

Veri Tabanı Sistemleri

Uygulama Geliştirme

H. Turgut Uyar Şule Öğüdücü

2002-2016

1 / 58

License

../license

© 2002-2016 T. Uyar, Ş. Öğüdücü

You are free to:

- Share – copy and redistribute the material in any medium or format
- Adapt – remix, transform, and build upon the material

Under the following terms:

- Attribution – You must give appropriate credit, provide a link to the license, and indicate if changes were made.
- NonCommercial – You may not use the material for commercial purposes.
- ShareAlike – If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

For more information:

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Read the full license:

<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

2 / 58

Konular

Veri Tabanı Arayüzleri

Giriş
İşlemler
Hata Denetimi
Komutlar

Nesne/Bağıntı Eşleştirmesi

Giriş
SQLAlchemy
Sorgulamalar
Dış Anahtarlar

3 / 58

Giriş

- uygulama kodunda veri işlemleri nasıl yapılacak?
- veri tabanı sunucusuna bağlan
- sunucu, veri tabanı, kullanıcı adı, parola
- işlemleri yürüt
- sonuçları uyarla
- bağlantıyı kopar

4 / 58

Amaçlar

- ▶ kod belirli bir ürüne bağlı olmamalı
- ▶ başka bir ürüne kolayca taşınabilmeli
- ▶ soyutlama katmanları performans problemlerine yol açıyor
- ▶ örneğin ODBC standart ama yavaş
- ▶ sürücüler dil standart arayüzlerine uyarak gerçekleştirir
- ▶ Java: JDBC, Python: DBAPI

5 / 58

Python DBAPI

- ▶ sürücüyü yükle
- ▶ başka sürücülere kolay uyarlamak için adını değiştir

örnek

```
import psycopg2 as dbapi2
# import sqlite3 as dbapi2
```

6 / 58

Bağlantı

- ▶ bağlantı bilgileri: kullanıcı adı, parola, sunucu, veri tabanı
- ▶ veri kaynağı adı (DSN):
user=.. password=.. host=.. port=.. dbname=..
- ▶ tekdüzen kaynak tanımlayıcısı (URI):
protocol://user:password@host:port/dbname

örnekler

```
user='vagrant' password='vagrant' host='localhost'
port=5432 dbname='itucsdB'
```

```
postgres://vagrant:vagrant@localhost:5432/itucsdB
```

7 / 58

Bağlantı Örneği

```
dsn = """user='vagrant' password='vagrant'
        host='localhost' port=5432 dbname='itucsdB'"""
connection = dbapi2.connect(dsn)

# database operations

connection.close()
```

8 / 58

Güncelleme İşlemleri

- ▶ güncelleme işlemleri için (insert, delete, update, create, ...)
- ▶ bağlantı için bir imleç tanımla
- ▶ sorguları çalıştır
- ▶ bekleyen değişiklikleri işle
- ▶ imleci kapat

9 / 58

Güncelleme Örneği

```
connection = dbapi2.connect(dsn)
cursor = connection.cursor()
statement = """CREATE TABLE PERSON (
    ID SERIAL PRIMARY KEY,
    NAME VARCHAR(40) UNIQUE NOT NULL
)"""
cursor.execute(statement)
connection.commit()
cursor.close()
connection.close()
```

10 / 58

Sorgulama İşlemleri

- ▶ sorgulama işlemleri için (select)
- ▶ bağlantı için bir imleç tanımla
- ▶ sorguları çalıştır
- ▶ imleçle döngü içinde satırlar üzerinde dolaş (her satır bir çoklu)
- ▶ imleci kapat

11 / 58

Sorgulama Örneği

```
connection = dbapi2.connect(dsn)
cursor = connection.cursor()
statement = """SELECT TITLE, SCORE FROM MOVIE
    WHERE (YR = 1999)"""
cursor.execute(statement)
for row in cursor:
    title, score = row
    print('{ }: {}'.format(title, score))
cursor.close()
connection.close()
```

12 / 58

Sorgulama Örneği

- ▶ çoklu ataması ile daha basit işlem

```
for row in cursor:
    title, score = row
    print('{}: {}'.format(title, score))

for title, score in cursor:
    print('{}: {}'.format(title, score))
```

13 / 58

Sorgulama Örnekleri

- ▶ filmler ve yönetmenleri

```
statement = """SELECT TITLE, NAME
                FROM MOVIE JOIN PERSON
                ON (MOVIE.DIRECTORID = PERSON.ID)"""
cursor.execute(statement)
for title, name in cursor:
    print('{}: {}'.format(title, name))
```

14 / 58

Hata Denetimi

- ▶ veri tabanı işlemleri ile ilgili hatalar
- ▶ hata durumunda işlemi geri al (**except**)
- ▶ açık bütün kaynakları kapat (**finally**)

15 / 58

Şablon

```
try:
    connection = dbapi2.connect(dsn)
    cursor = connection.cursor()
    cursor.execute(statement)
    connection.commit()
    cursor.close()
except dbapi2.DatabaseError:
    if connection:
        connection.rollback()
finally:
    if connection:
        connection.close()
```

16 / 58

Bağlantı Bağlam Yöneticileri

- ▶ bazı sürücülerde bağlantılar bağlam yöneticileridir: `with`
- ▶ automatic commit (try), rollback (except), close (finally)
- ▶ şablon:

```
with dbapi2.connect(dsn) as connection:  
    cursor = connection.cursor()  
    cursor.execute(statement)  
    cursor.close()
```

17 / 58

Bağlantı Bağlam Yöneticisi Örneği

```
with dbapi2.connect(dsn) as connection:  
    cursor = connection.cursor()  
    statement = """CREATE TABLE MOVIE (  
        ID SERIAL PRIMARY KEY,  
        TITLE VARCHAR(80),  
        YR NUMERIC(4),  
        SCORE FLOAT,  
        VOTES INTEGER DEFAULT 0,  
        DIRECTORID INTEGER REFERENCES PERSON (ID)  
    )"""  
    cursor.execute(statement)  
    cursor.close()
```

18 / 58

İmleç Bağlam Yöneticileri

- ▶ bazı sürücülerde imleçler de bağlam yöneticileridir
- ▶ otomatik kapama
- ▶ template:

```
with dbapi2.connect(dsn) as connection:  
    with connection.cursor() as cursor:  
        cursor.execute(statement)
```

19 / 58

İmleç Bağlam Yöneticisi Örneği

```
with dbapi2.connect(dsn) as connection:  
    with connection.cursor() as cursor:  
        statement = """CREATE TABLE CASTING (  
            MOVIEID INTEGER REFERENCES MOVIE (ID),  
            ACTORID INTEGER REFERENCES PERSON (ID),  
            ORD INTEGER,  
            PRIMARY KEY (MOVIEID, ACTORID)  
        )"""  
        cursor.execute(statement)
```

20 / 58

Komutlar

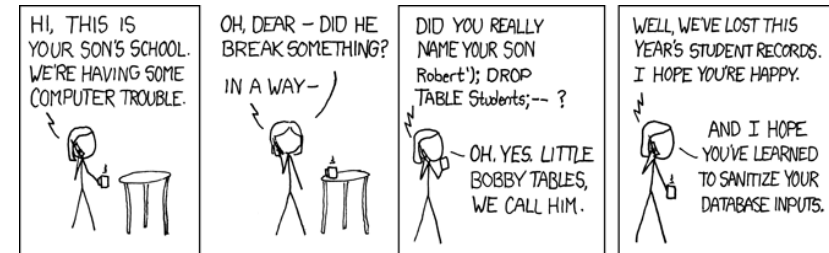
- ▶ komutları katar formatlama ile oluşturmak güvenli değil
- ▶ dış kaynaklardan yapılan girdilere asla güvenme
- ▶ **SQL enjeksiyonu** saldırıları

kötü örnek

```
name = input('What is your name? ')
statement = """INSERT INTO Students (Name)
              VALUES ('%s')""" % name
cursor.execute(statement)
```

21 / 58

SQL Enjeksiyonu Örneği



```
INSERT INTO Students (Name)
VALUES ('Robert'); DROP TABLE Students;-- '
INSERT INTO Students (Name)
VALUES ('Robert'); DROP TABLE Students;-- ')
```

<http://xkcd.com/327/>

22 / 58

Yer Tutucular

- ▶ değerler için yer tutucular
- ▶ farklı sürücülerin farklı biçimde: %s, ?, ...
- ▶ gerçek parametreler çoklu veya sözlük ile sağlanır

23 / 58

Yer Tutucu Örnekleri

- ▶ çoklu kullanarak:

```
statement = """INSERT INTO MOVIE (TITLE, YR)
              VALUES (%s, %s)"""
cursor.execute(statement, (title, year))
```
- ▶ sözlük kullanarak:

```
statement = """INSERT INTO MOVIE (TITLE, YR)
              VALUES %(title)s, %(year)s)"""
cursor.execute(statement, {'year': year,
                           'title': title})
```

24 / 58

Sonuçların Alınması

- ▶ imleçle döngü içinde dolaşmak yerine sonuçların alınması:
 .fetchall()
 .fetchone()

25 / 58

Sonuç Alma Örneği

- ▶ yönetmenler ve yönettikleri filmler

```
statement = """SELECT ID, NAME FROM PERSON"""
cursor.execute(statement)
people = cursor.fetchall()
for person_id, name in people:
    statement = """SELECT TITLE FROM MOVIE
                    WHERE (DIRECTORID = %s)"""
    cursor.execute(statement, (person_id,))
    directed = cursor.fetchall()
    print('{}:'.format(name))
    for (title,) in directed:
        print(' {}'.format(title))
```

26 / 58

Referanslar

Yardımcı Kaynak

- ▶ Python Database API Specification v2.0:
 <https://www.python.org/dev/peps/pep-0249/>

27 / 58

Problem

- ▶ veri modeli ile yazılım modeli arasında uyumsuzluk
- ▶ veri tabanı bağıntılar şeklinde: bağıntı, çoklu, dış anahtar, ...
- ▶ yazılım nesneye dayalı: nesne, yöntem, ...

28 / 58

Model Farkı Örneği

- filme oyuncu ekleme: SQL tanımları

```
CREATE TABLE MOVIE (ID INTEGER PRIMARY KEY,  
  TITLE VARCHAR(80) NOT NULL)
```

```
CREATE TABLE PERSON (ID INTEGER PRIMARY KEY,  
  NAME VARCHAR(40) NOT NULL)
```

```
CREATE TABLE CASTING (  
  MOVIEID INTEGER REFERENCES MOVIE (ID),  
  ACTORID INTEGER REFERENCES PERSON (ID),  
  PRIMARY KEY (MOVIEID, ACTORID)  
)
```

29 / 58

Model Farkı Örneği

- filme oyuncu ekleme: SQL işlemleri

```
INSERT INTO MOVIE (ID, TITLE)  
VALUES (110, 'Sleepy Hollow')
```

```
INSERT INTO PERSON (ID, NAME)  
VALUES (26, 'Johnny Depp')
```

```
INSERT INTO CASTING (MOVIEID, ACTORID)  
VALUES (110, 26)
```

30 / 58

Model Farkı Örneği

- filme oyuncu ekleme: Python tanımları

```
class Person:  
    def __init__(self, name):  
        self.name = name  
  
class Movie:  
    def __init__(self, title):  
        self.title = title  
        self.cast = []  
  
    def add_actor(self, person):  
        self.cast.append(person)
```

31 / 58

Model Farkı Örneği

- filme oyuncu ekleme: Python işlemleri

```
movie = Movie('Sleepy Hollow')  
actor = Person('Johnny Depp')  
movie.add_actor(actor)
```

32 / 58

Nesne/Bağıntı Dönüşümü

- ▶ yazılım bileşenleri veri tabanı bileşenleriyle eşleştirilir
- ▶ nesne arayüzünü SQL komutlarına çeviriyor

model	SQL	software
relation	table	class
tuple	row	object (instance)
attribute	column	attribute

33 / 58

SQLAlchemy

- ▶ nesne-bağıntı dönüştürücü
- ▶ bir Python sınıfı
- ▶ SQL tablo tanımı
- ▶ dönüştürücü sınıfı tabloya dönüştürür

34 / 58

Bağlantı Örneği

```
from sqlalchemy import create_engine

uri = 'postgres://vagrant:vagrant@localhost:5432/itucsdbs'
engine = create_engine(uri, echo=True)

from sqlalchemy import MetaData

metadata = MetaData()
```

35 / 58

Sınıf Örneği

```
class Movie:
    def __init__(self, title, year=None,
                  score=None, votes=None):
        self.title = title
        self.yr = year
        self.score = score
        self.votes = votes
```

36 / 58

Tablo Örneği

```
from sqlalchemy import Column, Table
from sqlalchemy import Float, Integer, String

movie_table = Table(
    'Movie', metadata,
    Column('id', Integer, primary_key=True),
    Column('title', String(80), nullable=False),
    Column('yr', Integer),
    Column('score', Float)
    Column('votes', Integer)
)
```

37 / 58

Dönüştürücü Örneği

```
from sqlalchemy.orm import mapper

mapper(Movie, movie_table)
```

38 / 58

Tablo Yaratma

```
metadata.create_all(bind=engine)
```

```
CREATE TABLE "Movie" (
    id SERIAL NOT NULL,
    title VARCHAR(80) NOT NULL,
    yr INTEGER,
    score FLOAT,
    votes INTEGER,
    PRIMARY KEY (id)
)
```

39 / 58

Oturumlar

- ▶ veri işlemleri oturumlar içinde yürütülür
- ▶ sonlandırma veya geri alma ile sonlandırılır
- ▶ oturumlar değişikliğe uğrayan veya yeni nesneleri izler

40 / 58

Oturum Örneği

```
from sqlalchemy.orm import sessionmaker
```

```
Session = sessionmaker(bind=engine)
session = Session()
```

41 / 58

Oturum Örneği: Güncelleme

```
movie.votes = 23283
session.commit()
```

```
UPDATE "Movie" SET votes=%(votes)s
WHERE "Movie".id = %(Movie_id)s
```

```
{'Movie_id': 1, 'votes': 23283}
```

42 / 58

Oturum Örneği: Silme

```
session.delete(movie)
session.commit()
```

```
DELETE FROM "Movie"
WHERE "Movie".id = %(id)s
```

```
{'id': 1}
```

43 / 58

Sorgulama Örnekleri

```
session.query(Movie)
```

```
SELECT "Movie".id AS "Movie_id",
       "Movie".title AS "Movie_title",
       "Movie".yr AS "Movie_yr",
       "Movie".score AS "Movie_score",
       "Movie".votes AS "Movie_votes"
FROM "Movie"
```

44 / 58

Sorgulama Örnekleri: Sütunları Seçme

```
session.query(Movie.title, Movie.score)
```

```
SELECT "Movie".title AS "Movie_title",  
       "Movie".score AS "Movie_score"  
FROM "Movie"
```

45 / 58

SQLAlchemy Örneği: Sıralama

```
session.query(Movie).order_by(Movie.yr)
```

```
SELECT "Movie".id AS "Movie_id",  
       "Movie".title AS "Movie_title",  
       "Movie".yr AS "Movie_yr",  
       "Movie".score AS "Movie_score",  
       "Movie".votes AS "Movie_votes"  
FROM "Movie"  
ORDER BY "Movie".yr
```

46 / 58

SQLAlchemy Örneği: Satırları Seçme

```
session.query(Movie).filter_by(yr=1999)
```

```
SELECT "Movie".id AS "Movie_id",  
       "Movie".title AS "Movie_title",  
       "Movie".yr AS "Movie_yr",  
       "Movie".score AS "Movie_score",  
       "Movie".votes AS "Movie_votes"  
FROM "Movie"  
WHERE "Movie".yr = %(yr_1)s  
  
{'yr_1': 1999}
```

47 / 58

SQLAlchemy Örneği: Yüklemlerle Satırları Seçme

```
session.query(Movie).filter(Movie.yr < 1999)
```

```
SELECT "Movie".id AS "Movie_id",  
       "Movie".title AS "Movie_title",  
       "Movie".yr AS "Movie_yr",  
       "Movie".score AS "Movie_score",  
       "Movie".votes AS "Movie_votes"  
FROM "Movie"  
WHERE "Movie".yr < %(yr_1)s  
  
{'yr_1': 1999}
```

48 / 58

Dış Anahtarlar

- ▶ tablo tanımlarına dış anahtar sütunlarını ekle
- ▶ dönüştürücüye "relationship" özelliği ekle
- ▶ özellik ismi başvuran tablodaki nitelik ismi
- ▶ "backref" parametresi başvuru yapılan tablodaki nitelik ismi

49 / 58

Dış Anahtar Örneği

```
class Person:
    def __init__(self, name):
        self.name = name

person_table = Table(
    'Person', metadata,
    Column('id', Integer, primary_key=True),
    Column('name', String(40), nullable=False,
           unique=True)
)

mapper(Person, person_table)
```

50 / 58

Dış Anahtar Örneği

```
from sqlalchemy import ForeignKey

movie_table = Table(
    'Movie', metadata,
    Column('id', Integer, primary_key=True),
    Column('title', String(80)),
    Column('yr', Integer),
    Column('score', Float),
    Column('votes', Integer),
    Column('directorid', Integer, ForeignKey('Person.id'))
)
```

51 / 58

Dış Anahtar Örneği

```
from sqlalchemy.orm import relationship

mapper(Movie, movie_table,
        properties={
            'director':
                relationship(Person,
                             backref='directed')
        })
```

52 / 58

Dış Anahtar Örneği

```
person = movie.director
print(person.name)
```

```
SELECT "Person".id AS "Person_id",
       "Person".name AS "Person_name"
FROM "Person"
WHERE "Person".id = %(param_1)s

{'param_1': 8}
```

53 / 58

Backref Örneği

```
for movie in person.directed:
    print(movie.title)
```

```
SELECT "Movie".id AS "Movie_id",
       "Movie".title AS "Movie_title",
       ...
       "Movie".directorid AS "Movie_directorid"
FROM "Movie"
WHERE %(param_1)s = "Movie".directorid

{'param_1': 8}
```

54 / 58

Dış Anahtar Örneği

- ▶ ikinci bir tablo kullanarak

```
casting_table = Table(
    'Casting', metadata,
    Column('movieid', Integer, ForeignKey('Movie.id'),
           primary_key=True),
    Column('actorid', Integer, ForeignKey('Person.id'),
           primary_key=True),
    Column('ord', Integer)
)
```

55 / 58

Dış Anahtar Örneği

```
mapper(Movie, movie_table,
       properties={
           'director':
               relationship(Person,
                           backref='directed'),
           'cast':
               relationship(Person,
                           backref='acted',
                           secondary=casting_table)
       })
```

56 / 58

Dış Anahtar Örneği

```
for movie in session.query(Movie):
    print('{}:'.format(movie.title))
    for person in movie.cast:
        print('  {}'.format(person.name))

for person in session.query(Person):
    print('{}:'.format(person.name))
    for movie in person.acted:
        print('  {}'.format(movie.title))
```

Kaynaklar

Yardımcı Kaynak

- SQLAlchemy Documentation:
<http://docs.sqlalchemy.org/>