Class4-DatabaseModule-Week12

(M:Manuel olarak, K:SQL komutlariyla, C:Python kodlariyla)

Asagidaki sorulardan K ve C ile cozulmesini istediklerimizin cozumlerini (komut veya kodlarini) ustte sorusu altta cozumu olacak sekilde bir dosyaya yapistirip gondermenizi istiyoruz.

1. **'pycoders' isimli bir server kurun. (M)**

Kuruldu.

2. **2- 'class4' database olusturun (M). Database silin (M). Ayni database yine olusturun (K)**

Kod:

```
create database class4
```

3. **https://www.postgresqltutorial.com/postgresql-sample-database/ adresine gidin ve ER modeli inceleyin. Tablolar arasindaki en az 5 iliskiyi yazin.(Hangi tablolar arasinda ne tur bir iliski var)**

Iliski 1: category ve film_category tablolari "category id" verisi ile birbirlerine iliskilendirilmistir. Burdaki iliski turu "one to many" dir. category tablosundaki primary key olan category_id verisi film_category tablosunda foreign key olarak iliskilidir.

Iliski 2: film tablosundaki primary key olan film_id verisi, film_category tablosunda foreign key olarak kullanilmistir. Burdaki iliski “one to one” dir.

Iliski 3: Languge tablosundaki primary key olan language_id verisi film tablosunda foreign key olarak kullanilmistir. Burdaki iliski turu de “one to many” dir.

Iliski 4: film tablosundaki primary key olan film_id verisi inventory tablosunda foreign key olarak kullanilmistir. Burdaki iliski turu de “one to one” dir.

Iliski 5: actor tablosundaki primary key olan actor_id verisi film tablosunda foreign key olarak kullanilmistir. Burdaki iliski turu de “one to many” dir.

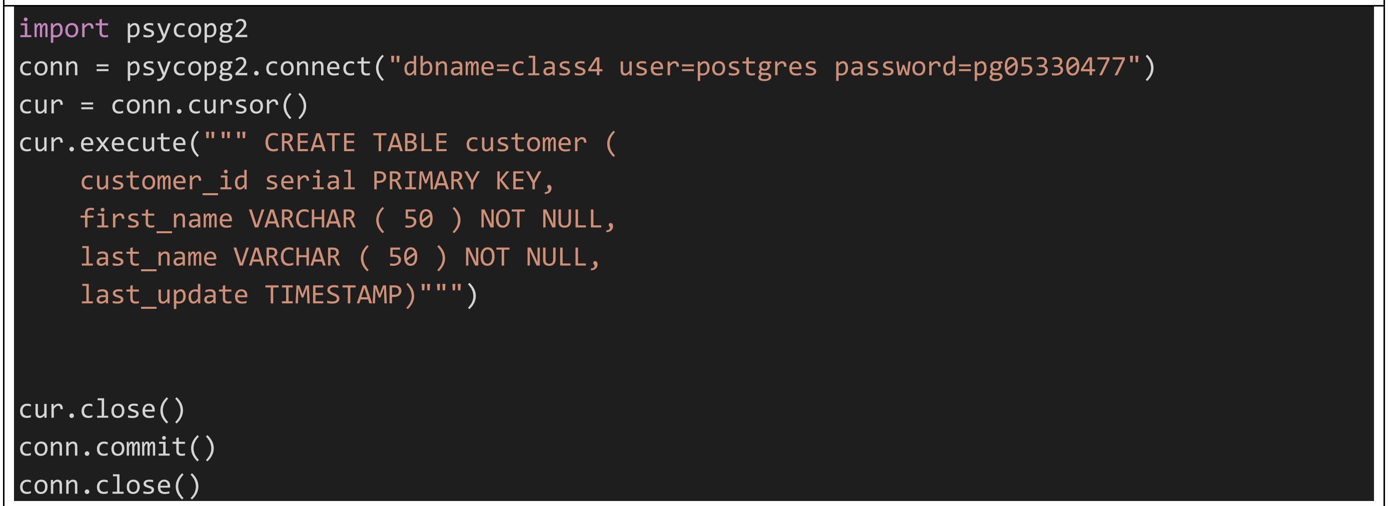
4. **ER modeldeki tablolardan 3 tanesini M olusturun.**

Olusturuldu.

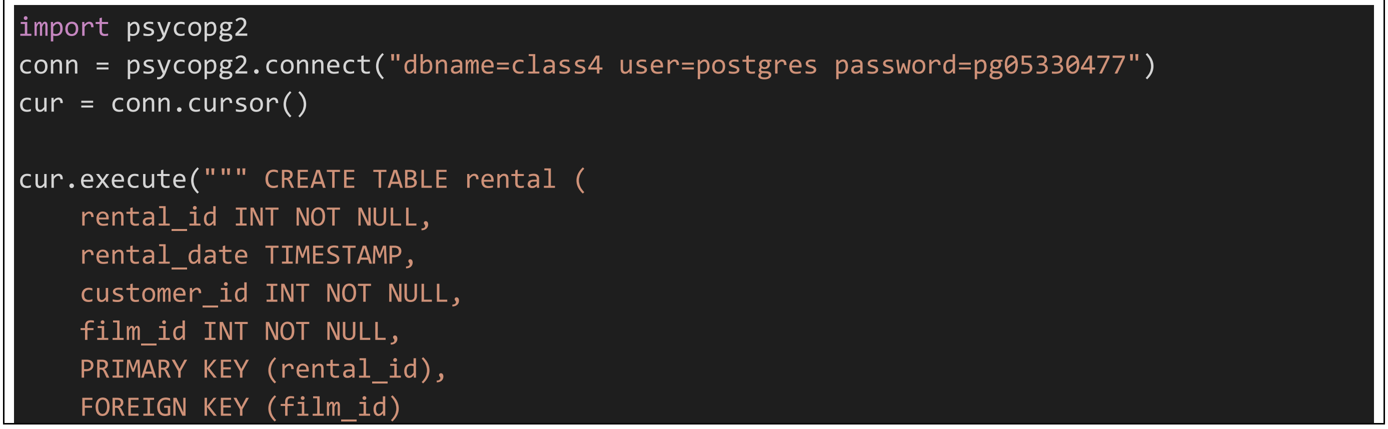
5. **ER modeldeki tablolardan 3 tanesini K olusturun.**

```
CREATE TABLE actor (

        actor_id serial PRIMARY KEY,

        first_name VARCHAR ( 50 ) NOT NULL,

        last_name VARCHAR ( 50 ) NOT NULL,

        last_login TIMESTAMP)



CREATE TABLE language (

        language_id serial PRIMARY KEY,
        name VARCHAR ( 20 ) UNIQUE NOT NULL,

        last_update TIMESTAMP)


CREATE TABLE film_actor (

 film_id INT NOT NULL,

 actor_id INT NOT NULL,

 last_update TIMESTAMP,

 PRIMARY KEY (film_id, actor_id),

 FOREIGN KEY (film_id)

   REFERENCES film (film_id),

 FOREIGN KEY (actor_id)

   REFERENCES actor (actor_id)

        )
```

6. **ER modeldeki tablolardan 3 tanesini C olusturun.**



```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()
cur.execute(""" CREATE TABLE customer (
    customer_id serial PRIMARY KEY,
    first_name VARCHAR ( 50 ) NOT NULL,
    last_name VARCHAR ( 50 ) NOT NULL,
    last_update TIMESTAMP)""")


cur.close()
conn.commit()
conn.close()
```



```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(""" CREATE TABLE rental (
    rental_id INT NOT NULL,
    rental_date TIMESTAMP,
    customer_id INT NOT NULL,
    film_id INT NOT NULL,
    PRIMARY KEY (rental_id),
    FOREIGN KEY (film_id)
```

```
        REFERENCES film (film_id),
    FOREIGN KEY (customer_id)
        REFERENCES customer (customer_id) )"""
)


cur.close()
conn.commit()
conn.close()
```

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

""" CREATE TABLE payment (
    payment_id INT NOT NULL,
    payment_date TIMESTAMP,
    rental_id INT NOT NULL,
    PRIMARY KEY (payment_id),
    FOREIGN KEY (rental_id)
    REFERENCES rental (rental_id))"""
)


cur.close()
conn.commit()
conn.close()
```

(4-5-6. sorulari cozerken toblolar arasindaki iliskileri gozardi edebilirsiniz. Tum kolonlari girmek zorunda degilsiniz, en az 2 kolon olmasi yeterli.)

Iliskileri de olusturmaya calistim

7. Olusturdugunuz 3 tabloya M ile 5 veri girisi yapin.

Yapildi.

8. Olusturdugunuz 3 tabloya K ile 5 veri girisi yapin.

--TABLO 1

insert into customer(customer_id, first_name, last_name, last_update)

VALUES

    ('1', 'Ali', 'Ates', '3-4-2021'),

```
    ('2', 'Ahmet', 'Toprak', '3-4-2021'),

    ('3', 'Aylin', 'Deniz', '3-4-2021'),

    ('4', 'Fatma', 'Simsek', '3-4-2021'),

    ('5', 'Hasan', 'Kara', '3-4-2021')
```

--TABLO 2

```
insert into rental(rental_id, rental_date, customer_id, film_id)

VALUES

    ('1', '1-1-2021', '1','1'),

    ('2', '1-1-2021', '2','2'),

    ('3', '1-1-2021', '3','3'),

    ('4', '1-1-2021', '4','4'),

    ('5', '1-1-2021', '5','5')
```

--TABLO 3

```
insert into payment(payment_id, payment_date, rental_id)

VALUES

    ('1', '1-1-2021', '1'),

    ('2', '1-1-2021', '2'),

    ('3', '1-1-2021', '3'),

    ('4', '1-1-2021', '4'),

    ('5', '1-1-2021', '5')
```

9. **Olusturdugunuz 3 tabloya C ile 5 veri girisi yapin.**

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

""" insert into language(language_id, name, last_update)
VALUES
    ('1', 'English', '1-1-2021'),
    ('2', 'Dutch', '1-1-2021'),
    ('3', 'Turkish', '1-1-2021'),
    ('4', 'Chinese', '1-1-2021'),
    ('5', 'Spanish', '1-1-2021')
"""
)
cur.close()
conn.commit()
conn.close()
```

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

""" insert into category(category_id, name, last_update)
VALUES
    ('1', 'Action', '1-1-2021'),
    ('2', 'Comedy', '1-1-2021'),
    ('3', 'Dram', '1-1-2021'),
    ('4', 'Horror', '1-1-2021'),
    ('5', 'Scifi', '1-1-2021')
"""
)
cur.close()
conn.commit()
conn.close()
```

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

""" insert into film_category(film_id, category_id, last_update)
VALUES
    ('1', '2', '1-1-2021'),
    ('2', '2', '1-1-2021'),
    ('3', '2', '1-1-2021'),
    ('4', '2', '1-1-2021'),
    ('5', '2', '1-1-2021')
"""
)
cur.close()
conn.commit()
conn.close()
```

10. **3 tablodaki birer veriyi M ile degistirin.**

Degistirildi.

11. 3 tablodaki birer veriyi K ile degistirin.

```
UPDATE language

SET last_update = '1-2-2021'

WHERE language_id = 2

RETURNING *;


UPDATE category

SET last_update = '1-2-2021'

WHERE category_id = 1

RETURNING *;


UPDATE customer

SET first_name = 'Mehmet'

WHERE customer_id = 1

RETURNING *;
```

12. 3 tablodaki birer veriyi C ile degistirin.

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

""" UPDATE rental
SET rental_date = '1-3-2021'
WHERE rental_id = 1
"""
)
cur.close()
conn.commit()
conn.close()
```

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(
""" UPDATE payment
SET payment_date = '1-3-2021'
WHERE payment_id = 1
"""
)
cur.close()
conn.commit()
conn.close()
```

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

""" UPDATE actor
SET last_login = '1-4-2021'
WHERE actor_id = 1
"""
)
cur.close()
conn.commit()
conn.close()
```

13. 3 tablonun son satirini M ile silin.

Silindi.

14. 3 tablonun son satirini K ile silin.

```
DELETE FROM rental

WHERE rental_id = (

  SELECT MAX (rental_id)

  FROM rental

);



DELETE FROM film_category

WHERE film_id = (

  SELECT MAX (film_id)

  FROM film_category

);
```

```
DELETE FROM film_actor

WHERE film_id = (

  SELECT MAX (film_id)

  FROM film_actor

);
```

15. **3 tablonun son satirini C ile silin.**

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

""" DELETE FROM film
WHERE film_id = (
   SELECT MAX (film_id)
   FROM film
);"""
)
cur.close()
conn.commit()
conn.close()
```

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

""" DELETE FROM language
WHERE language_id = (
   SELECT MAX (language_id)
   FROM language
);"""
)
cur.close()
conn.commit()
conn.close()
```

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()
```

```
cur.execute(

""" DELETE FROM customer
WHERE customer_id = (
   SELECT MAX (customer_id)
   FROM customer
);"""
)
cur.close()
conn.commit()
conn.close()
```

16. 1 tabloyu M ile silin.

Silindi.

17. 1 tabloyu K ile silin.

drop table film_category

18. 1 tabloyu C ile silin.

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

""" drop table film_actor"""
)
cur.close()
conn.commit()
conn.close()
```

19. Kalan tablolardan 1 tanesinin 2 veya 3 sutununu K ile baska bir tablo olarak olusturun.

CREATE TABLE yeni_tablo AS

SELECT (first_name), (last_name) FROM customer;

20. Kalan tablolardan 1 tanesinin 2 veya 3 sutununu C ile baska bir tablo olarak olusturun.

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

""" CREATE TABLE yeni_tablo2 AS
```

```
  SELECT (first_name), (last_name) FROM actor;"""
)
cur.close()
conn.commit()
conn.close()
```

21. Tablolardan 1 tanesini M ile truncate edin.

Edildi.

22. Tablolardan 1 tanesini K ile truncate edin.

truncate table yeni_tablo

23. Tablolardan 1 tanesini C ile truncate edin.

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

"""truncate table rental"""
)
cur.close()
conn.commit()
conn.close()
```

24. Truncate edilmis tablolari M ile silin.

Silindi.

25. 2 tabloyu K ile silin.

drop table language

drop table category

26. 2 tabloyu C ile silin.

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

""" drop table film"""
)
cur.close()
conn.commit()
conn.close()
```

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

""" drop table customer"""
)
cur.close()
conn.commit()
conn.close()
```

27. Elimizde veri olan 1 tablo kalmis olmasi lazim. Bu tabloyu csv olarak bilgisayariniza yukleyin.

Yuklendi.

28. Postgresql arayuzundeki son tabloyu da K ile silin.

drop table actor

29. Bilgisayarinizdaki csv yi arayuze import edin.

COPY actor(id, first_name, last_name, last_update)

FROM 'C:\Users\Moes\Desktop\2_Muhammet\PYCODERS\Week 11\actor.csv'

DELIMITER ','

CSV HEADER;

30. Import ettiginiz bu tabloyu C ile silin.

```
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=pg05330477")
cur = conn.cursor()

cur.execute(

""" drop table actor"""
)
cur.close()
conn.commit()
conn.close()
```

31. https://www.postgresqltutorial.com/postgresql-sample-database/ linkindeki ornek DB yi bilgisayariniza indirin ve arayuze yukleyin.

Eklendi.

32. **DB nizde 15 adet tablo olmasi lazim. Her tabloyu teker teker goruntuleyin ve kolon isimlerine bakarak, 5 tabloda hangi kolonun PK ve FK oldugunu yazin.**

Adress tablosunda adress_id PK, city_id FK

Actor tablosunda actor_id PK

Category tablosunda category_id PK

City tablosunda ciy_id PK, country_id FK

Country tablosunda country_id PK

33. **Sorgular? (Asagidaki sorularin cevaplarini ve bu cevabi bulurken kullandiginiz kodlari yazin)**

34. **Action filmlerinin ortalama suresi ne kadar?**

```
SELECT AVG(length)

FROM film where (description) like '%Action%';
```

35. **En cok staff olan store hangisidir?**

```
SELECT

        store_id,

        count (store_id)

FROM

        staff

GROUP BY

        store_id

ORDER BY

        count (store_id) DESC

limit 1
```

36. **'Gene Willis' adli actorun oynadigi filmlerin ratingi nedir?**

```
SELECT

        film_id,

        rating

FROM

        film
```

```
WHERE
    film_id IN (
        SELECT
            film_actor.film_id
        FROM
            film_actor
        WHERE
            actor_id = (select (actor_id) from actor
where (first_name)='Gene' and (last_name)='Willis')
    );
```

37. Aktif customer sayisi nedir? (584)

```
select count (*) from customer
    where active = 1
```

38. 'C' harfiyle baslayan filmler hangileridir?

```
select title from film
where title like 'C%'
```

39. 4$ den az odeme yapan musterilerin e-mail edresleri nedir?

```
SELECT
    email
FROM
    customer
WHERE
    customer_id IN (
        SELECT
            payment.customer_id
        FROM
            payment
        WHERE
            amount < 4
```

```
          );
```

40. **Moscow'da ikamet eden staff ve customer tablosu? (sadece isim/soyisim sutunu olsun)**

```
select first_name, last_name
from customer
where address_id = (select address_id from address where city_id = (select city_id from city where city = 'Moscow'))
union
select first_name, last_name
from staff
where address_id = (select address_id from address where city_id = (select city_id from city where city = 'Moscow'))
```

41. **En az kiralanan 5 film hangisidir?**

```
create table yeni as (SELECT film_id, inventory_id
FROM inventory
RIGHT JOIN rental USING (inventory_id))
SELECT
          film_id,
          count (inventory_id)
FROM
          yeni
GROUP BY
          film_id
ORDER BY
          count (film_id) DESC
limit 5
```

42. **2006 yilinda yayinlanan ingilizce filmler hangileridir?**

```
select count(*)
from film
where release_year = 2006 and language_id = (select language_id from language
where name='English')
```