データベース入門

PYTHONで学ぶデータベース入門



Jason Myers

Co-Author of Essential SQLAlchemy and Software Engineer

datacamp

データベースはテーブルで構成されます

Census

state	sex	age	pop2000	pop2008
New York	F	0	120355	122194
New York	F	1	118219	119661
New York	F	2	119577	116413

State_Fact

name	abbreviation	type
New York	NY	state
Washington DC	DC	capitol
Washington	WA	state

テーブルは列と行で構成されます

Census

state	sex	age	pop2000	pop2008
New York	F	0	120355	122194
New York	F	1	118219	119661
New York	F	2	119577	116413

テーブルは列と行で構成されます

Census

state	sex	age	pop2000	pop2008
New York	F	0	120355	122194
New York	F	1	118219	119661
New York	F	2	119577	116413

テーブル間には関係があります

Census

state	sex	age	pop2000	pop2008
New York	F	0	120355	122194
New York	F	1	118219	119661
New York	F	2	119577	116413

State_Fact

name	abbreviation	type
New York	NY	state
Washington DC	DC	capitol
Washington	WA	state

練習しましょう！

PYTHONで学ぶデータベース入門

データベースに接続する

PYTHONで学ぶデータベース入門



Jason Myers

Co-Author of Essential SQLAlchemy and Software Engineer

datacamp

SQLAlchemy の概要

- 主要な構成要素は2つ
  - Core（リレーショナルモデル重視）
  - ORM（ユーザーデータモデル重視）

データベースには多くの種類がある

- SQLite
- PostgreSQL
- MySQL
- Microsoft SQL Server
- Oracle SQL
- その他多数

データベースへの接続

```
from sqlalchemy import create_engine
engine = create_engine('sqlite:///census_nyc.sqlite')
connection = engine.connect()
```

- Engine: SQLAlchemy からDBへの共通インターフェース
- 接続文字列: DBの場所と認証に必要な情報一式

接続文字列について一言

```
'sqlite:///census_nyc.sqlite'
```

Driver + Dialect

接続文字列について一言

```
'sqlite:///census_nyc.sqlite'
```

ファイル名

データベースの中身は？

クエリの前に、中身を確認します。たとえばテーブル名:

```
from sqlalchemy import create_engine
engine = create_engine('sqlite:///census_nyc.sqlite')
print(engine.table_names())
```

```
['census', 'state_fact']
```

リフレクション

リフレクションはDBを読み取り、SQLAlchemy の `Table` オブジェクトを作成します

```
from sqlalchemy import MetaData, Table
metadata = MetaData()
census = Table('census', metadata, autoload=True,
  autoload_with=engine)
print(repr(census))
```

```
Table('census', MetaData(bind=None), Column('state', VARCHAR(
length=30), table=<census>), Column('sex', VARCHAR(length=1),
table=<census>), Column('age', INTEGER(), table=<census>),
Column('pop2000', INTEGER(), table=<census>), Column('pop2008',
INTEGER(), table=<census>), schema=None)
```

練習しましょう！

PYTHONで学ぶデータベース入門

SQLクエリ入門

PYTHONで学ぶデータベース入門



Jason Myers

Co-Author of Essential SQLAlchemy and Software Engineer

datacamp

SQL文

- データの選択・挿入・更新・削除
- データの作成・変更

基本のSQLクエリ

```
SELECT column_name FROM table_name
```

- `SELECT pop2008 FROM People`
- `SELECT * FROM People`

基本のSQLクエリ

```
from sqlalchemy import create_engine
engine = create_engine('sqlite:///census_nyc.sqlite')
connection = engine.connect()
stmt = 'SELECT * FROM people'
result_proxy = connection.execute(stmt)
results = result_proxy.fetchall()
```

ResultProxy と ResultSet

```
result_proxy = connection.execute(stmt)


results = result_proxy.fetchall()
```

- `result_proxy` は `ResultProxy`
- `results` は `ResultSet`

ResultSetの扱い方

```
first_row = results[0]
print(first_row)
```

```
('Illinois', 'M', 0, 89600, 95012)
```

```
print(first_row.keys())
```

```
['state', 'sex', 'age', 'pop2000', 'pop2008']
```

```
print(first_row.state)
```

```
'Illinois'
```

SQLAlchemyでクエリを作成

- Python風にSQL文を構築できる
- バックエンドDBの違いを吸収する

SQLAlchemyでのクエリ実行

```
from sqlalchemy import Table, MetaData
metadata = MetaData()
census = Table('census', metadata, autoload=True,
  autoload_with=engine)
stmt = select([census])
results = connection.execute(stmt).fetchall()
```

SQLAlchemyのSELECT文

- 1つ以上のTable/Columnのリストが必要
- Tableを渡すとその全列を選択

```
stmt = select([census])
print(stmt)
```

```
'SELECT * from CENSUS'
```

Ayo berlatih!

PYTHONで学ぶデータベース入門

おめでとうございます！

PYTHONで学ぶデータベース入門



Jason Myers

Co-Author of Essential SQLAlchemy and Software Engineer

datacamp

すでにできること

- 関係モデルを理解している
- 基本的なSQLクエリを作成できる

この後の内容

- SQLクエリ力を強化
- SQLAlchemyでDBから有用な情報を抽出する方法を学ぶ
- リレーショナルDBの作成と書き込みを学ぶ
- 米国国勢調査データを深掘り！

次のチャプターでお会いしましょう！

PYTHONで学ぶデータベース入門