Trabalho Prático - Fundamentos de Banco de Dados

Daniel Youssef, Gustavo Dutra e Vitória Lik.

Etapa 1

Definição da fonte de dados

A fonte de dados escolhida é a página sobre a lista de países que têm patrimônios reconhecidos pela UNESCO.

World Heritage Sites by country

33 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

(Redirected from List of World Heritage Sites by country)

As of July 2024, there are a total of 1,223 World Heritage Sites located across 168 countries, of which 952 are cultural, 231 are natural, and 40 are mixed properties.[1] The countries have been divided by the World Heritage Committee into five geographical regions: Africa, the Arab States, Asia and the Pacific, Europe and North America, and Latin America and the Caribbean. With 60 selected areas, Italy is the country with the most sites, followed by China with 59, and Germany with 54.[2]

World Heritage Sites by country as of January 2024

Of the 196 states parties of the World Heritage Convention, 28 have no properties inscribed on the World Heritage List: The Bahamas, Bhutan, Brunei, Burundi, the Comoros, the Cook Islands, Djibouti, Equatorial Guinea, Eswatini, Grenada, Guinea-

Etapa 1: Definição dos elementos da normalização

A tabela está ÑÑ, será normalizada para 3FN e que desconsidera a coluna ‘shared sites’

List of countries with World Heritage Sites [edit]

Appearance hide

Country	Cultural sites	Natural sites	Mixed sites	Total sites	Shared sites[3]	UNESCO region[4]
Afghanistan	2			2		Asia and the Pacific
Albania	2	1[note 1]	1[note 2]	4	2	Europe and North America
Algeria	6		1	7		Arab States
Andorra	1			1		Europe and North America
Angola	1			1		Africa
Antigua and Barbuda	1			1		Latin America & the Caribbean

Etapa 2.1: Normalização dos dados - 1FN e 2FN

1FN

Heritage_sites_counts(Country,Cultural_sites,Natural_sites,Mixed_sites,Regions,Notes**)**

2FN

Regions (id_regions, name)

Site_types (id_site_types, type_name)

Heritage_sites_counts(Country,Cultural_sites,Natural_sites,Mixed_sites,Regions,Notes**)**

id_regions referencia Regions

id_sites_types referencia Site_types

Etapa 2.2: Normalização dos dados - 3FN

Regions (id_regions, name)
Countries (id_countries, name, region_id)
region_id referencia Regions(id_regions)

Site_types (id_site_types, type_name)

Heritage_sites_counts (id_heritage_sites, country_id, site_type_id, site_count)
country_id referencia Countries(id_countries)
site_type_id referencia Site_types(id_site_types)

Notes (id_note, tag, description)

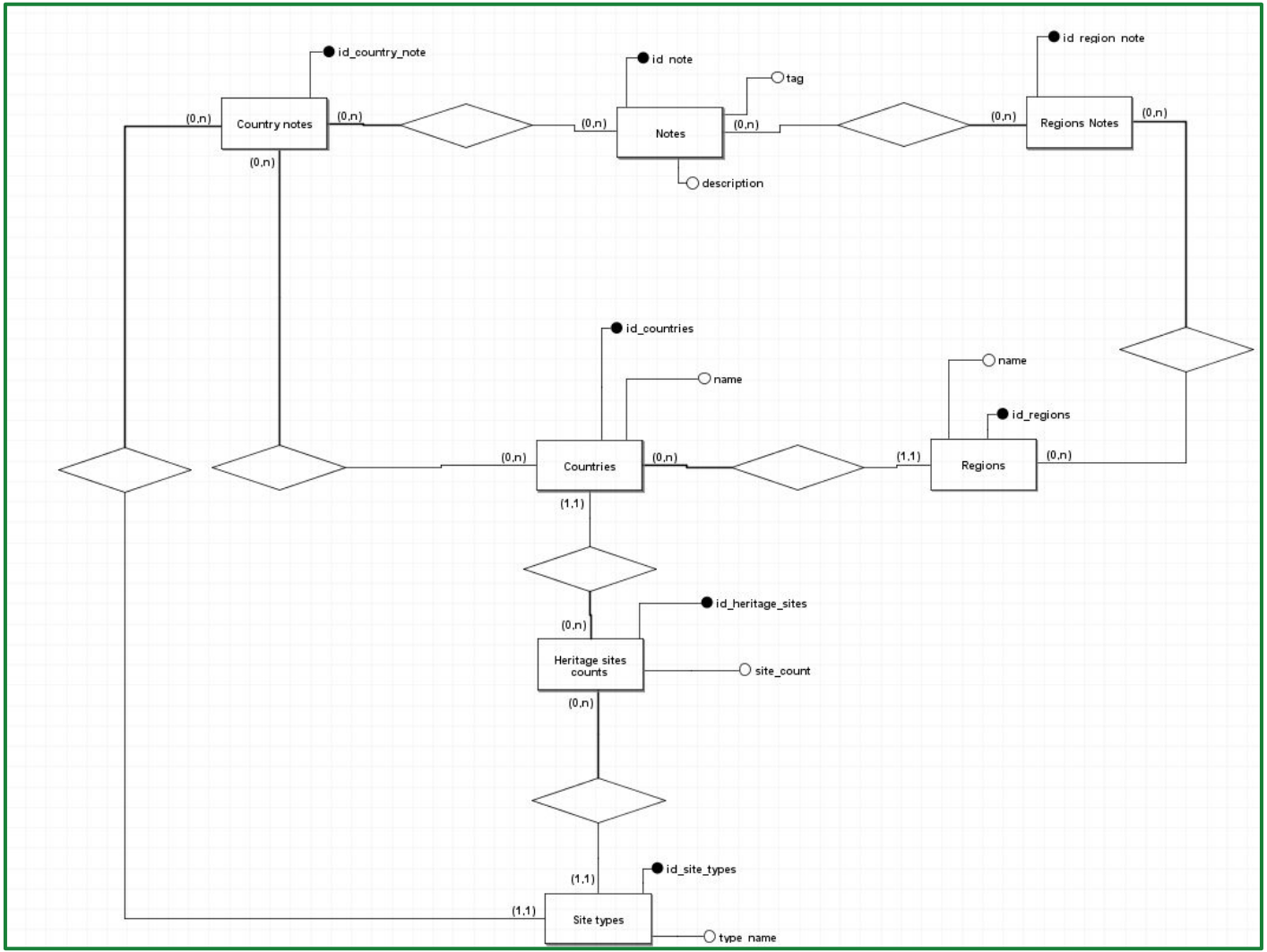
Region_notes (id_region_note, region_id, note_id)
region_id referencia Regions(id_regions)
note_id referencia Notes(id_note)

Country_notes (id_country_note, country_id, note_id, site_type_id)
country_id referencia Countries(id_countries)
note_id referencia Notes(id_note)
site_type_id referencia Site_types(id_site_types)

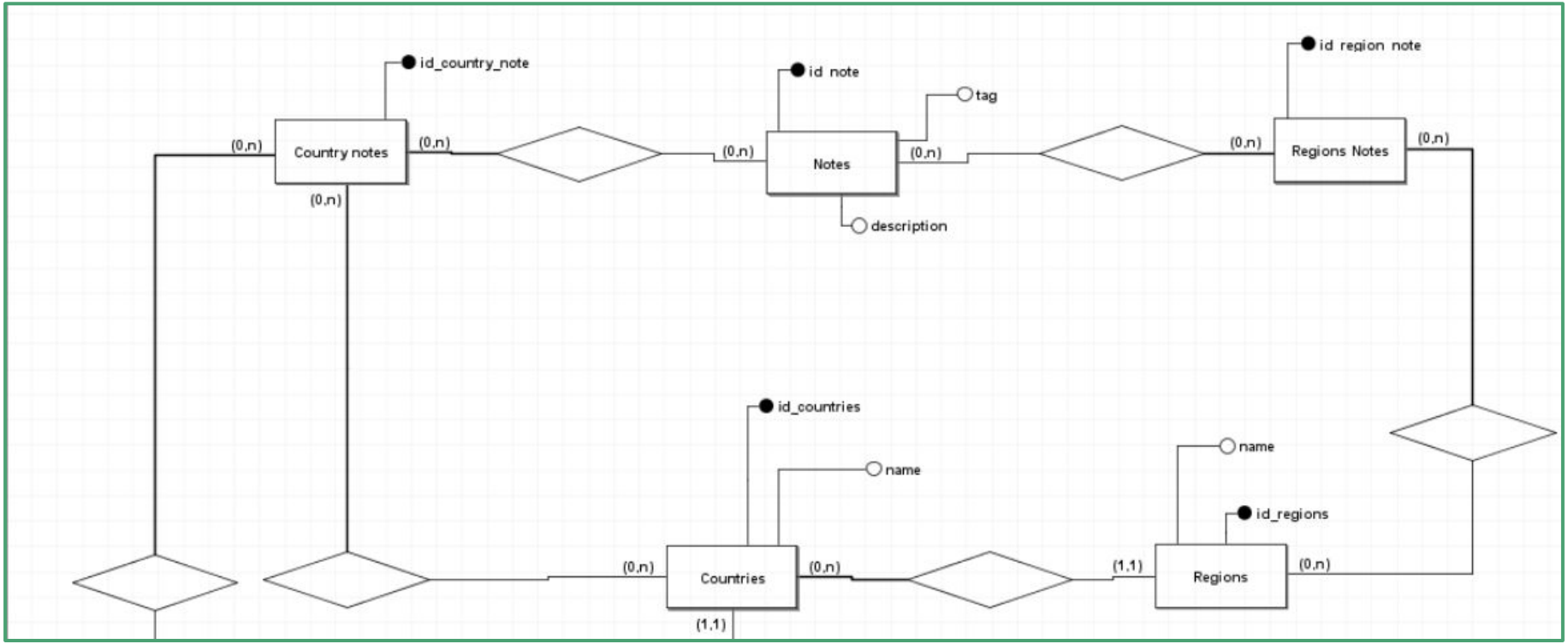
ETAPA 2.3:

MODELO

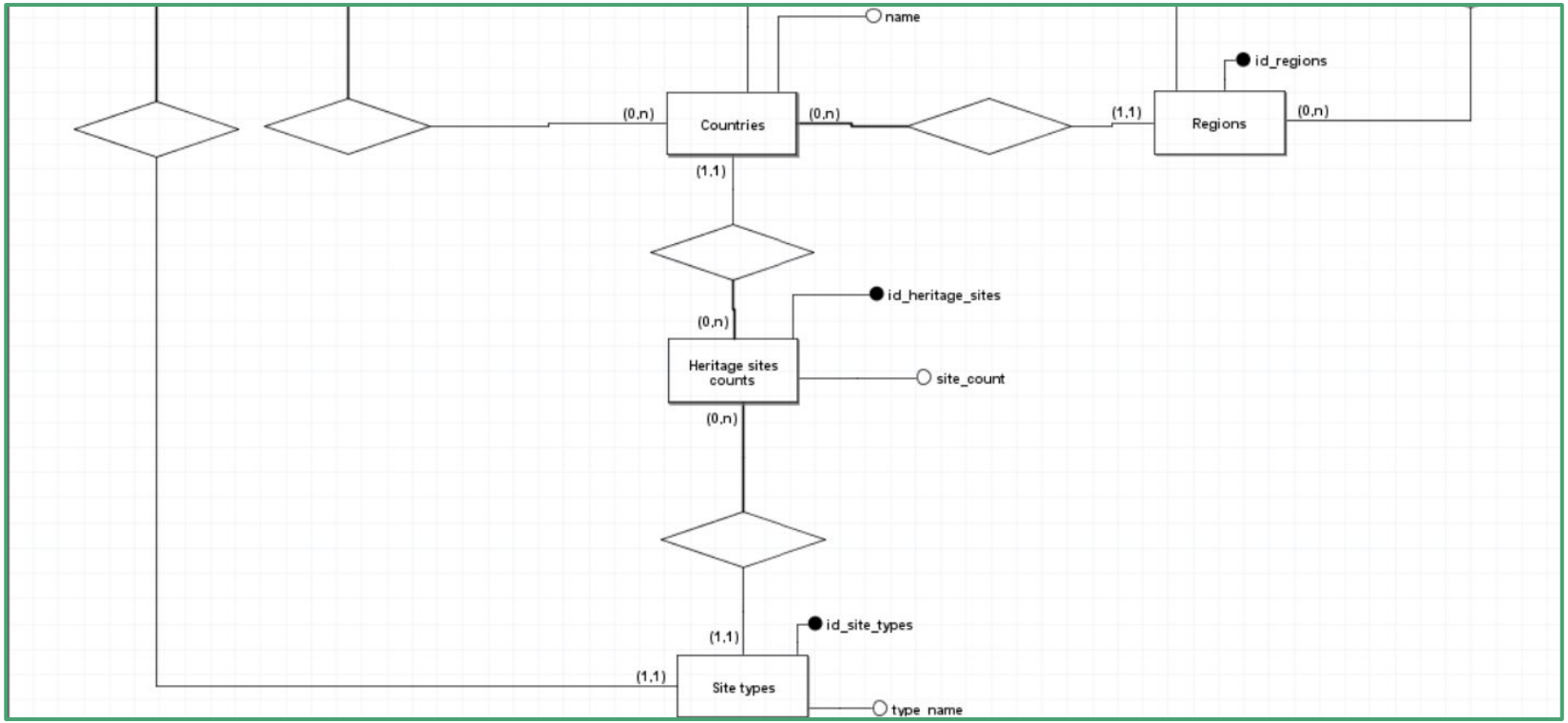
ER



Modelo ER - 1


id_country_note
Country notes
(0,n)
(0,n)
(0,n)
id note
tag
Notes
(0,n)
(0,n)
description
id region note
(0,n)
Regions Notes
(0,n)
id_countries
name
name
id_regions
(0,n)
Countries
(0,n)
(1,1)
Regions
(0,n)
(1,1)

Modelo ER - 2


name
id_regions
(0,n)
Countries
(0,n)
(1,1)
Regions
(0,n)
(1,1)
id_heritage_sites
(0,n)
Heritage sites counts
site_count
(0,n)
id_site_types
(1,1)
(1,1)
Site types
type_name

Etapa 3

Script para criação e alimentação do banco de dados

Etapa 3.1: Criação do banco de dados

```python
# 8. Conexão e metadados do MySQL    "Conexão": Unknown word.
engine = create_engine('mysql+pymysql://usuario:1234@localhost:3306/unesco_db')    "usuario": Unknown word.
metadata = MetaData()

# 9. Definição das tabelas    "Definição": Unknown word.
regions = Table('regions', metadata,
                Column('id_regions', Integer, primary_key=True, autoincrement=True),
                Column('name', String(100), unique=True, nullable=False))

countries = Table('countries', metadata,
                  Column('id_countries', Integer, primary_key=True, autoincrement=True),
                  Column('name', String(100), unique=True, nullable=False),
                  Column('region_id', Integer, ForeignKey('regions.id_regions')))

site_types = Table('site_types', metadata,
                   Column('id_site_types', Integer, primary_key=True, autoincrement=True),
                   Column('type_name', String(50), unique=True, nullable=False))

heritage_sites_counts = Table('heritage_sites_counts', metadata,
                              Column('id_heritage_sites', Integer, primary_key=True, autoincrement=True),
                              Column('country_id', Integer, ForeignKey('countries.id_countries')),
                              Column('site_type_id', Integer, ForeignKey('site_types.id_site_types')),
                              Column('site_count', Integer, default=0))
```

```python
notes = Table('notes', metadata,
              Column('id_note', Integer, primary_key=True, autoincrement=True),
              Column('tag', String(50), unique=True, nullable=False),
              Column('description', Text))

region_notes = Table('region_notes', metadata,
                     Column('id_region_note', Integer, primary_key=True, autoincrement=True),
                     Column('region_id', Integer, ForeignKey('regions.id_regions')),
                     Column('note_id', Integer, ForeignKey('notes.id_note')))

country_notes = Table('country_notes', metadata,
                      Column('id_country_note', Integer, primary_key=True, autoincrement=True),
                      Column('country_id', Integer, ForeignKey('countries.id_countries')),
                      Column('note_id', Integer, ForeignKey('notes.id_note')),
                      Column('site_type_id', Integer, ForeignKey('site_types.id_site_types')))

#10. Criar tabelas no banco    "Criar": Unknown word.
metadata.create_all(engine)
```

Etapa 3.1: Alimentação do banco de dados

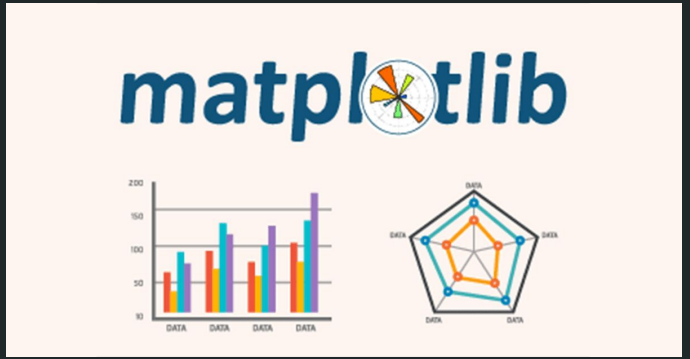
```python
# 11. Inserção de dados    "Inserção": Unknown word.
with engine.begin() as conn:

    # Inserido todas notações necessárias    "Inserido": Unknown word.
    for tag, description in meaningful_notes.items():
        # Verifica caso já exista notas no banco    "Verifica": Unknown word.
        exists = conn.execute(
            select(notes.c.id_note).where(notes.c.tag == tag)
        ).fetchone()

        if not exists:
            # Inserção de notas    "Inserção": Unknown word.
            conn.execute(
                notes.insert().values(
                    tag=tag,
                    description=description
                )
            )
            print(f"Inserted note {tag}")
        else:
            print(f"Note {tag} already exists, skipping")
        You, anteontem • commit: Implementando o sistemas de notas e tra…
    sample_notes = conn.execute(
        select(notes).limit(5)
    ).fetchall()
```

```python
# Inserir regiões    "Inserir": Unknown word.
region_map = {}
for reg in unesco_df['Region_clean'].unique():
    result = conn.execute(select(regions.c.id_regions).where(regions.c.name == reg)).fetchone()
    if result:
        region_id = result[0]
    else:
        conn.execute(regions.insert().values(name=reg))
        region_id = conn.execute(select(regions.c.id_regions).where(regions.c.name == reg)).fetchone()[0]
    region_map[reg] = region_id
```

```python
# Inserir notas globais e criar mapa (com descrição da extração)    "Inserir": Unknown word.
note_map = {}
# Notas da região    "Notas": Unknown word.
for note_list in unesco_df['Region_notes']:
    for note in note_list:
        if note not in note_map:
            result = conn.execute(select(notes.c.id_note).where(notes.c.tag == note)).fetchone()
            if result:
                note_id = result[0]
            else:
                description = note_descriptions.get(note, None)
                conn.execute(notes.insert().values(tag=note, description=description))
                note_id = conn.execute(select(notes.c.id_note).where(notes.c.tag == note)).fetchone()[0]
```

Etapa 4

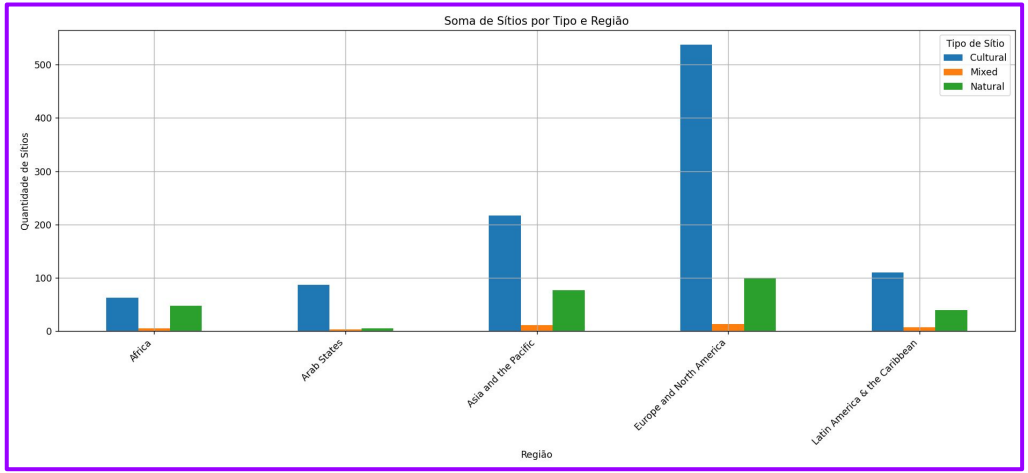
Script para visualização das consultas SQL



Etapa 4.1: Visualização dos dados

Selecionar a quantidade de sítios agrupados por tipo em uma região específica

```
SELECT r.name, s.type_name, sum(h.site_count)
AS soma

FROM site_types s join heritage_sites_counts h
ON h.site_type_id=s.id_site_types

JOIN countries c ON c.id_countries=h.country_id

JOIN regions r ON c.region_id=r.id_regions

GROUP BY s.type_name, r.name
```



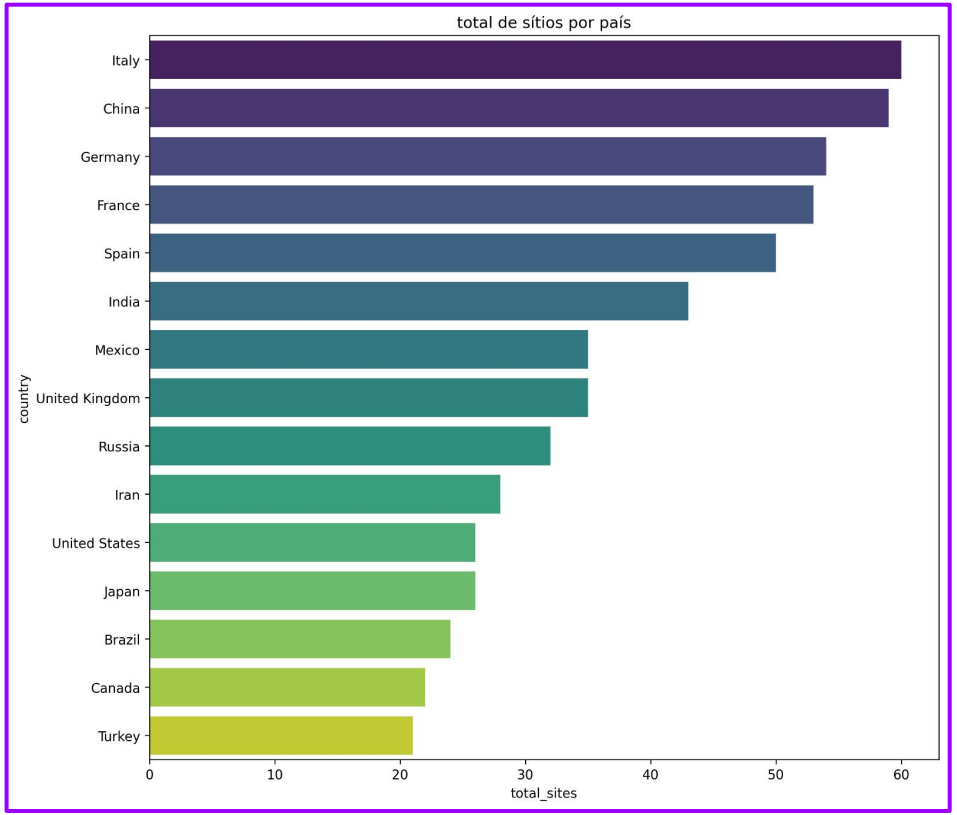
Etapa 4.2: Visualização dos dados

Mostre o total de sítios por país

SELECT c.name, sum(h.site_count) **AS** numero_de_sitios

FROM countries c **JOIN** heritage_sites_counts h **ON** c.id_countries=h.country_id

GROUP BY c.name



Etapa 4.3: Visualização dos dados

Mostre os países europeus com mais de 10 sítios, ordene em ordem decrescente

SELECT c.name, h.site_count

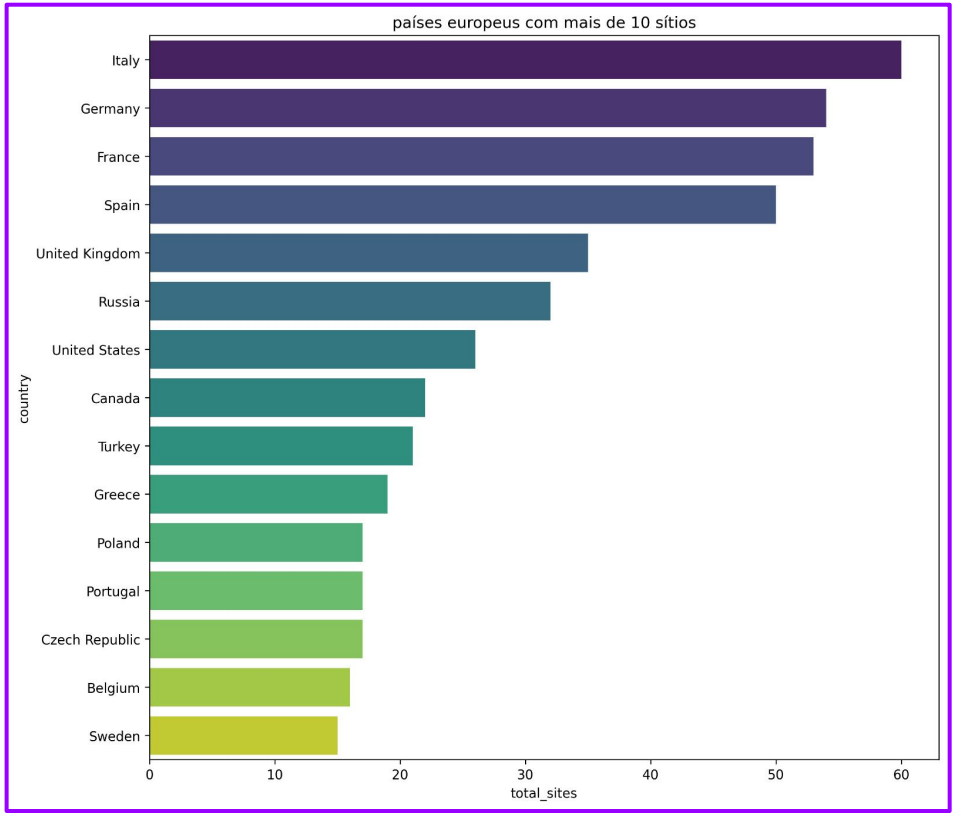
FROM regions r **JOIN** countries c **ON** c.region_id=r.id_regions

JOIN heritage_sites_counts h **ON** c.id_countries=h.country_id

WHERE r.name like '%Europe%'

AND h.site_count > 10

ORDER BY h.site_count **DESC**



Etapa 4.3: Visualização dos dados

Mostre os países que não possuem sítios mistos

```
SELECT DISTINCT c.name

FROM countries c JOIN heritage_sites_counts h
ON c.id_countries=h.country_id

WHERE h.country_id NOT IN(SELECT
h.country_id FROM heritage_sites_counts h JOIN
site_types s ON s.id_site_types=h.site_type_id

                                                W
HERE type_name='Mixed' and site_count = 0)
```

Países sem Sítios Mistos

Países (1ª Coluna)	Países (2ª Coluna)
Albania	Lesotho
Algeria	Mali
Australia	Mexico
Brazil	New Zealand
Canada	North Macedonia
Chad	Palau
China	Peru
Colombia	South Africa
France	Spain
Gabon	Sweden
Greece	Tanzania
Guatemala	Turkey
India	United Kingdom
Iraq	United States
Jamaica	Vietnam
Jordan	

Etapa 4.4: Visualização dos dados

Mostre as regiões que concentram mais sítios

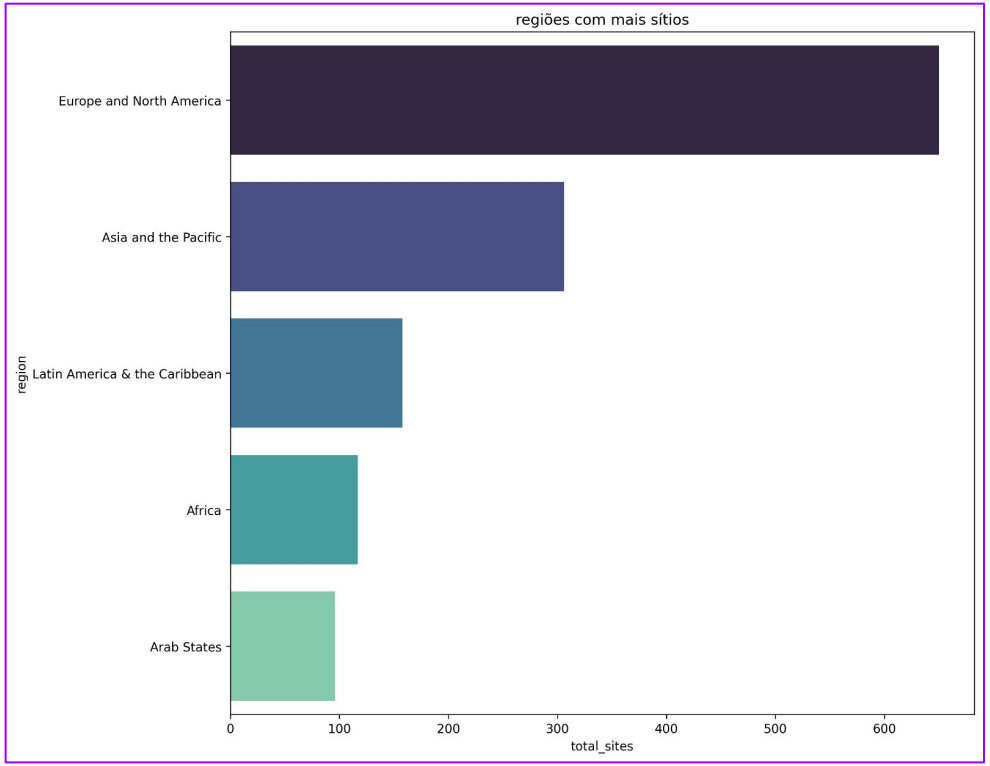
SELECT r.name, **SUM**(h.site_count) as soma_de_sitios

FROM regions r **JOIN** countries c **ON** c.region_id=r.id_regions

JOIN heritage_sites_counts h **ON** c.id_countries=h.country_id

GROUP BY 1

ORDER BY 2 **DESC**



Obrigado(a) por sua atenção!