

# Viagens realizadas a serviço - Ano 2020

Bonifácio, Leonardo de Holanda

200095056@aluno.unb.br

Moura, Dione Ramos Canuto

200094980@aluno.unb.br

20 de março de 2021

## 1 Requisitos

Deseja-se criar um banco de dados contendo todas as ocorrências de viagens a serviço pelo Governo Federal. Para isso, informações como identificador da viagem, lugar de origem e destino, ministério responsável, passageiro e resumo dos gastos são levados em conta.

Toda viagem deve conter:

- Identificador do processo da viagem;
- Se ela foi realizada ou não;
- Cunho de urgência;
- Data de ida e volta;
- Ministério responsável;
- Passageiro com seu respectivo cargo e função;
- País, Unidade Federativa e cidade dos trechos;
- Qual o meio de transporte da viagem;
- Gasto com as passagens e taxas.

## 2 Modelo lógico

O modelo lógico, feito no MySQL Workbench, pode ser visto no Anexo [A](#). A lógica utilizada foi de três centralizadores (Tabela VIAGEM, PASSAGEM e PAGAMENTO), que tem todas as informações necessárias para realizar as consultas desejadas. As tabelas secundárias são compostas de variáveis categóricas com suas respectivas *Primary Keys*. Essas foram transformadas em *Foreign Keys* nas tabelas principais no processo de ETL.

### 3 PrintScreen das Tabelas

O *dataset* foi baixado no [Portal de transparência](#) e, antes dos tratamentos, era composto de cerca de 300 mil registros no total. Os arquivos do projeto, como modelagem, *script* de criação e inserção do banco podem ser encontrados no [Github](#).

Todas as tabelas podem ser vistas no Anexo B. Algumas imagens foram cortadas para melhorar a visualização. Os dados foram tratados utilizando Python e a biblioteca Pandas e Numpy.

Dentre os processos de ETL pode-se citar:

- Transformação de *string*: *lower case*, retirada de acentos caracteres especiais, remoção de espaços excedentes;
- Remoção ou substituição de dados faltantes;
- Tratamento de dados confusos ou errados;
- Criação dos *dataframes* para inserção;
- Inserção no banco de dados com a biblioteca Python *mysql*.

### 4 SQL

- Trigger [C.1](#): ao inserir ou atualizar um registro na tabela VIAGEM (Fig. 2), é inserido um registro na tabela LOG com os dados antigo e/ou novos.
- View [C.2](#): mostra os meios de transporte mais utilizados (Fig. 17) nas viagens.
- Procedure [C.3](#): mostra os 5 países mais visitados pelos funcionários dos Órgãos (Fig. 18), identificados pelo nome.
- Consulta Total gasto em viagens [C.4](#): gasto total em viagens no período.
- Consulta Total gasto em viagens dividido por cunho de emergência [C.5](#): gasto total em viagens no período.
- Consulta Média gasto por Ministério [C.6](#): gasto média por passagem para cada Ministério.
- Consulta Total de viagens por cidade [C.7](#): número total de viagens dividido por cidade.
- Consulta Total de viagens por ministério [C.8](#): número total de viagens dividido por ministério.

## A Modelo Lógico

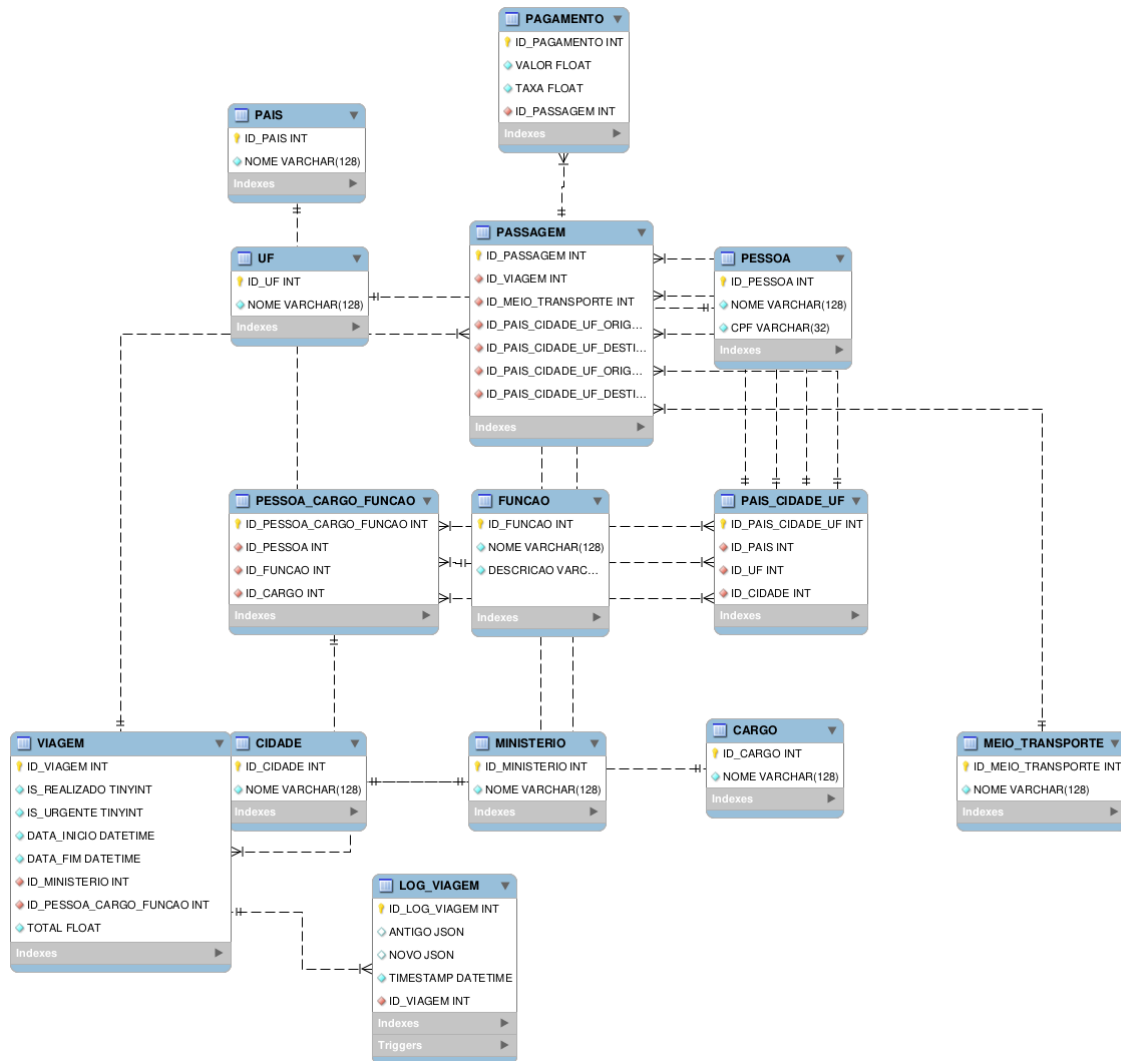


Figura 1: Modelo lógico no MySQL Workbench. Fonte: Autor.

## B PrintScreen das Tabelas

ID_VIAGEM	IS_REALIZADO	IS_URGENTE	DATA_INICIO	DATA_FIM	ID_MINISTERIO	ID_PESSOA_CARGO_FUNCAO	TOTAL
16332319	0	0	2020-01-05...	2021-01...	26000	1	0
16386658	1	0	2020-03-10...	2020-03...	26000	2	1053.89
16401038	1	0	2020-01-01...	2020-09...	26000	3	0
16413295	1	0	2020-03-02...	2020-03...	26000	4	628.56
16415470	1	1	2020-01-11...	2020-01...	26000	5	6903.68
16441929	1	0	2020-01-07...	2020-01...	52000	6	0
16443199	1	0	2020-01-07...	2020-01...	52000	7	864.75
16450992	1	0	2020-02-01...	2020-03...	36000	8	0
16451110	0	0	2020-03-31...	2020-04...	26000	9	4500.2
16460934	1	0	2020-02-01...	2020-02...	26000	10	0
16465411	0	0	2020-02-08...	2020-02...	52000	11	221.85

Figura 2: Tabela VIAGEM. Fonte: Autor.

id_passagem	id_viagem	id_meio	id_puc_origem_ida	id_puc_destino_ida	id_puc_origem_volta	id_puc_destino_volta
1	16386658	1	1	2	3	3
2	16386658	1	2	1	3	3
3	16413295	1	4	2	2	4
4	16415470	1	5	6	3	3
5	16415470	1	6	5	3	3
6	16443199	1	5	7	3	3
7	16451110	1	5	8	3	3
8	16451110	1	8	5	3	3
9	16465411	1	5	2	3	3
10	16466765	1	5	4	3	3
11	16466765	1	4	2	3	3
12	16466765	1	2	5	3	3
13	16466876	1	9	2	3	3
14	16466876	1	10	11	3	3
15	16466876	1	11	9	3	3
16	16466876	1	2	10	3	3
17	16466876	1	11	9	3	3

Figura 3: Tabela PASSAGEM. Fonte: Autor.

id_pagamento	valor	taxa	id_passagem
1	522.04	1.14	1
2	529.57	1.14	2
3	627.42	1.14	3
4	3451.84	0.0	4
5	3451.84	0.0	4
6	3451.84	0.0	5
7	3451.84	0.0	5
8	864.75	0.0	6
9	2250.1	0.0	7
10	2250.1	0.0	7
11	2250.1	0.0	8
12	2250.1	0.0	8
13	221.85	0.0	9
14	448.75	0.0	10
15	278.59	0.0	11
16	237.57	0.0	12
17	791.95	0.0	13
18	900.85	0.0	14
19	589.85	0.0	15
20	865.57	1123.8	16

Figura 4: Tabela PAGAMENTO. Fonte: Autor.

id_paises	nome
1	bosnia-herzegovina
2	arabia saudita
3	belarus
4	zimbabue
5	chile
6	nova zelandia
7	suica
8	alemanha
9	suriname
10	indonesia
11	dinamarca
12	benin
13	kuaite
14	africa do sul
15	cabo verde
16	noruega
17	invalido
18	equador

Figura 5: Tabela PAÍS. Fonte: Autor.

id_uf	nome
1	pernambuco
2	arabia saudita
3	belarus
4	chile
5	nova zelandia
6	suica
7	paraiba
8	indonesia
9	benin
10	africa do sul
11	invalido
12	estados unidos da america
13	rio grande do sul
14	australia
15	ira

Figura 6: Tabela UF. Para fora do Brasil, copia-se o nome do país. Fonte: Autor.

id_cidade	nome
1	genebra
2	arabia saudita
3	lusaca
4	lanzhou
5	juina
6	colorado springs - colorado
7	harrisburg - pensilvania
8	chile
9	nova zelandia
10	suica
11	sarasota - florida
12	jurua
13	oriximina - porto trombetas
14	beruri
15	anama
16	columbia - carolina do sul
17	kuaite
18	itamarati
19	cabo frio
20	filadelfia - pensilvania

Figura 7: Tabela CIDADE. Para fora do Brasil, às vezes copia-se o nome do país. Fonte: Autor.

id_puc	id_pais	id_uf	id_cidade
1	63	54	462
2	63	76	256
3	118	83	355
4	63	16	29
5	63	40	466
6	57	83	490
7	63	13	299
8	107	83	218
9	63	1	365
10	63	82	450
11	63	79	87
12	63	26	542
13	63	73	239
14	63	7	519
15	63	7	121
16	63	26	593

Figura 8: Tabela PAIS\_UF\_CIDADE. Fonte: Autor.

id_pessoa	nome	cpf
1	claudia correa de andrade oliveira	***.211.458-**
2	alexsandro galeno araujo dantas	***.708.864-**
3	fernando carlos giacomelli	***.634.280-**
4	rodrigo ramos hospodar felippe valverde	***.464.507-**
5	martius vicente rodriguez y rodriguez	***.126.127-**
6	willen bandeira garido	***.207.387-**
7	rafael henrique rodrigues de souza	***.205.262-**
8	marcia cavalcanti raposo lopes	***.566.657-**
9	fatima siqueira caropreso	***.830.538-**
10	ramiro antonio da costa	***.752.909-**
11	wilton da cruz barros	***.940.007-**
12	frederico alberto marcondes felipe	***.414.718-**
13	john william macquarie	***.938.221-**
14	francismar ferreira lima	***.229.766-**
15	jefferson abranes dos santos	***.275.212-**

Figura 9: Tabela PESSOA. Fonte: Autor.

id_cargo	nome
1	professor do magisterio superior
2	sem informacao
3	pesquisador em saude publica
4	professor ens basico tecn tecnologico
5	professor tit-livre magist superior
6	professor magisterio superior-visitante
7	assistente de chancelaria
8	secretario executivo
9	informacoes protegidas por sigilo
10	analista administrativo
11	oficial de chancelaria
12	jornalista
13	terceiro secretario
14	primeiro secretario
15	tecnologista
16	bibliotecario-documentalista
17	tecnico em restauracao
18	analista de tecnologia da informacao
19	revisor de textos

Figura 10: Tabela CARGO. Fonte: Autor.

id_funcao	nome	descricao
1	sem informacao	sem informacao
2	fuc-0001	fuc-0001
3	cd-0003	cargo comissionado de direcao
4	ofsuperior	ofsuperior
5	of.general	oficiais gerais
6	fg-0001	fg-0001
7	sigilosa	informacoes protegidas por sigilo
8	cct-iv	cargo comissionado tecnico
9	fpe-1024	fpe-1024
10	cd-0004	cargo comissionado de direcao
11	cd-0002	cargo comissionado de direcao
12	fgr-1	fgr-1
13	fg-0003	fg-0003
14	fg-0002	fg-0002
15	fpe-1011	fpe-1011
16	das-1015	cargo de direcao
17	fpe-1012	fpe-1012
18	cct-iii	cargo comissionado tecnico
19	ofsubalter	ofsubalter
20	subtenente	subtenentes
21	fpe-1014	fpe-1014
22	ofintermed	ofintermed

Figura 11: Tabela FUNCAO. Fonte: Autor.

id_cpf	id_pessoa	id_cargo	id_funcao
1	1	1	1
2	2	1	2
3	3	1	1
4	4	2	1
5	5	1	3
6	6	2	4
7	7	2	4
8	8	3	1
9	9	1	1
10	10	4	1
11	11	2	4
12	12	2	5
13	13	1	1
14	14	1	1
15	15	1	1
16	16	5	1

Figura 12: Tabela PESSOA\_CARGO\_FUNCAO. Fonte: Autor.

id_meio	nome
1	aereo
2	rodoviario
3	fluvial
4	maritimo

Figura 13: Tabela MEIO\_TRANSPORTE. Fonte: Autor.

id_ministerio	nome
26000	ministerio da educacao
52000	ministerio da defesa
36000	ministerio da saude
35000	ministerio das relacoes exteriores
30000	ministerio da justica e seguranca publica
25000	ministerio da economia
37000	controladoria-geral da uniao
39000	ministerio da infraestrutura
32000	ministerio de minas e energia
-1	sem informacao
44000	ministerio do meio ambiente
63000	advocacia-geral da uniao
22000	ministerio da agricultura, pecuaria e abastecimento
54000	ministerio do turismo
20000	presidencia da república
24000	ministerio da ciencia, tecnologia, inovacoes e comunicacoes
20113	ministerio do planejamento, desenvolvimento e gestao
53000	ministerio do desenvolvimento regional
41000	ministerio das comunicacoes

Figura 14: Tabela MINISTERIO. Fonte: Autor.

ID_LOG_VIAGEM	ANTIGO	NOVO	TIMESTAMP	ID_VIAGEM
1	NULL	{"cpf": 1, "data_FIM": "2021-01-04 ...	2021-03-14 00:00:00	16332319
2	NULL	{"cpf": 2, "data_FIM": "2020-03-12 ...	2021-03-14 00:00:00	16386658
3	NULL	{"cpf": 3, "data_FIM": "2020-09-15 ...	2021-03-14 00:00:00	16401038
4	NULL	{"cpf": 4, "data_FIM": "2020-03-02 ...	2021-03-14 00:00:00	16413295
5	NULL	{"cpf": 5, "data_FIM": "2020-01-23 ...	2021-03-14 00:00:00	16415470
6	NULL	{"cpf": 6, "data_FIM": "2020-01-10 ...	2021-03-14 00:00:00	16441929
7	NULL	{"cpf": 7, "data_FIM": "2020-01-10 ...	2021-03-14 00:00:00	16443199
8	NULL	{"cpf": 8, "data_FIM": "2020-03-21 ...	2021-03-14 00:00:00	16450992
9	NULL	{"cpf": 9, "data_FIM": "2020-04-06 ...	2021-03-14 00:00:00	16451110
10	NULL	{"cpf": 10, "data_FIM": "2020-02-14...	2021-03-14 00:00:00	16460934
11	NULL	{"cpf": 11, "data_FIM": "2020-02-08...	2021-03-14 00:00:00	16465411

Figura 15: Tabela LOG. Fonte: Autor.



## C Consultas SQL

### C.1 Trigger LOG

#### C.1.1 Consulta

```
CREATE DEFINER = CURRENT_USER TRIGGER
'VIAGENS_DB'.'LOG_VIAGEM_AFTER_INSERT'
AFTER INSERT ON 'VIAGEM'
FOR EACH ROW
BEGIN
    INSERT INTO VIAGENS_DB.LOG_VIAGEM (
        ID_VIAGEM,
        ANTIGO,
        NOVO,
        TIMESTAMP
    )
    VALUES(
        NEW.ID_VIAGEM,
        null,
        JSON_OBJECT(
            "data_inicio", NEW.DATA_INICIO,
            "data_FIM", NEW.DATA_FIM,
            "ministerio", NEW.ID_MINISTERIO,
            "cpf", NEW.ID_PESSOA_CARGO_FUNCAO
        ),
        CURRENT_DATE
    );
END
```

```
CREATE DEFINER = CURRENT_USER TRIGGER
'VIAGENS_DB'.'LOG_VIAGEM_AFTER_UPDATE'
AFTER UPDATE ON 'VIAGEM'
FOR EACH ROW
BEGIN
    INSERT INTO VIAGENS_DB.LOG_VIAGEM (
        ID_VIAGEM,
        ANTIGO,
        NOVO,
        TIMESTAMP
    )
```

```

VALUES(
    NEW.ID_VIAGEM,
    JSON_OBJECT(
        "data_inicio", OLD.DATA_INICIO,
        "data_FIM", OLD.DATA_FIM,
        "ministerio", OLD.ID_MINISTERIO,
        "cpf", OLD.ID_PESSOA_CARGO_FUNCAO
    ),
    JSON_OBJECT(
        "data_inicio", NEW.DATA_INICIO,
        "data_FIM", NEW.DATA_FIM,
        "ministerio", NEW.ID_MINISTERIO,
        "cpf", NEW.ID_PESSOA_CARGO_FUNCAO
    ),
    CURRENT_DATE
);
END

```

### C.1.2 Resultado

ID_LOG_VIAGEM	ANTIGO	NOVO	TIMESTAMP	ID_VIAGEM
1	NULL	{"cpf": 1, "data_FIM": "2021-01-04 ...	2021-03-14 00:00:00	16332319
2	NULL	{"cpf": 2, "data_FIM": "2020-03-12 ...	2021-03-14 00:00:00	16386658
3	NULL	{"cpf": 3, "data_FIM": "2020-09-15 ...	2021-03-14 00:00:00	16401038
4	NULL	{"cpf": 4, "data_FIM": "2020-03-02 ...	2021-03-14 00:00:00	16413295
5	NULL	{"cpf": 5, "data_FIM": "2020-01-23 ...	2021-03-14 00:00:00	16415470
6	NULL	{"cpf": 6, "data_FIM": "2020-01-10 ...	2021-03-14 00:00:00	16441929
7	NULL	{"cpf": 7, "data_FIM": "2020-01-10 ...	2021-03-14 00:00:00	16443199
8	NULL	{"cpf": 8, "data_FIM": "2020-03-21 ...	2021-03-14 00:00:00	16450992
9	NULL	{"cpf": 9, "data_FIM": "2020-04-06 ...	2021-03-14 00:00:00	16451110
10	NULL	{"cpf": 10, "data_FIM": "2020-02-14...	2021-03-14 00:00:00	16460934
11	NULL	{"cpf": 11, "data_FIM": "2020-02-08...	2021-03-14 00:00:00	16465411

Figura 16: Logs gerados pela *trigger*. Fonte: Autor.

## C.2 View

### C.2.1 Consulta

```

CREATE view VIAGENS_DB.mostra_meio_tranporte as
SELECT mt.nome "Meio de Transporte", count(*) Total
FROM VIAGENS_DB.PASSAGEMM p, VIAGENS_DB.MEIO_TRANSPORTE mt
WHERE p.ID_MEIO_TRANSPORTE = mt.ID_MEIO_TRANSPORTE
GROUP by mt.nome

```

```
ORDER by Total desc;
```

### C.2.2 Resultado

#	Meio de Transporte	Total
1	aereo	102762
2	rodoviario	3408
3	fluvial	843
4	maritimo	2

Figura 17: Informação gerada pela *View*. Fonte: Autor.

## C.3 Procedure

### C.3.1 Consulta

```
CREATE PROCEDURE VIAGENS_DB.países_visitados()
  SELECT p.nome as Pais, count(*)
    as Total_Viagens
  FROM
    VIAGENS_DB.PAIS p,
    VIAGENS_DB.PASSAGEM pas,
    VIAGENS_DB.PAIS_CIDADE_UF pcu
  WHERE pcu.ID_PAIS_CIDADE_UF = pas.ID_PAIS_CIDADE_UF_DESTINO_IDA
    and pcu.ID_PAIS = p.ID_PAIS
  GROUP by p.nome
  ORDER by Total_Viagens desc
  LIMIT 5;
END
```

### C.3.2 Resultado

## C.4 Total gasto em viagens

### C.4.1 Consulta

```
SELECT sum(round(pag.valor,2)) total
FROM
  VIAGENS_DB.VIAGEM v,
  VIAGENS_DB.PASSAGEM pas,
  VIAGENS_DB.PAGAMENTO pag
```

#	País	Total_Viagens
1	brasil	104180
2	estados unidos da america	592
3	franca	197
4	portugal	162
5	suica	110

Figura 18: Informação gerada pela *Procedure*. Fonte: Autor.

```

WHERE
    v.ID_VIAGEM = pas.ID_VIAGEM
    and pas.ID_PASSAGEM = pag.ID_PASSAGEM
    and v.IS_REALIZADO=1;

```

#### C.4.2 Resultado

#	total
1	81959884.27

Figura 19: Informação gerada pela consulta. Fonte: Autor.

### C.5 Gasto por Tipo de viagem

#### C.5.1 Consulta

```

SELECT IF(V.IS_URGENTE = 1, 'URGENTE', 'NÃO URGENTE')
"CLASSIFICAÇÃO", SUM(ROUND(PAG.VALOR,2)) TOTAL
FROM
    VIAGEM_FINAL.VIAGEM V,
    VIAGEM_FINAL.PASSAGEM PAS,
    VIAGEM_FINAL.PAGAMENTO PAG
WHERE V.ID_VIAGEM=PAS.ID_VIAGEM
AND PAS.ID_PASSAGEM=PAG.ID_PASSAGEM
AND V.IS_REALIZADO=1
GROUP BY V.IS_URGENTE;

```

### C.5.2 Resultado

#	classificação	total
1	urgente	58690402.33
2	no urgente	23269481.94

Figura 20: Informação gerada pela consulta. Fonte: Autor.

## C.6 Média de Gasto por Ministério

### C.6.1 Consulta

```
SELECT
    m.nome "Ministério",
    avg(round(pag.valor,2)) Total
FROM
    VIAGENS_DB.VIAGEM v,
    VIAGENS_DB.PASSAGEM pas,
    VIAGENS_DB.PAGAMENTO pag,
    VIAGENS_DB.MINISTERIO m
WHERE v.ID_VIAGEM = pas.ID_VIAGEM
and   pas.ID_PASSAGEM = pag.ID_PASSAGEM
and   v.ID_MINISTERIO = m.ID_MINISTERIO
and   v.IS_REALIZADO = 1
GROUP by m.nome
ORDER by total desc;
```

### C.6.2 Resultado

#	Ministério	Total
1	ministerio das relacoes exteriores	3930.594059
2	ministerio das comunicacoes	1324.415701
3	ministerio do meio ambiente	1219.214211
4	ministerio de minas e energia	1174.222566
5	ministerio da ciencia, tecnologia, in...	1162.414583
6	presidencia da republica	1143.935598

Figura 21: Informação gerada pela Média de gasto por Órgão por passagem. Fonte: Autor.

## C.7 Número de viagens por cidade

### C.7.1 Consulta

```
SELECT cid.nome as Cidade, count(*) as Total_Viagens
FROM
    VIAGENS_DB.CIDADE cid,
    VIAGENS_DB.PASSAGEM pas,
    VIAGENS_DB.PAIS_CIDADE_UF pcu
WHERE pcu.ID_PAIS_CIDADE_UF = pas.ID_PAIS_CIDADE_UF_DESTINO_IDA
and   pcu.ID_CIDADE = cid.ID_CIDADE
GROUP by cid.nome
ORDER by Total_Viagens desc
LIMIT 5;
```

### C.7.2 Resultado

#	Cidade	Total_Viagens
1	brasil	27843
2	rio de janeiro	11335
3	sao paulo	6722
4	manaus	5011
5	recife	3859

Figura 22: Informação gerada pela consulta. Fonte: Autor.

## C.8 Quantitativo de Viagem por Ministério

### C.8.1 Consulta

```
SELECT m.nome as "Ministério", count(*) as Total_Viagens
FROM VIAGENS_DB.VIAGEM v, VIAGENS_DB.MINISTERIO m
WHERE v.ID_MINISTERIO=m.ID_MINISTERIO
GROUP BY m.nome
ORDER BY Total_Viagens desc
LIMIT 5;
```

### C.8.2 Resultado

#	Ministério	Total_Viagens
1	ministerio da defesa	84037
2	ministerio da justica e seguranca p...	69985
3	sem informacao	46718
4	ministerio da educacao	31547
5	ministerio da agricultura, pecuaria e...	19782

Figura 23: Informação gerada pela consulta. Fonte: Autor.