



Departamento de Ciência da Computação
Programa de Pós Graduação em Computação Aplicada

Fundamentos de Banco de Dados

Bruno Marquete da Silva
Rhuan Carlos Pereira de Queiroz



Introdução



Portal da Transparência

O trabalho foi desenvolvido a partir do acesso à dados abertos disponibilizados pelo Portal da Transparência do Governo Federal.

Viagens Realizadas a Serviço

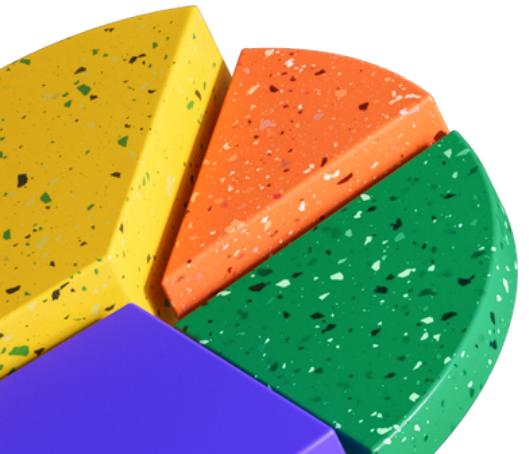
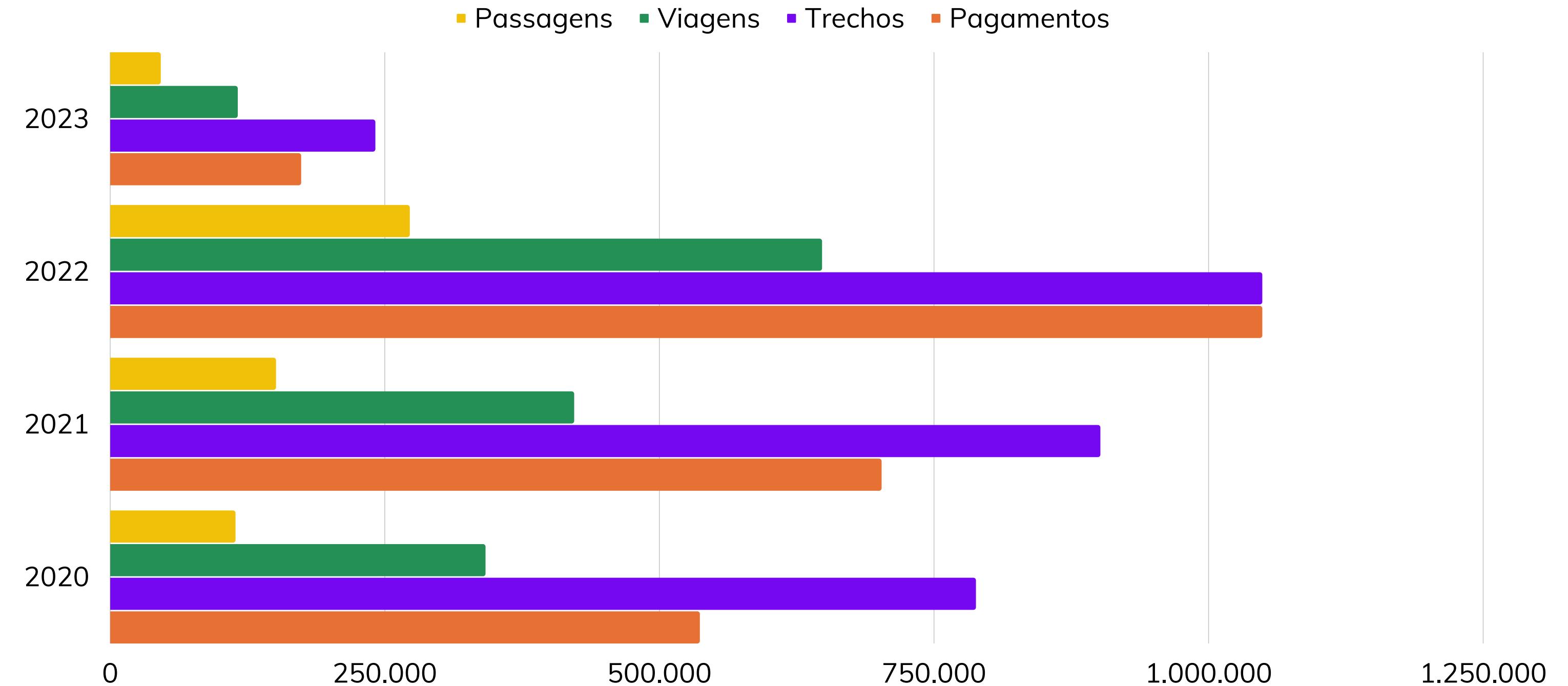
O domínio de dados utilizados foi o de viagens realizada a serviço que englobam passagens, trechos de viagens, viagens e pagamentos.

Dados fornecidos em CSV

Os dados abertos de viagens realizadas a serviço são fornecidos em formato CSV por meio de 4 (quatro) planilhas.

Dados de 2011 a 2023

Os dados fornecidos são do período de 2011 a 2023 na quais os arquivos CSV podem ser obtidos para cada ano selecionado.



Importamos dados de 2022



Passagens

272.672



Viagens

647.893



Trechos

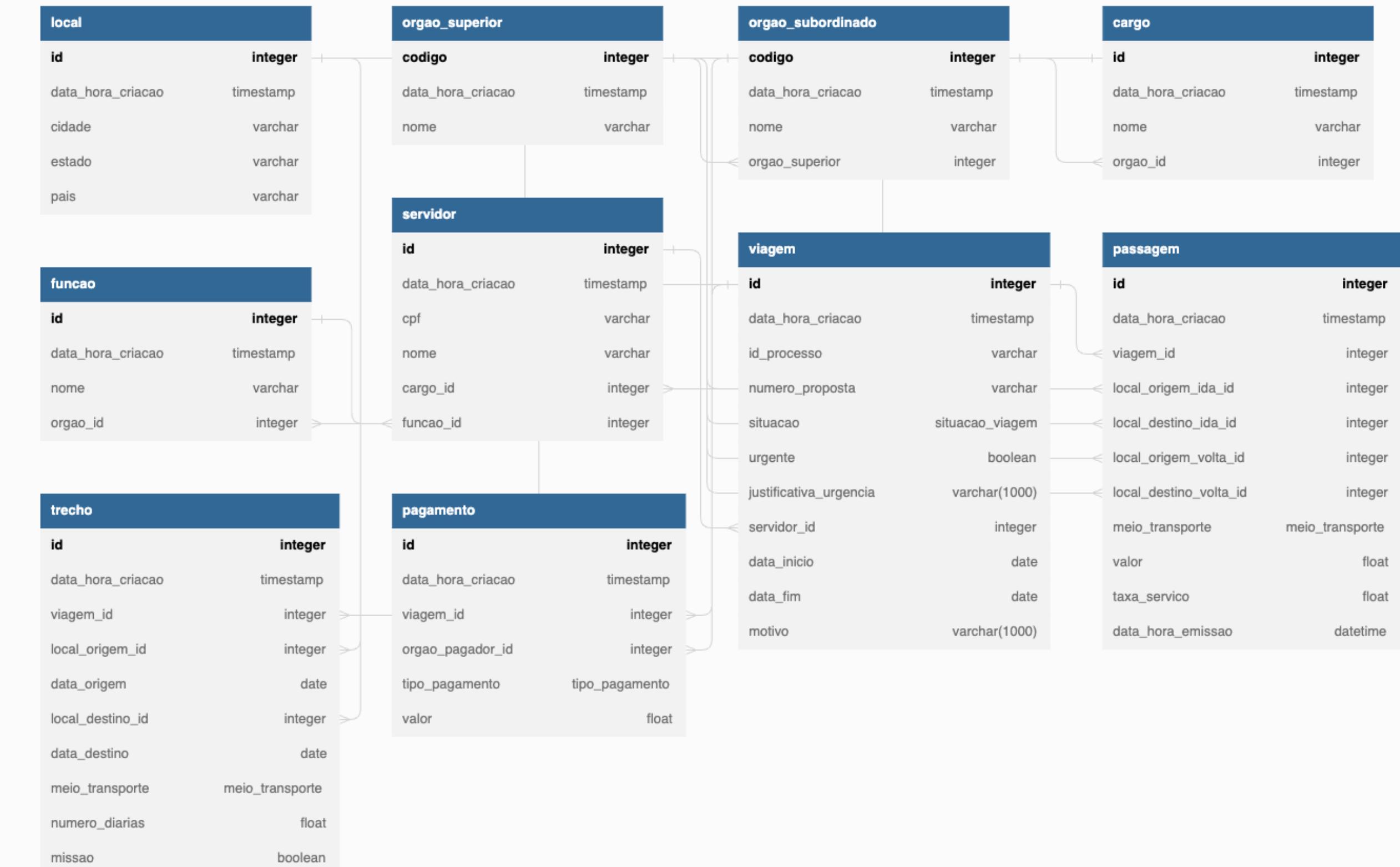
1.048.576



Pagamentos

1.048.576

Modelo de Dados Relacional



Script SQL Geração do Banco de Dados



```
CREATE DATABASE "viagens";
\c viagens

CREATE TYPE "situacao_viajem" AS ENUM (
    'REALIZADA',
    'NAO_REALIZADA'
);

CREATE TYPE "meio_transporte" AS ENUM (
    'AEREO',
    'RODOVIARIO',
    'FLUVIAL',
    'VEICULO_PROPRIOS',
    'VEICULO_OFICIAL',
    'MARITIMO',
    'FERROVIARIO',
    'INVALIDO'
);

CREATE TYPE "tipo_pagamento" AS ENUM (
    'PASSAGEM',
    'DIARIAS',
    'SERVICO_CORRELATO_SEGURO',
    'RESTITUICAO'
);

CREATE TABLE "local" (
    "id" SERIAL PRIMARY KEY,
    "data_hora_criacao" timestamp DEFAULT current_timestamp,
    "cidade" varchar NOT NULL,
    "estado" varchar,
    "pais" varchar NOT NULL
);

CREATE TABLE "orgao_superior" (
    "codigo" integer PRIMARY KEY,
    "data_hora_criacao" timestamp DEFAULT current_timestamp,
    "nome" varchar NOT NULL
);

CREATE TABLE "orgao_subordinado" (
    "codigo" integer PRIMARY KEY,
    "data_hora_criacao" timestamp DEFAULT current_timestamp,
    "nome" varchar NOT NULL,
    "orgao_superior" integer,
    CONSTRAINT orgao_orgao_sup_fk FOREIGN KEY ("orgao_superior") REFERENCES "orgao_superior" ("codigo")
);
```

Script SQL Geração do Banco de Dados



```
CREATE TABLE "cargo" (
    "id" SERIAL PRIMARY KEY,
    "data_hora_criacao" timestamp DEFAULT current_timestamp,
    "nome" varchar NOT NULL,
    "orgao_id" integer NOT NULL,
    FOREIGN KEY ("orgao_id") REFERENCES "orgao_subordinado" ("codigo")
);

CREATE TABLE "funcao" (
    "id" SERIAL PRIMARY KEY,
    "data_hora_criacao" timestamp DEFAULT current_timestamp,
    "nome" varchar NOT NULL,
    "orgao_id" integer NOT NULL,
    FOREIGN KEY ("orgao_id") REFERENCES "orgao_subordinado" ("codigo")
);

CREATE TABLE "servidor" (
    "id" SERIAL PRIMARY KEY,
    "data_hora_criacao" timestamp DEFAULT current_timestamp,
    "cpf" varchar,
    "nome" varchar NOT NULL,
    "cargo_id" integer,
    "funcao_id" integer,
    FOREIGN KEY ("cargo_id") REFERENCES "cargo" ("id"),
    FOREIGN KEY ("funcao_id") REFERENCES "funcao" ("id")
);

CREATE TABLE "viagem" (
    "id" SERIAL PRIMARY KEY,
    "data_hora_criacao" timestamp DEFAULT current_timestamp,
    "id_processo" varchar UNIQUE NOT NULL,
    "numero_proposta" varchar NOT NULL,
    "situacao" situacao_vagem NOT NULL,
    "urgente" boolean NOT NULL DEFAULT false,
    "justificativa_urgencia" varchar(1000),
    "servidor_id" integer NOT NULL,
    "data_inicio" date NOT NULL,
    "data_fim" date NOT NULL,
    "motivo" varchar(1000),
    FOREIGN KEY ("servidor_id") REFERENCES "servidor" ("id")
);
```

Script SQL Geração do Banco de Dados



```
CREATE TABLE "passagem" (
    "id" SERIAL PRIMARY KEY,
    "data_hora_criacao" timestamp DEFAULT current_timestamp,
    "viagem_id" integer NOT NULL,
    "local_origem_ida_id" integer,
    "local_destino_ida_id" integer,
    "local_origem_volta_id" integer,
    "local_destino_volta_id" integer,
    "meio_transporte" meio_transporte,
    "valor" float NOT NULL DEFAULT 0,
    "taxa_servico" float NOT NULL DEFAULT 0,
    "data_hora_emissao" timestamp,
    FOREIGN KEY ("viagem_id") REFERENCES "viagem" ("id") ON DELETE CASCADE,
    FOREIGN KEY ("local_origem_ida_id") REFERENCES "local" ("id"),
    FOREIGN KEY ("local_origem_volta_id") REFERENCES "local" ("id"),
    FOREIGN KEY ("local_destino_ida_id") REFERENCES "local" ("id"),
    FOREIGN KEY ("local_destino_volta_id") REFERENCES "local" ("id")
);

CREATE TABLE "trecho" (
    "id" SERIAL PRIMARY KEY,
    "data_hora_criacao" timestamp DEFAULT current_timestamp,
    "viagem_id" integer NOT NULL,
    "local_origem_id" integer NOT NULL,
    "data_origem" date NOT NULL,
    "local_destino_id" integer NOT NULL,
    "data_destino" date NOT NULL,
    "meio_transporte" meio_transporte,
    "numero_diarias" float,
    "missao" boolean NOT NULL DEFAULT false,
    FOREIGN KEY ("local_origem_id") REFERENCES "local" ("id"),
    FOREIGN KEY ("local_destino_id") REFERENCES "local" ("id"),
    FOREIGN KEY ("viagem_id") REFERENCES "viagem" ("id") ON DELETE CASCADE
);

CREATE TABLE "pagamento" (
    "id" SERIAL PRIMARY KEY,
    "data_hora_criacao" timestamp DEFAULT current_timestamp,
    "viagem_id" integer NOT NULL,
    "orgao_pagador_id" integer NOT NULL,
    "tipo_pagamento" tipo_pagamento,
    "valor" float NOT NULL DEFAULT 0,
    FOREIGN KEY ("viagem_id") REFERENCES "viagem" ("id") ON DELETE CASCADE,
    FOREIGN KEY ("orgao_pagador_id") REFERENCES "orgao_subordinado" ("codigo")
);
```

Processo de ETL



Linguagem Python

Linguagem de programação orientada a objetos de fácil compreensão e largamente utilizada em processamento e análise de dados.

Biblioteca Pandas

Permite uma abordagem rápida e flexível com estruturas robustas para realizar limpeza, tratamento e análise exploratória de dados.

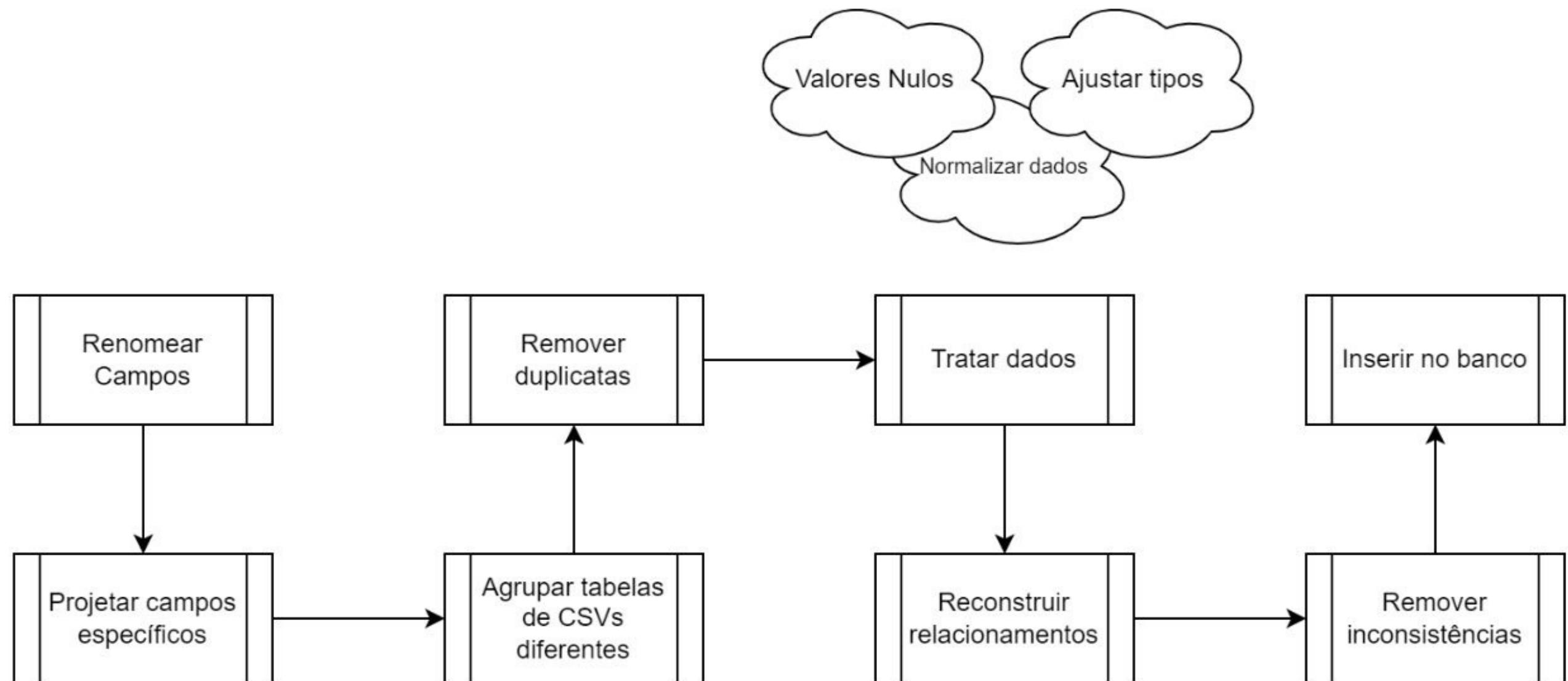
Biblioteca Numpy

Possibilita trabalhar com computação numérica, diversas tarefas matemáticas e processamento de arrays multimensionais.

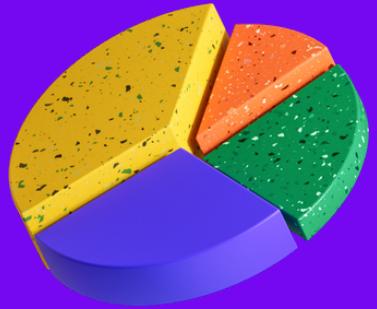
PostgreSQL

Sistema de gerenciamento de banco de dados relacionais de código aberto, com recursos robustos e ampla aceitação na indústria.

Processo de ETL



Consultas SQL



TOP 10 ÓRGÃOS QUE MAIS GASTARAM

```
-- TOP 10 ÓRGÃOS QUE MAIS GASTARAM
select upper(org.nome) as orgao,
       round(sum(pag.valor)::numeric, 2) as gasto_total
  from pagamento pag
inner join orgao_subordinado org on pag.orgao_pagador_id = org.codigo
inner join viagem viag on pag.viagem_id = viag.id
inner join passagem pas on pas.viagem_id = viag.id
 group by org.nome
having sum(pag.valor) > 0
order by gasto_total desc
limit 10;
```

Consultas SQL



TOP 10 ÓRGÃOS QUE MAIS GASTARAM

ABC	orgao	123	gasto_total
1	MINISTÉRIO DAS RELAÇÕES EXTERIORES - UNIDADES COM VÍNCULO DIRETO	87,014,775.97	
2	INSTITUTO NACIONAL DO SEGURO SOCIAL	70,025,765.92	
3	MINISTÉRIO DA AGRICULTURA, PECUÁRIA E ABASTECIMENTO - UNIDADES COM VÍNCULO DIRETO	46,090,618.85	
4	MINISTÉRIO DA ECONOMIA - UNIDADES COM VÍNCULO DIRETO	40,618,420.92	
5	INSTITUTO BRASILEIRO DO MEIO AMBIENTE E DOS RECURSOS NATURAIS RENOVÁVEIS	38,396,567.57	
6	MIN.DA GEST.E DA INOVACAO EM SERVICOS PUBLICO - UNIDADES COM VÍNCULO DIRETO	31,479,398.98	
7	AGÊNCIA NACIONAL DE VIGILÂNCIA SANITÁRIA	26,104,421.03	
8	AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL	25,152,171.13	
9	MINISTÉRIO DA SAÚDE - UNIDADES COM VÍNCULO DIRETO	23,657,116.7	
10	INSTITUTO NACIONAL DE COLONIZAÇÃO E REFORMA AGRÁRIA	16,894,461.49	

TOP 10 ÓRGÃOS COM MAIS PASSAGENS EMITIDAS EM URGÊNCIA

Consultas SQL



```
-- TOP 10 ÓRGÃOS COM MAIS PASSAGENS EMITIDAS EM URGÊNCIA
select upper(org.nome) as orgao,
       count(viag.urgente) as total_urgencias
  from pagamento pag
inner join orgao_subordinado org on pag.orgao_pagador_id = org.codigo
inner join viagem viag on pag.viagem_id = viag.id
 where viag.urgente = true
group by org.nome
having count(viag.urgente) > 0
order by total_urgencias desc
limit 10;
```

Consultas SQL



TOP 10 ÓRGÃOS COM MAIS PASSAGENS EMITIDAS EM URGÊNCIA

ABC orgao	total_urgencias
1 FUNDAÇÃO INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA	35,426
2 INSTITUTO NACIONAL DO SEGURO SOCIAL	32,001
3 MINISTÉRIO DA AGRICULTURA, PECUÁRIA E ABASTECIMENTO - UNIDADES COM VÍNCULO DIRETO	20,370
4 MINISTÉRIO DA ECONOMIA - UNIDADES COM VÍNCULO DIRETO	16,095
5 INSTITUTO BRASILEIRO DO MEIO AMBIENTE E DOS RECURSOS NATURAIS RENOVÁVEIS	15,655
6 INSTITUTO NACIONAL DE COLONIZAÇÃO E REFORMA AGRÁRIA	12,943
7 MINISTÉRIO DA SAÚDE - UNIDADES COM VÍNCULO DIRETO	8,440
8 FUNDAÇÃO NACIONAL DO ÍNDIO	7,660
9 FUNDAÇÃO NACIONAL DE SAÚDE	5,832
10 MIN.DA GEST.E DA INOVACAO EM SERVICOS PUBLICO - UNIDADES COM VÍNCULO DIRETO	5,153

TOP 10 SERVIDORES QUE MAIS REALIZARAM VIAGENS

Consultas SQL



```
-- TOP 10 SERVIDORES QUE MAIS REALIZARAM VIAGENS
select ser.nome,
       upper(org.nome) as orgao,
       count(viag.id) as qtd_vagens
  from viagem viag
inner join passagem pas on pas.viagem_id = viag.id
inner join servidor ser on viag.servidor_id = ser.id
inner join cargo car on ser.cargo_id = car.id
inner join orgao_subordinado org on car.orgao_id = org.codigo
 group by ser.nome, org.nome
having count(viag.id) > 0
order by qtd_vagens desc
limit 10;
```

Consultas SQL



TOP 10 SERVIDORES QUE MAIS REALIZARAM VIAGENS

	nome	orgao	qtd_viagens
1	RAPHAEL CAMARA MEDEIROS PARENTE	MINISTÉRIO DA SAÚDE - UNIDADES COM VÍNCULO DIRETO	124
2	TAMOIO ATHAYDE MARCONDES	FUNDAÇÃO NACIONAL DE ARTES	101
3	GILBERTO WALLER JUNIOR	CONTROLADORIA-GERAL DA UNIÃO - UNIDADES COM VÍNCULO DIRETO	93
4	ROBERTO DE OLIVEIRA CAMPOS NETO	BANCO CENTRAL DO BRASIL - ORÇAMENTO FISCAL E SEGURIDADE SOCIAL - UNIDADES COM VÍNCULO DIRETO	92
5	PAULO CESAR FAGUNDES NEVES	FUNDAÇÃO UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO	83
6	ELEUSA MARIA GUTEMBERG	INSTITUTO NACIONAL DE COLONIZAÇÃO E REFORMA AGRÁRIA	80
7	AUREO DO PRADO GONCALVES	MINISTÉRIO DA INFRAESTRUTURA - UNIDADES COM VÍNCULO DIRETO	72
8	MARCOS CESAR DE OLIVEIRA PINTO	SEM INFORMAÇÃO	69
9	GIUSEPPE SERRA SECA VIEIRA	INSTITUTO NACIONAL DE COLONIZAÇÃO E REFORMA AGRÁRIA	68
10	JOSE TAROUCO CORREA JUNIOR	MINISTÉRIO DA INFRAESTRUTURA - UNIDADES COM VÍNCULO DIRETO	66

Consultas SQL



TOP 10 DESTINOS DE VIAGENS

```
● ● ●  
-- TOP 10 DESTINOS DE VIAGENS  
select destinos.destino,  
       count(destinos.destino) as quantidade  
from  
(  
    select concat(destino_ida.cidade,  
                  ' - ', destino_ida.estado,  
                  ' - ', destino_ida.pais) as destino  
    from viagem viag  
    inner join passagem pas  
      on pas.viagem_id = viag.id  
    inner join "local" destino_ida  
      on pas.local_destino_ida_id =  
          destino_ida.id  
    where destino_ida.cidade <> 'Sem Informação'  
  
    union all  
  
    select concat(destino_volta.cidade,  
                  ' - ', destino_volta.estado,  
                  ' - ', destino_volta.pais) as destino  
    from viagem viag  
    inner join passagem pas  
      on pas.viagem_id = viag.id  
    inner join "local" destino_volta  
      on pas.local_destino_volta_id =  
          destino_volta.id  
    where destino_volta.cidade <> 'Sem Informação'  
  
) as destinos  
group by destinos.destino  
having count(destinos.destino) > 0  
order by quantidade desc  
limit 10;
```

Consultas SQL



TOP 10 DESTINOS DE VIAGENS

ABC	destino	123	quantidade
1	Brasília - Distrito Federal - Brasil	38,772	
2	Rio de Janeiro - Rio de Janeiro - Brasil	12,907	
3	São Paulo - São Paulo - Brasil	9,770	
4	Belo Horizonte - Minas Gerais - Brasil	4,818	
5	Belém - Pará - Brasil	4,532	
6	Porto Alegre - Rio Grande do Sul - Brasil	4,522	
7	Manaus - Amazonas - Brasil	4,451	
8	Recife - Pernambuco - Brasil	4,328	
9	Salvador - Bahia - Brasil	4,205	
10	Curitiba - Paraná - Brasil	4,094	

VALOR MÉDIO DAS PASSAGENS EMITIDAS COM DESTINO SÃO PAULO

Consultas SQL

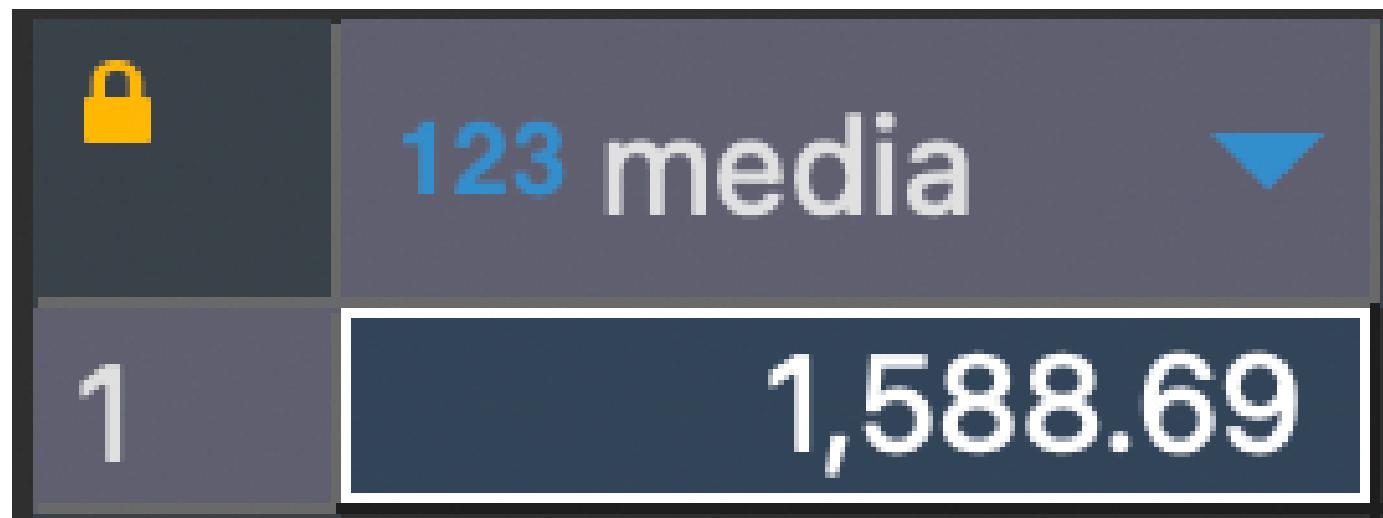


```
-- VALOR MÉDIO DAS PASSAGENS EMITIDAS COM DESTINO SÃO PAULO
select round(avg(pass.valor)::numeric, 2) as media
from viagem viag
    inner join passagem pass
        on pass.viagem_id = viag.id
    left join "local" origem_ida
        on pass.local_origem_ida_id = origem_ida.id
    left join "local" destino_ida
        on pass.local_origem_ida_id = destino_ida.id
    left join "local" origem_volta
        on pass.local_origem_ida_id = origem_volta.id
    left join "local" destino_volta
        on pass.local_origem_ida_id = destino_volta.id
where
    destino_ida.id in (select id
                        from "local" l
                        where l.cidade = 'São Paulo'
                            and l.estado = 'São Paulo'
                            and l.pais = 'Brasil')
        or
    destino_volta.id in (select id
                        from "local" l
                        where l.cidade = 'São Paulo'
                            and l.estado = 'São Paulo'
                            and l.pais = 'Brasil');
```

Consultas SQL



VALOR MÉDIO DAS PASSAGENS EMITIDAS COM DESTINO SÃO PAULO



Procedure



```
● ● ●  
-- CRIA PROCEDURE QUE CALCULA E GRAVA A TAXA DE SERVIÇO DE UMA PASSAGEM COM BASE EM UM  
-- PERCENTUAL ESPECÍFICO DO VALOR DA PASSAGEM  
create or replace procedure calcular_gravar_taxa_servico(id_passagem int, percentual int)  
language plpgsql as $$  
begin  
    update passagem  
    set taxa_servico = valor * (percentual / 100)  
    where id = id_passagem  
        and valor is not null;  
    commit;  
end;  
$$;  
  
-- INVOCA A PROCEDURE CRIADA PARA A PASSAGEM DE ID 60 E APLICANDO 20%  
call calcular_gravar_taxa_servico(60, 20);
```

Function



```
-- CRIA FUNCTION QUE RETORNA OS PAGAMENTOS REALIZADOS PARA UMA VIAGEM
create or replace function pagamentos_viagem(id_viagem int)
returns table(tipo_pagamento int, valor float)
language plpgsql as $$
begin
    return query select pag.tipo_pagamento, pag.valor
                      from pagamento as pag
                     where pag.viagem_id = id_viagem;
end;
$$;

-- INVOCAR A FUNCTION CRIADA PARA A VIAGEM DE ID 50
call pagamentos_viagem(50);
```

Function



```
-- CRIA FUNCTION QUE RETORNA TRIGGER QUE DESCONSIDERA VALORES NEGATIVOS
-- NO VALOR E TAXA DE SERVIÇO DE PASSAGENS
create or replace function desconsiderar_valores_negativos_passagem()
returns trigger
language plpgsql as $$
begin
    update passagem
    set valor = 0
    where valor is not null
        and valor < 0;
    update passagem
    set taxa_servico = 0
    where taxa_servico is not null
        and taxa_servico < 0;
    commit;
end;
$$;
```

Function



```
-- CRIA FUNCTION QUE RETORNA TRIGGER QUE DESCONSIDERA VALORES NEGATIVOS
-- EM PAGAMENTOS
create or replace function desconsiderar_valores_negativos_pagamento()
returns trigger
language plpgsql as $$
begin
    update pagamento
    set valor = 0
    where valor is not null
        and valor < 0;
    commit;
end;
$$;
```

Function



```
-- CRIA FUNCTION QUE RETORNA TRIGGER QUE REMOVE VALOR "SEM INFORMAÇÃO" DE LOCAL
create or replace function remover_sem_informacao_local()
returns trigger
language plpgsql as $$
begin
    update "local"
    set cidade = null
    where cidade = 'Sem Informação';
    update "local"
    set estado = null
    where estado = 'Sem Informação';
    update "local"
    set pais = null
    where pais = 'Sem Informação';
    commit;
end;
$$;
```

Trigger



```
-- CRIA TRIGGER QUE INVOCARÁ FUNÇÃO PARA DESCONSIDERAR VALORES NEGATIVOS  
-- DE VALOR E TAXA DE SERVIÇO SEMPRE QUE FOR INSERIDA OU ATUALIZADA UMA PASSAGEM  
create trigger triguer_desconsiderar_valores_negativos_passagem  
after insert or update on passagem  
for each row  
execute function desconsiderar_valores_negativos_passagem();  
  
-- CRIA TRIGGER QUE INVOCARÁ FUNÇÃO PARA DESCONSIDERAR VALORES NEGATIVOS  
-- DE VALOR DE PAGAMENTO SEMPRE QUE FOR INSERIDO OU ATUALIZADO UM PAGAMENTO  
create trigger triguer_desconsiderar_valores_negativos_pagamento  
after insert or update on pagamento  
for each row  
execute function desconsiderar_valores_negativos_pagamento();  
  
-- CRIA TRIGGER QUE INVOCARÁ FUNÇÃO PARA REMOVER VALOR "SEM INFORMAÇÃO" DE LOCAL  
create trigger triguer_remover_sem_informacao_local  
after insert or update on "local"  
for each row  
execute function remover_sem_informacao_local();
```

View



```
-- CRIA VIEW QUE EXIBE DADOS DETALHADOS DE PASSAGENS DE VIAGENS EMITIDAS PARA
-- SERVIDORES PÚBLICOS EM VIAGEM
create or replace view viagens_servidores as
select v.id_processo,
       v.numero_proposta,
       case
           when v.situacao = 'REALIZADA' then 'REALIZADA'
           when v.situacao = 'NAO_REALIZADA' then 'NÃO REALIZADA'
           else 'NÃO REALIZADA'
       end as situacao,
       case
           when v.urgente = true then 'SIM'
           else 'NÃO'
       end as urgente,
       to_char(v.data_inicio, 'DD/MM/YYYY') as data_inicio,
       to_char(v.data_fim, 'DD/MM/YYYY') as data_fim,
       upper(s.nome) as nome_servidor,
       upper(c.nome) as nome_cargo,
       upper(o.nome) as nome_orgao,
       to_char(p.data_hora_emissao, 'DD/MM/YYYY HH12:MI:SS') as data_hora_emissao_passagem,
       concat(origem_ida.cidade, ' - ',origem_ida.estado, ' - ', origem_ida.pais) as origem_ida,
       concat(destino_ida.cidade, ' - ',destino_ida.estado, ' - ', destino_ida.pais) as destino_ida,
       concat(origem_volta.cidade, ' - ',origem_volta.estado, ' - ', origem_volta.pais) as
origem_volta,
       concat(destino_volta.cidade, ' - ',destino_volta.estado, ' - ', destino_volta.pais) as
destino_volta,
       p.valor,
       p.taxa_servico,
       p.valor + p.taxa_servico as soma_total_valores
from viagem v
     inner join servidor s on v.servidor_id = s.id
     left join cargo c on v.servidor_id = c.id
     left join orgao_subordinado o on c.orgao_id = o.codigo
     inner join passagem p on p.viagem_id = v.id
     inner join "local" origem_ida on p.local_origem_ida_id = origem_ida.id
     inner join "local" destino_ida on p.local_origem_ida_id = destino_ida.id
     inner join "local" origem_volta on p.local_origem_ida_id = origem_volta.id
     inner join "local" destino_volta on p.local_origem_ida_id = destino_volta.id;
```

View



ABC id_processo	ABC numero_proposta	ABC situacao	ABC urgente	ABC data_inicio	ABC data_fim	ABC nome_servidor	ABC nome_cargo	ABC nome_orgao	ABC data_hora_e
562	17960976	005521/22-3C	REALIZADA	SIM	24/04/2022	05/08/2022	LUIZ ANTONIO SANTOS BEZERRA	PESQUISADOR	MINISTÉRIO DA AGRIC 30/06/2022 10:
563	17960976	005521/22-3C	REALIZADA	SIM	24/04/2022	05/08/2022	LUIZ ANTONIO SANTOS BEZERRA	PESQUISADOR	MINISTÉRIO DA AGRIC 19/04/2022 06:
564	17961021	001197/22	REALIZADA	NÃO	28/04/2022	30/04/2022	PAULO DE BARROS LYRA FILHO	[NULL]	[NULL] 11/04/2022 04:
565	17961021	001197/22	REALIZADA	NÃO	28/04/2022	30/04/2022	PAULO DE BARROS LYRA FILHO	[NULL]	[NULL] 11/04/2022 04:
566	17961106	006318/22-1C	REALIZADA	SIM	02/05/2022	17/05/2022	ALEX DA SILVA SANTOS	[NULL]	[NULL] 09/05/2022 11:
567	17961106	006318/22-1C	REALIZADA	SIM	02/05/2022	17/05/2022	ALEX DA SILVA SANTOS	[NULL]	[NULL] 12/04/2022 01:
568	17961137	000313/22	REALIZADA	SIM	19/04/2022	20/04/2022	RICARDO MARCELO FONSECA	[NULL]	[NULL] 12/04/2022 03:
569	17961137	000313/22	REALIZADA	SIM	19/04/2022	20/04/2022	RICARDO MARCELO FONSECA	[NULL]	[NULL] 12/04/2022 03:
570	17961155	000315/22	REALIZADA	NÃO	27/06/2022	01/07/2022	MARINES DE PAULI THOMAZ	[NULL]	[NULL] 26/05/2022 06:
571	17961155	000315/22	REALIZADA	NÃO	27/06/2022	01/07/2022	MARINES DE PAULI THOMAZ	[NULL]	[NULL] 26/05/2022 06:
572	17961171	005795/22	REALIZADA	NÃO	24/04/2022	10/05/2022	MARIA DAS GRACAS FIGUEIREDO PESS	[NULL]	[NULL] 12/04/2022 10:
573	17961171	005795/22	REALIZADA	NÃO	24/04/2022	10/05/2022	MARIA DAS GRACAS FIGUEIREDO PESS	[NULL]	[NULL] 12/04/2022 10:
574	17961175	000298/22-1C	REALIZADA	SIM	01/05/2022	01/06/2022	IRACEMA FERREIRA LIMA	[NULL]	[NULL] 13/05/2022 06:
575	17961175	000298/22-1C	REALIZADA	SIM	01/05/2022	01/06/2022	IRACEMA FERREIRA LIMA	[NULL]	[NULL] 11/04/2022 05:
576	17961200	006480/22	REALIZADA	SIM	17/04/2022	23/04/2022	CARLOS HENRIQUE ANGRISANI SANTA	[NULL]	[NULL] 02/05/2022 12:
577	17961200	006480/22	REALIZADA	SIM	17/04/2022	23/04/2022	CARLOS HENRIQUE ANGRISANI SANTA	[NULL]	[NULL] 02/05/2022 12:
578	17961247	000266/22	REALIZADA	SIM	10/04/2022	14/04/2022	CLAUDIA LINHARES SALES	[NULL]	[NULL] 11/04/2022 09:
579	17961247	000266/22	REALIZADA	SIM	10/04/2022	14/04/2022	CLAUDIA LINHARES SALES	[NULL]	[NULL] 11/04/2022 09:
580	17961293	000685/22	REALIZADA	SIM	14/04/2022	14/04/2022	STEPHANIE SCHNOLL	[NULL]	[NULL] 08/04/2022 06:
581	17961337	000018/22	REALIZADA	NÃO	04/07/2022	09/07/2022	GILSON ANTONIO CARNEIRO	[NULL]	[NULL] 12/04/2022 08:
582	17961337	000018/22	REALIZADA	NÃO	04/07/2022	09/07/2022	GILSON ANTONIO CARNEIRO	[NULL]	[NULL] 12/04/2022 08:
583	17961387	000466/22	REALIZADA	SIM	13/04/2022	14/04/2022	VINICIUS DANTAS DAMASCENO DE AR	CONTADOR	UNIVERSIDADE TECNI 11/04/2022 09:
584	17961387	000466/22	REALIZADA	SIM	13/04/2022	14/04/2022	VINICIUS DANTAS DAMASCENO DE AR	CONTADOR	UNIVERSIDADE TECNI 11/04/2022 09:
585	17961398	005807/22-3C	REALIZADA	SIM	11/04/2022	08/07/2022	VERA LUCIA DUARTE DE JESUS	[NULL]	[NULL] 24/06/2022 05:

ABC origem_ida	ABC destino_ida	ABC origem_volta	ABC destino_volta	123 valor	123 taxa_servico	123 soma_total_valores
Belém - Pará - Brasil	1,330.83	0	1,330.83			
Recife - Pernambuco - Brasil	1,288.69	0	1,288.69			
Alta Floresta - Mato Grosso - Brasil	1,941.96	0	1,941.96			
Goiânia - Goiás - Brasil	1,446.25	0	1,446.25			
Brasília - Distrito Federal - Brasil	613.42	0	613.42			
Maceió - Alagoas - Brasil	2,989.28	0	2,989.28			
Brasília - Distrito Federal - Brasil	940.21	0	940.21			
João Pessoa - Paraíba - Brasil	946.33	0	946.33			
Goiânia - Goiás - Brasil	1,137.28	0	1,137.28			
Rio de Janeiro - Rio de Janeiro - Brasil	Rio de Janeiro - Rio de Janeiro - Brasil	Rio de Janeiro - Rio de Janeiro - Brasil	Rio de Janeiro - Rio de Janeiro - Brasil	871.01	0	871.01
Curitiba - Paraná - Brasil	670.93	0	670.93			
Maringá - Paraná - Brasil	3,074.27	0	3,074.27			
Curitiba - Paraná - Brasil	791.38	75.19	866.57			
Cascavel - Paraná - Brasil	650.36	62.68	713.04			
Foz do Iguaçu - Paraná - Brasil	1,521.76	0	1,521.76			
João Pessoa - Paraíba - Brasil	2,190.35	0.49	2,190.84			
Guarulhos - São Paulo - Brasil	935.43	0	935.43			
Porto Velho - Rondônia - Brasil	1,266.27	0	1,266.27			
Washington - Estados Unidos da Am	4,072.51	0	4,072.51			
Guarulhos - São Paulo - Brasil	4,072.56	0	4,072.56			
Fortaleza - Ceará - Brasil	1,134.24	0	1,134.24			
Recife - Pernambuco - Brasil	2,791.69	0	2,791.69			
Brasília - Distrito Federal - Brasil	2,789.2	0	2,789.2			
Uberlândia - Minas Gerais - Brasil	1,157.11	0	1,157.11			



Departamento de Ciência da Computação
Programa de Pós Graduação em Computação Aplicada

Muito obrigado!

