2. CREATE DATABASE class4

WITH

OWNER = postgres

ENCODING = 'UTF8'

CONNECTION LIMIT = -1;

3. \* “category” tablosu ile “film\_category” tablosu arasında one to one ilişkisi var. (anahtar “category\_id”)

\* “film\_category” tablosu ile “film” tablosu arasında one to one ilişkisi var. (anahtar “film\_id”)

\* “film” tablosu ile “language” tablosu arasında one to many ilişkisi var. (anahtar “language\_id”)

\* “film” tablosu ile “invertory” tablosu arasında one to many ilişkisi var. (anahtar “invertory\_id”)

\* “invertory” tablosu ile “rental” tablosu arasında one to one ilişkisi var. (anahtar “rental\_id”)

5. \* CREATE TABLE public.category

(

category\_id smallint NOT NULL,

name character varying(20) NOT NULL,

PRIMARY KEY (category\_id)

);

ALTER TABLE public.category

OWNER to postgres;

* CREATE TABLE public.film\_category

(

film\_id smallint NOT NULL,

category\_id smallint NOT NULL,

PRIMARY KEY (film\_id, category\_id)

);

ALTER TABLE public.film\_category

OWNER to postgres;

* CREATE TABLE public.film

(

film\_id smallint NOT NULL,

title character varying(15) NOT NULL,

language\_id smallint NOT NULL,

PRIMARY KEY (film\_id, language\_id)

);

ALTER TABLE public.film

OWNER to postgres;

6. \* import psycopg2

conn = psycopg2.connect("dbname=class4 user=postgres password=12345")

cur = conn.cursor()

command = '''CREATE TABLE public.category

(

category\_id smallint NOT NULL,

name character varying(20) NOT NULL,

PRIMARY KEY (category\_id)

)

'''

cur.execute(command)

cur.close()

conn.commit()

conn.close()

\* import psycopg2

conn = psycopg2.connect("dbname=class4 user=postgres password=12345")

cur = conn.cursor()

command = '''CREATE TABLE public. film\_category

(

film\_id smallint NOT NULL,

category\_id smallint NOT NULL,

PRIMARY KEY (film\_id, category\_id)

)

'''

cur.execute(command)

cur.close()

conn.commit()

conn.close()

\* import psycopg2

conn = psycopg2.connect("dbname=class4 user=postgres password=12345")

cur = conn.cursor()

command = '''CREATE TABLE public. film

(

film\_id smallint NOT NULL,

title character varying(15) NOT NULL,

language\_id smallint NOT NULL,

PRIMARY KEY (film\_id, language\_id)

)

'''

cur.execute(command)

cur.close()

conn.commit()

conn.close()

8. \* insert into film (film\_id, title, language\_id) values (1975, 'hababam sınıfı', 100), (1996, ‘babam ve oğlum’, 100), (1996, 'eşkiya', 100), (1976, 'tosun paşa', 100), (2004, 'gora', 200);

\* insert into category (category\_id,name) values (10, 'komedi'), (20, 'dram');

\* insert into film\_category (film\_id,category\_id) values (2004, 10), (1975, 10),(1996, 10), (1997, 20), (1976, 20);

9. \* conn = psycopg2.connect("dbname=class4 user=postgres password=12345")  
cur = conn.cursor()  
cur.execute('INSERT INTO film VALUES(%s,%s,%s)', (1975, 'hababam sınıfı', 100))  
cur.execute('INSERT INTO film VALUES(%s,%s,%s)', (1996, ‘babam ve oğlum’, 100))  
cur.execute('INSERT INTO film VALUES(%s,%s,%s)', (1996, 'eşkiya', 100))  
cur.execute('INSERT INTO film VALUES(%s,%s,%s)', (1976, 'tosun paşa', 100))  
cur.execute('INSERT INTO film VALUES(%s,%s,%s)', (2004, 'gora', 200))  
cur.close()  
conn.commit()  
conn.close()

\* conn = psycopg2.connect("dbname=class4 user=postgres password=12345")  
cur = conn.cursor()  
cur.execute('INSERT INTO category VALUES(%s,%s)', (10, 'komedi'))  
cur.execute('INSERT INTO category VALUES(%s,%s)', ((20, 'dram'))  
cur.close()  
conn.commit()  
conn.close()

\* conn = psycopg2.connect("dbname=class4 user=postgres password=12345")  
cur = conn.cursor()  
cur.execute('INSERT INTO film\_category VALUES(%s,%s)', (2004, 10))  
cur.execute('INSERT INTO film\_category VALUES(%s,%s)', (1975, 10))

cur.execute('INSERT INTO film\_category VALUES(%s,%s)', (1996, 10))  
cur.execute('INSERT INTO film\_category VALUES(%s,%s)', (1997, 20))

cur.execute('INSERT INTO film\_category VALUES(%s,%s)', (1976,20))  
cur.close()  
conn.commit()  
conn.close()

11. \* update film\_category set category\_id=30 where category\_id=10;

\* update film set language\_id=200 where title='hababam sınıfı';

\* update category set category\_id=30 where name='komedi';

12. \* conn = psycopg2.connect("dbname=class4 user=postgres password=12345")  
cur = conn.cursor()  
cur.execute('UPDATE film\_category SET category\_id=%s WHERE category\_id=%s', (30,10))  
cur.close()  
conn.commit()  
conn.close()

\* conn = psycopg2.connect("dbname=class4 user=postgres password=12345")  
cur = conn.cursor()  
cur.execute('UPDATE film SET language\_id=%s WHERE title=%s', (200,’hababam sınıfı’))  
cur.close()  
conn.commit()  
conn.close()

\* conn = psycopg2.connect("dbname=class4 user=postgres password=12345")  
cur = conn.cursor()  
cur.execute('UPDATE category SET category\_id=%s WHERE name=%s', (30,’komedi’))  
cur.close()  
conn.commit()  
conn.close()

14. \* DELETE FROM film

WHERE category\_id = (SELECT title

FROM film

ORDER BY category\_id DESC

LIMIT 1);

\* DELETE FROM film\_category

WHERE category\_id = (SELECT category\_id

FROM film

ORDER BY category\_id DESC

LIMIT 1);

15. \* conn = psycopg2.connect("dbname=class4 user=postgres password=12345")  
cur = conn.cursor()  
command = '''DELETE FROM film  
WHERE category\_id = (SELECT title

FROM film

ORDER BY category\_id DESC

LIMIT 1)'''  
cur.execute(command)  
cur.close()  
conn.commit()  
conn.close()

\* conn = psycopg2.connect("dbname=class4 user=postgres password=12345")  
cur = conn.cursor()  
command = '''DELETE FROM film\_category  
WHERE category\_id = (SELECT category\_id

FROM film

ORDER BY category\_id DESC

LIMIT 1)'''  
cur.execute(command)  
cur.close()  
conn.commit()  
conn.close()

17. drop table category;

18. conn = psycopg2.connect("dbname=class4 user=postgres password=12345")

cur = conn.cursor()

command = '''DROP Table category

'''

cur.execute(command)

cur.close()

conn.commit()

conn.close()

19. CREATE TABLE category\_new

AS SELECT film\_id, title

FROM film;

20. conn = psycopg2.connect("dbname=class4 user=postgres password=12345")  
cur = conn.cursor()  
command = '''CREATE TABLE category\_new   
AS SELECT film\_id, title

FROM film;

'''  
cur.execute(command)  
cur.close()  
conn.commit()  
conn.close()

22. truncate table film\_category

23. conn = psycopg2.connect("dbname=class4 user=postgres password=12345")  
cur = conn.cursor()  
command = ''' TRUNCATE TABLE film\_category  
'''  
cur.execute(command)  
cur.close()  
conn.commit()  
conn.close()

25. drop table category\_new, film\_category;

26. conn = psycopg2.connect("dbname=class4 user=postgres password=12345")  
 cur = conn.cursor()  
command = ''' category\_new, film\_category;  
'''  
cur.execute(command)  
cur.close()  
conn.commit()  
conn.close()

27. COPY film TO 'C:\Users\10\film.csv' DELIMITER ',' CSV HEADER;

28. drop table film;

29. COPY film TO 'C:\Users\10\film.csv' DELIMITER ',' CSV HEADER;

30. conn = psycopg2.connect(**"dbname=class4 user=postgres password=12345"**)  
cur = conn.cursor()  
command = **'''** **DROP Table film\_category\_3;  
'''**cur.execute(command)  
cur.close()  
conn.commit()  
conn.close()

1. \* address” TABLE ==> address\_id: primary key city\_id: foreign key from “city” TABLE

\* “actor” TABLE ==> actor\_id: primary key

\* “category” TABLE ==> category\_id: primary key

\* “film” TABLE ==> film\_id: primary key language\_id: foreign key from “language”TABLE

\* “country” TABLE ==> country\_id: primary key

1. \* select avg(length) from film

\* where film\_id IN (select film\_id from film\_category

\* where category\_id= (select category\_id from category

\* where name='Action'));

34. \* SELECT count(\*) FROM customer WHERE store\_id=1;

\* SELECT count(\*) FROM customer WHERE store\_id=2;

\* SELECT store\_id,COUNT(\*) FROM customer GROUP BY store\_id ORDER BY COUNT(\*) DESC LIMIT 1;

35. SELECT rating FROM film WHERE film\_id IN (SELECT film\_id FROM film\_actor WHERE actor\_id=(SELECT actor\_id FROM actor WHERE first\_name='Gene' and last\_name='Willis'));

36. SELECT COUNT(\*) FROM customer WHERE active=1;

37. SELECT \* FROM film WHERE title LIKE 'C%';

38. SELECT email FROM customer WHERE customer\_id IN (SELECT customer\_id FROM payment WHERE amount<4);

39. 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' ));

40. SELECT film\_id,COUNT(\*) FROM inventory GROUP BY film\_id ORDER BY COUNT(\*) DESC LIMIT 1;

41. SELECT \* FROM film WHERE language\_id=(SELECT language\_id FROM language WHERE name='English');

**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)

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

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)

4- ER modeldeki tablolardan 3 tanesini M olusturun.

5- ER modeldeki tablolardan 3 tanesini K olusturun.

6- ER modeldeki tablolardan 3 tanesini C olusturun.

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

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

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

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

10- 3 tablodaki birer veriyi M ile degistirin.

11- 3 tablodaki birer veriyi K ile degistirin.

12- 3 tablodaki birer veriyi C ile degistirin.

13- 3 tablonun son satirini M ile silin.

14- 3 tablonun son satirini K ile silin.

15- 3 tablonun son satirini C ile silin.

16- 1 tabloyu M ile silin.

17- 1 tabloyu K ile silin.

18- 1 tabloyu C ile silin.

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

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

21- Tablolardan 1 tanesini M ile truncate edin.

22- Tablolardan 1 tanesini K ile truncate edin.

23- Tablolardan 1 tanesini C ile truncate edin.

24- Truncate edilmis tablolari M ile silin.

25- 2 tabloyu K ile silin.

26- 2 tabloyu C ile silin.

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

28- Postgresql arayuzundeki son tabloyu da K ile silin.

29- Bilgisayarinizdaki csv yi arayuze import edin.

30- Import ettiginiz bu tabloyu C ile silin.

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

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.

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

33- Action filmlerinin ortalama suresi ne kadar?

34- En cok staff olan store hangisidir?

35- 'Gene Willis' adli actorun oynadigi filmlerin ratingi nedir?

36- Aktif customer sayisi nedir?

37- 'C' harfiyle baslayan filmler hangileridir?

38- 4$ den az odeme yapan musterilerin e-mail edresleri nedir?

39- Moscow'da ikamet eden staff ve customer tablosu? (sadece isim/soyisim sutunu olsun)

40- En az kiralanan 5 film hangisidir?

41- 2006 yilinda yayinlanan ingilizce filmler hangileridir?