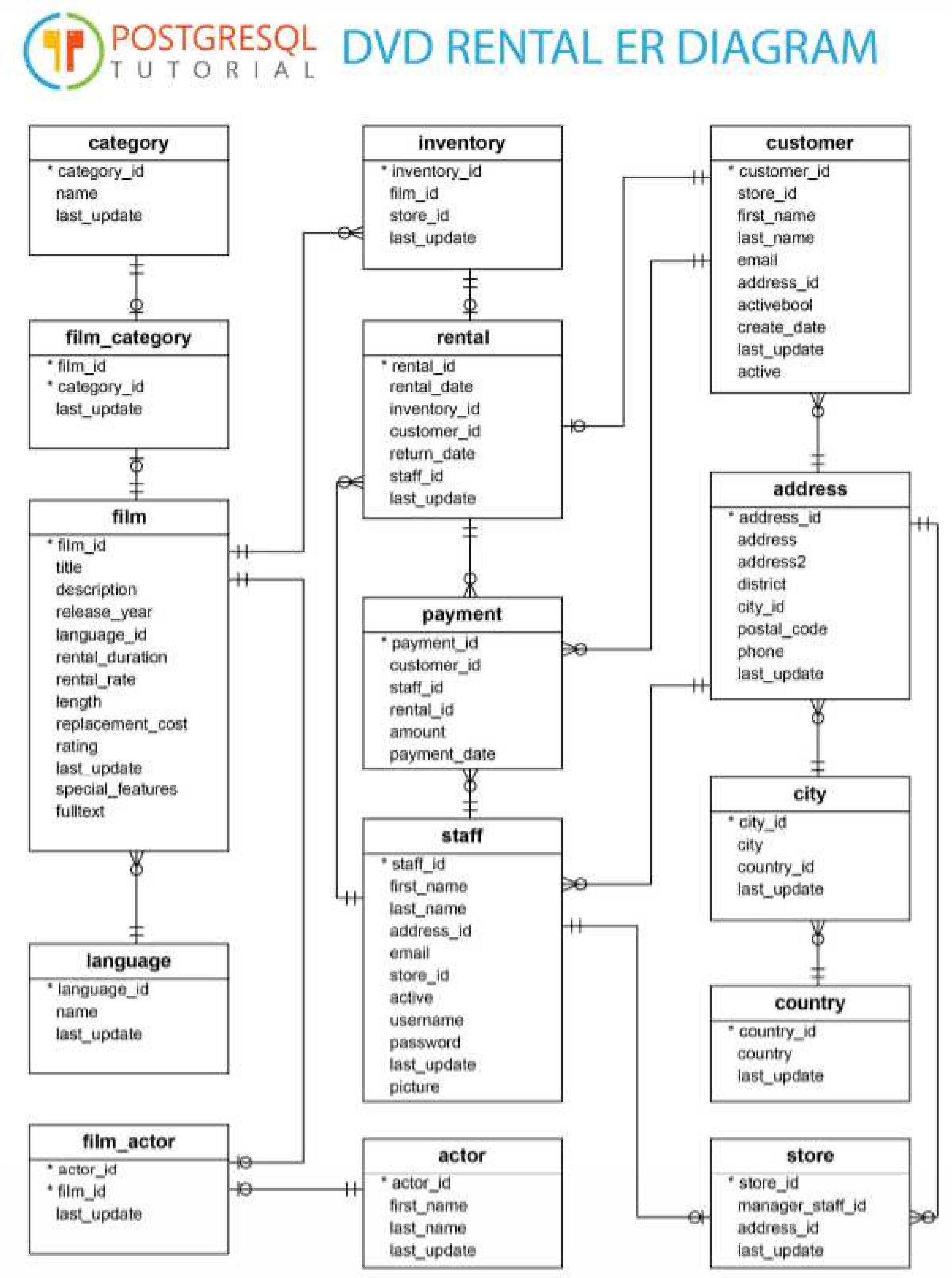
Tasks

1- 'pycoders' isimli bir server kurun. (Manuel)

2- 'class4' database olusturun (Manuel). Database silin (Manuel). Ayni database yine olusturun (SQL)

```
CREATE DATABASE class4
WITH
OWNER = postgres
ENCODING = 'UTF8'
CONNECTION LIMIT = -1;
```

3- [https://www.postgresqltutorial.com/postgresql-sample-database/ (https://www.postgresqltutorial.com/postgresql-sample-database/)](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)



```
a- store ve staff arasinda: (0 or 1) to 1
b- country ve city arasinda: 1 to (0 or many)
c- language ve film arasinda: 1 to (0 or many)
d- adress ve store arasinda: 1 to (0 or many)
e- staff ve store arasinda 1 to (0 or 1) iliski vardir.
```

4- ER modeldeki tablolardan 3 tanesini Manuel olusturun.

5- ER modeldeki tablolardan 3 tanesini SQL olusturun.

```
-- Table: public.actor
CREATE TABLE public.actor
(
    actor_id INTEGER PRIMARY KEY,
    first_name VARCHAR(255),
    last_name VARCHAR(255),
    last_update DATE,
);

-- Table: public.stuff
CREATE TABLE public.staff
(
    staff_id INTEGER PRIMARY KEY,
    first_name VARCHAR(255),
    last_name VARCHAR(255),
    adress_id INTEGER,
    store_id INTEGER,
);

-- Table: public.customer
CREATE TABLE public.customer
(
    customer_id INTEGER PRIMARY KEY,
    address_id INTEGER,
    first_name VARCHAR(255),
    last_name VARCHAR(255),
);
```

6- ER modeldeki tablolardan 3 tanesini Python olusturun.

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

In [6]:

```
import psycopg2

conn = psycopg2.connect("dbname=class4 user=postgres password=1234")
cur = conn.cursor()
commands = (
        """
        CREATE TABLE category (
            category_id INTEGER PRIMARY KEY,
            name VARCHAR(255) NOT NULL,
            last_update DATE NOT NULL
        )
        """,
        """ CREATE TABLE inventory (
            inventory_id INTEGER PRIMARY KEY,
            film_id INTEGER NOT NULL,
            store_id INTEGER NOT NULL,
            last_update DATE NOT NULL
        )
        """,
        """
        CREATE TABLE language (
                language_id INTEGER PRIMARY KEY,
                name VARCHAR(255) NOT NULL,
                last_update DATE NOT NULL
        )
        """)
#create a table
for command in commands:
    cur.execute(command)
# close communication with the PostgreSQL database server
cur.close()
# commit the changes
conn.commit()
# close the connection
conn.close()
```

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

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

```

INSERT INTO category(category_id, name, last_update) VALUES (6, 'Adventure', '2021-01-
04');
INSERT INTO category(category_id, name, last_update) VALUES (7, 'Horror', '2021-05-04');
INSERT INTO category(category_id, name, last_update) VALUES (8, 'Musical', '2021-01-04');
INSERT INTO category(category_id, name, last_update) VALUES (9, 'Horror', '2021-01-04');
INSERT INTO category(category_id, name, last_update) VALUES (10, 'Action', '2021-01-04');

INSERT INTO actor(actor_id, first_name, last_name, last_update) VALUES (6, 'Eva','Green',
'2021-01-04');
INSERT INTO actor(actor_id, first_name, last_name, last_update) VALUES (7, 'Monica',
'Belluci','2021-05-04');
INSERT INTO actor(actor_id, first_name, last_name, last_update) VALUES (8, 'Mark',
'Wahlberg','2021-01-04');
```

```
INSERT INTO actor(actor_id, first_name, last_name, last_update) VALUES (9,
'Catherine','Zeta-Jones','2021-01-04');
INSERT INTO actor(actor_id, first_name, last_name, last_update) VALUES (10,
'Edward','Norton', '2021-01-04');

INSERT INTO country(country_id, country, last_update) VALUES (1, 'USA', '2021-06-04');
INSERT INTO country(country_id, country, last_update) VALUES (2, 'England', '2021-01-04');
INSERT INTO country(country_id, country, last_update) VALUES (3, 'Netherland', '2021-01-
04');
INSERT INTO country(country_id, country, last_update) VALUES (4, 'Germany', '2021-05-04');
INSERT INTO country(country_id, country, last_update) VALUES (5, 'Turkey', '2021-07-04');
```

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

In [21]:

```python
import psycopg2
conn = psycopg2.connect("dbname=class4 user=postgres password=1234")
cur = conn.cursor()

commands = ('''INSERT INTO country VALUES (6, 'Spain', '03/03/2020')''',
'''INSERT INTO country VALUES (7, 'China', '03/04/2021')''',
'''INSERT INTO country VALUES (8, 'Japan', '04/01/2021')''',
'''INSERT INTO country VALUES (9, 'Italy', '04/04/2021')''',
'''INSERT INTO country VALUES (10, 'France', '04/07/2021')''')

for command in commands:
    cur.execute(command)

commands = ('''INSERT INTO city VALUES (1, 'Istanbul', '03/03/2020')''',
'''INSERT INTO city VALUES (2, 'Amsterdam', '03/04/2021')''',
'''INSERT INTO city VALUES (3, 'New York', '04/01/2021')''',
'''INSERT INTO city VALUES (4, 'Paris', '04/04/2021')''',
'''INSERT INTO city VALUES (5, 'Berlin', '04/07/2021')''')

for command in commands:
    cur.execute(command)

commands = ('''INSERT INTO customer VALUES (1, 3, 'Fatma', 'Zehra')''',
'''INSERT INTO customer VALUES (2, 5, 'Ali', 'Murteza')''',
'''INSERT INTO customer VALUES (3, 1, 'Muhammed', 'Emin')''',
'''INSERT INTO customer VALUES (4, 4, 'Hasan', 'Serif')''',
'''INSERT INTO customer VALUES (5, 2, 'Huseyin', 'Seyyid')''')

for command in commands:
    cur.execute(command)

cur.close()
# commit the changes
conn.commit()
# close the connection
conn.close()
```

10- 3 tablodaki birer veriyi Manuel ile degistirin.

11- 3 tablodaki birer veriyi SQL ile degistirin.

```
update city set city='Ordu' where city_id=3;
update actor set last_update='04/07/2021' where actor_id=10;
update category set last_update='04/04/2021' where category_id=1;
```

12- 3 tablodaki birer veriyi Python ile degistirin.

In [25]:

```
conn=psycopg2.connect("dbname=class4 user=postgres password=1234")
cur=conn.cursor()

commands = ('''UPDATE city SET city='Utrecht' WHERE city_id=1;''',
'''UPDATE actor SET last_update='01/01/2021' WHERE actor_id=1;''',
'''UPDATE category SET last_update='09/09/2019' WHERE category_id=1;''')

for command in commands:
    cur.execute(command)

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

13- 3 tablonun son satirini Manuel ile silin.

14- 3 tablonun son satirini SQL ile silin.

```
DELETE FROM city WHERE city_id=2;
DELETE FROM actor WHERE actor_id=10;
DELETE FROM category WHERE category_id=4;
```

15- 3 tablonun son satirini Python ile silin.

In [ ]:

```
conn=psycopg2.connect("dbname=class4 user=postgres password=1234")
cur=conn.cursor()

commands = ('''DELETE FROM city WHERE city_id=1;''',
'''DELETE FROM actor WHERE actor_id=2;''',
'''DELETE FROM category WHERE category_id=5;''')

for command in commands:
    cur.execute(command)

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

16- 1 tabloyu Manuel ile silin.

17- 1 tabloyu SQL ile silin.

```
DROP TABLE store;
```

18- 1 tabloyu Python ile silin.

In [27]:

```
conn=psycopg2.connect("dbname=class4 user=postgres password=1234")
cur=conn.cursor()
cur.execute("DROP TABLE inventory;")
cur.close()
conn.commit()
conn.close()
```

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

```
CREATE TABLE actor_full_names AS
SELECT first_name, last_name
FROM actor;
```

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

In [28]:

```
conn=psycopg2.connect("dbname=class4 user=postgres password=1234")
cur=conn.cursor()
cur.execute("CREATE TABLE customer_full_names AS SELECT first_name, last_name FROM customer
cur.close()
conn.commit()
conn.close()
```

21- Tablolardan 1 tanesini Manuel ile truncate edin.

22- Tablolardan 1 tanesini SQL ile truncate edin.

```
TRUNCATE language
```

23- Tablolardan 1 tanesini Python ile truncate edin.

In [29]:

```
conn=psycopg2.connect("dbname=class4 user=postgres password=1234")
cur=conn.cursor()
cur.execute("TRUNCATE customer_full_names;")
cur.close()
conn.commit()
conn.close()
```

24- Truncate edilmis tablolari Manuel ile silin.

25- 2 tabloyu SQL ile silin.

```
DROP TABLE actor_full_names, city;
```

26- 2 tabloyu Python ile silin.

In [30]:

```
conn=psycopg2.connect("dbname=class4 user=postgres password=1234")
cur=conn.cursor()
cur.execute("DROP TABLE customer, category;")
cur.close()
conn.commit()
conn.close()
```

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

28- Postgresql arayuzundeki son tabloyu da SQL ile silin.

```
DROP TABLE actor;
```

29- Bilgisayarinizdaki csv yi arayuze import edin.

30- Import ettiginiz bu tabloyu Python ile silin.

In [ ]:

```
conn=psycopg2.connect("dbname=class4 user=postgres password=1234")
cur=conn.cursor()
cur.execute("DROP TABLE sctor;")
cur.close()
conn.commit()
conn.close()
```

31- https://www.postgresqltutorial.com/postgresql-sample-database/ (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.

```

-'actor' table primarykey: actor_id
-'adress' table primarykey: adress_pkey foreignkey: fk_address_city
-'category' table primarykey:category_pkey foreignkey:
-'city' table primarykey: city_pkey  foreignkey: fk_city
-'country' table primarykey: country_pkey   foreignkey:
-'customer' table primarykey: customer_pkey  foreignkey: customer_address_id_fkey
-'film' table primarykey: film_pkey   foreignkey: film_language_id_fkey
-'film_actor' table primarykey: film_actor_pkey  foreignkey: film_actor_actor_id_fkey,
film_actor_film_id_fkey
```

```
-'film_category' table primarykey: film_category_pkey  foreignkey:
film_category_category_id_fkey, film_category_film_id_fkey
-'inventory' table primarykey: inventory_pkey  foreignkey: inventory_film_id_fkey
-'language' table primarykey: language_pkey   foreignkey:
-'payment' table primarykey: payment_pkey  foreignkey: payment_customer_id_fkey,
payment_rental_id_fkey, payment_staff_id_fkey
-'' table primarykey: rental_pkey  foreignkey: rental_customer_id_fkey,
rental_inventory_id_fkey, rental_staff_id_key
-'rental' table primarykey:   foreignkey:
-'staff' table primarykey: staff_pkey  foreignkey: staff_address_id_fkey
-'store' table primarykey: store_pkey  foreignkey: store_address_id_fkey,
store_manager_staff_id_fkey
```

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

33- Action filmlerinin ortalama suresi ne kadar?

```
select avg(length) as average_lenght_of_action
from film_category fc
inner join category c
on c.category_id=fc.category_id
inner join film f
on f.film_id=fc.film_id
where name='Action'
```

34- En cok staff olan store hangisidir?

```
select store.store_id,count(staff_id) as max_staff_num
from staff
inner join store
on staff.store_id=store.store_id
group by store.store_id
order by max_staff_num desc
limit 1
```

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

```
select a.first_name, a.last_name, f.rating
from actor as a
inner join film_actor as fa
on a.actor_id=fa.actor_id
inner join film as f
on f.film_id=fa.film_id
where a.first_name='Gene' and a.last_name='Willis';
```

36- Aktif customer sayisi nedir?

```
select sum(active) as active_customer_num
from customer;
```

37- 'C' harfiyle baslayan filmler hangileridir?

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

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

```
select email
from customer as c
inner join payment as p
on p.customer_id=c.customer_id
where amount < 4;
```

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

```
select first_name, last_name
from address as a
inner join customer as c
on c.address_id=a.address_id
where district like 'Mos%'

union

select first_name, last_name
from address as a
inner join staff as s
on s.address_id=a.address_id
where district like 'Mos%'
```

40- En az kiralanan 5 film hangisidir?

```
select title, count(inventory_id) as num_of_rental
from (select i.inventory_id, i.film_id, r.rental_id
      from inventory as i
      right join rental  as r
      on i.inventory_id=r.inventory_id ) as x
inner join film as f
on f.film_id=x.film_id
group by f.film_id
order by count(f.film_id) asc
limit 5;
```

41- 2006 yilinda yayinlanan ingilizce filmler hangileridir?

```
select title
from film as f
inner join language as l
on l.language_id=f.language_id
where l.name='English' and f.release_year=2006;
```