Census

PYTHON □□□□□□□□□□□□□□□□□□□□□□□□□□□□



Jason Myers

Co-Author of Essential SQLAlchemy and Software Engineer

datacamp

Census

- SQLAlchemy □□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□
- Data Science □□□□□□□□□

1: SQLAlchemy □□□□□□□□□□□□

- engine `MetaData`

```
from sqlalchemy import create_engine, MetaData
engine = create_engine('sqlite:///census_nyc.sqlite')
metadata = MetaData()
```

1: SQLAlchemy □□□□□□□□□□□□

- census

```
from sqlalchemy import (Table, Column, String,
       Integer, Decimal, Boolean)
employees = Table('employees', metadata,
       Column('id', Integer()),
       Column('name', String(255)),
       Column('salary', Decimal()),
       Column('active', Boolean()))
metadata.create_all(engine)
```

!

PYTHON □□□□□□□□□□□□□□□□□□□□□□□□□□□□

[illegible]

PYTHON [illegible]



Jason Myers

Co-Author of Essential SQLAlchemy and Software Engineer

datacamp

2:

- CSV values list

```
values_list = []
for row in csv_reader:
    data = {'state': row[0], 'sex': row[1], 'age': row[2],
            'pop2000': row[3], 'pop2008': row[4]}
    values_list.append(data)
```

2:

- values list census

```
from sqlalchemy import insert
stmt = insert(employees)
result_proxy = connection.execute(stmt, values_list)
print(result_proxy.rowcount)
```

```
2
```

!

PYTHON □□□□□□□□□□□□□□□□□□□□□□□□□□□□

[illegible]

PYTHON [illegible]



Jason Myers

Co-Author of Essential SQLAlchemy and Software Engineer

datacamp

3: Data Science □□□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
from sqlalchemy import select
stmt = select([census.columns.sex,
          (func.sum(census.columns.pop2008 *
           census.columns.age) /
           func.sum(census.columns.pop2008)
         ).label('average_age')])
stmt = stmt.group_by('census.columns.sex')
results = connection.execute(stmt).fetchall()
```

3: Data Science □□□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
from sqlalchemy import case, cast, Float
stmt = select([
        (func.sum(
            case([
                (census.columns.state == 'New York',
                 census.columns.pop2008)
            ], else_=0)) /
         cast(func.sum(census.columns.pop2008),
              Float) * 100).label('ny_percent')])
```

3: Data Science □□□□□□□□□

- 5
  2000 2008

```
stmt = select([census.columns.age,
          (census.columns.pop2008-
          census.columns.pop2000).label('pop_change')
      ])
stmt = stmt.order_by('pop_change')
stmt = stmt.limit(5)
```

!

PYTHON □□□□□□□□□□□□□□□□□□□□□□□□□□□□

PYTHON !



Jason Myers

Co-Author of Essential SQLAlchemy and Software Engineer

datacamp

!

PYTHON □□□□□□□□□□□□□□□□□□□□□□□□□□□□