk, serial
ua_date	Quando conseguiu	date		not null
ua_us_id	De que user	int		Fk, not null
ua_ac_id	Qual foi a medalha	int		Fk, not null

Tabela routes_spots

Nome	Descrição	Tipo de dados	Tamanho	Restrições
rs_id	is do spot de uma rota	serial		Pk
rs_rt_id	Qual a rotar	int		Fk, not null
rs_sp_id	Qual o spot	int		Fk, not null

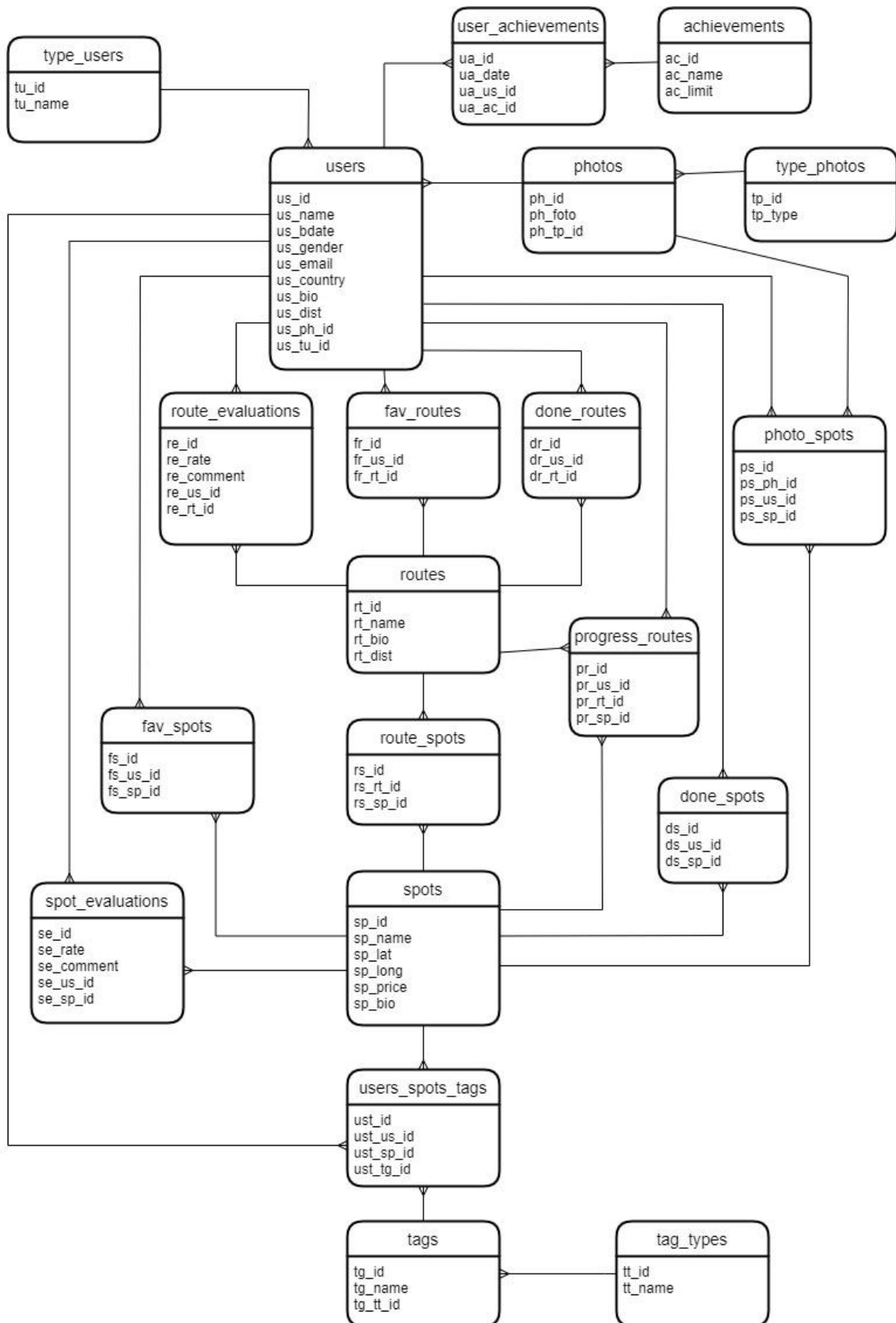
Tabela achievements

Nome	Descrição	Tipo de dados	Tamanho	Restrições
ac_id	Id da medalha	int		Pk, serial
ac_name	Nome da medalha (ex.: 1 museu, 5 museus, ...)	varchar	30	not null

Tabela user_achievements

Nome	Descrição	Tipo de dados	Tamanho	Restrições
ua_id	Id da medalha de um user	int		Pk, serial
ua_date	Quando conseguiu	date		not null
ua_us_id	De que user	int		Fk, not null
ua_ac_id	Qual foi a medalha	int		Fk, not null

Diagrama de Base de Datos



Guia de Dados

Introdução

A Base de Dados visa representar relações entre Rotas (*routes*), Pontos Turísticos (*spots*) e Utilizadores (*users*). As rotas são compostas por pelo menos dois pontos. Os utilizadores possuem tipos (*type_users*), que definem níveis de acesso diferentes, e podem avaliar (*route_evaluations* e *spot_evaluations*), favoritar (*fav_routes* e *fav_spots*) e concluir (*done_routes* e *done_spots*) tanto as rotas quanto os pontos. Enquanto as rotas não estão completas, o utilizador tem relação com o progresso da rota (*progress_routes*), que guarda a informação dos pontos ainda não concluídos. Os pontos possuem *tags* que servem para classificá-los. Há também os *achievements* que são dados aos utilizadores que cumprem certos requisitos, tais como número de pontos avaliados, quantidade de pontos e rotas concluídas, distâncias percorridas entre outros.

Forma de nomenclatura de atributos

O nome dos atributos é composto pela abreviatura da tabela a qual pertence e uma palavra que o representa, separados por *underscore*. Quando o nome de uma tabela é constituído só por uma palavra e começa com uma vogal usamos as duas primeiras letras (e.g.: *users* -> *us_name*), e quando começa com uma consoante usamos as duas primeiras consoantes (e.g.: *routes* -> *rt_id*). Quando o nome da tabela é constituído por mais que uma palavra é usada a primeira letra de cada palavra (e.g.: *fav_routes* -> *fs_rt_id*).

Tabela routes

Nesta tabela é armazenado: id (*rt_id*), nome de uma rota (*rt_name*), uma breve descrição (*rt_bio*), e a distância do primeiro ponto até ao último (*rt_dist*).

Informação dos dados:

Existem 5 rotas:

- As rotas 1 e 2 têm uma descrição;
- A rota 5 é composta apenas por pontos grátis;
- A rota 1 é a única com comentários;
- As rotas de 1 a 4 tem 7 avaliações;
- A rota 5 é a única que não tem avaliações;
- A rota 3 é a mais popular e a 2 é a menos popular;
- A rota 2 é a única que está em progresso e é pelo utilizador 2;
- A rota 1 foi a mais concluída.
- Da rota 5 para baixo ninguém as fez.

Observações: As rotas herdam as tags dos pontos que a constituem. Exemplo, se tenho um ponto 1 com a tag A e B e um ponto 2 com a tag A e C a rota constituída por esses pontos vai ter duas tags A, uma B e uma C. A tabela routes está ligada às seguintes tabelas: route_evaluations, fav_routes, done_routes, progress_routes e routes_spots.

Tabela spots

Esta tabela armazena: id (sp_id), nome (sp_name), latitude e longitude (sp_lat e sp_long), preço (sp_price) em booleano, representando se o local é pago ou não ou, se *null*, o preço é indeterminado e uma descrição (sp_bio).

Informação dos dados:

Existem 16 locais diferentes, dos quais:

- Os 12 primeiros possuem preços com *true* ou *false* e as outras estão a *null*;
- Os locais de 12 a 15 possuem comentários feitos pelo utilizador 2;
- Os locais 1 a 6, 8 e 11 a 16, possuem avaliações, sendo o 2 o mais avaliado com um total de 7 notas.
- Apenas o local 14 não é favoritos de algum utilizador;
- Vários locais são favoritos de 2 utilizadores;
- Apenas a 16 não possui tags;
- Vários locais possuem 3 tags;

Observações: Os locais possuem *tags* que servem como uma classificação, indicando tipos ao qual pertencem o local. A tabela *spot* está ligada às tabelas *route_spots*, *spot_evaluations*, *fav_spots*, *done_spots*, *users_spots_tags* e *progress_routes*.

Tabela users

Nesta tabela é armazenado: id (us_id), nome (us_name), data de nascimento (us_bdate), género (us_gender), email (us_email), país de origem (us_country), biografia (us_bio), distância já percorrida (us_dis), nível do utilizador(us_tu_id). Um user por default quando cria uma conta recebe o nível 1.

Informação dos dados:

Existem 11 utilizadores:

- Os users 2 e 3 têm biografia;
- Os users 2 a 4 indicaram o país;
- Dois são alemães, os restantes têm nacionalidades diferentes;
- Os users 2 a 6 tem email;
- Os users 2 a 4 são os que fizeram rotas;

- Os restantes utilizadores foram utilizados para dar rate aos spots;
- O user 3 foi o que deu mas tags aos spots (5 tags);
- O user 4 deu apenas 2 tags a spots.

Observações: Criamos o atributo `us_dist`, pois uma pessoa pode não acabar uma rota, assim a distância percorrida não podia ser uma soma de todas as rotas feitas. O user 1 é do tipo supremo, isto é, pode fazer tudo como, por exemplo, criar tags de sistema, criar spots e eliminar o que quiser.

Pesquisas Relevantes

Ordenar as routes de acordo com a quantidade de favoritos

```
select rt_id, rt_name, count(fr_rt_id)
from routes, fav_routes where fr_rt_id = rt_id
group by rt_name, rt_id
order by count(fr_rt_id) desc
```

Ordena as routes pela distância

```
select * from routes
order by rt_dist desc
```

Ordena os spots de acordo com a média das avaliações

```
select sp_name as spName, se_sp_id as id, avg(se_rate) as spAvg
from spot_evaluations
inner join spots on se_sp_id = sp_id
group by sp_name, se_sp_id
order by spAvg desc
```

Seleciona a descrição do spot de acordo com o id

```
select sp_bio from spots where sp_id = :id
```

Seleciona todos os favoritos spots de um user de acordo com seu id

```
select * from spots
inner join fav_spots on sp_id = fs_sp_id
where fs_us_id = :id
```

Mostra as tags de um spot de acordo com seu id

```
select sp_id, sp_name, tg_id, tg_name from spots
inner join users_spots_tags on sp_id = ust_sp_id
inner join tags on tg_id = ust_tg_id and ust_sp_id = :id
```

Mostra quais spots faltam passar para completar as routes em progresso de um user de acordo com seu id (exemplo user 2)

```
select rt_name, sp_name
from (select rs_id next_spot from progress_routes, route_spots
      where pr_rt_id = rs_rt_id and pr_sp_id = rs_sp_id and pr_us_id = :idUser)
next_start_spot, spots, routes, route_spots, progress_routes
where sp_id = rs_sp_id and rt_id = rs_rt_id and pr_rt_id = rt_id and rs_id >= next_spot
order by rs_id;
```

**Mostra o nome de uma route se todos os spots que a compõem forem gratis
(exemplo route 4)**

```
select rt_name
from (select rt_name, count(rt_name) total, count(sp_price = false) gratis
      from routes, route_spots, spots
      where sp_id = rs_sp_id and rt_id = rs_rt_id and rt_id = :idRoute
      group by rt_name) test
where total = gratis;
```