Exercise 1

Query 1:

```
SELECT * FROM country
        ORDER BY country_id OFFSET 11 LIMIT 6;
```

σ country_id >=12, country_id <=17 (country)

Query 2:

```
SELECT city.city, address.address FROM address
        JOIN city ON address.city_id = city.city_id
        WHERE city LIKE 'A%';
```

π city.city, address.address σ (city ℑ LIKE ‘A%’ (city)) (address ⋈ address.city_id = city.city_id city)

Query 3:

```
SELECT customer.first_name, customer.last_name, city.city FROM customer
        JOIN address ON address.address_id = customer.address_id
        JOIN city ON city.city_id = address.city_id;
```

π customer.first_name, customer.last_name, city.city (customer ⋈ address.address_id = customer.address_id address ⋈ city.city_id = address.city_id city)

Query 4:

```
SELECT first_name, last_name, payment.amount FROM customer
        JOIN payment ON customer.customer_id = payment.customer_id
        WHERE payment.amount > 11;
```

π first_name, last_name, payment.amount σ payment,amount > 11 (customer ⋈ customer.customer_id = payment.customer_id payment)

Query 5:

```
SELECT customer.first_name, COUNT(*) FROM customer
        GROUP BY customer.first_name
        HAVING COUNT(*) > 1;
```

π customer.first_name σ (customer.first_name ℑ COUNT (customer) > 1) (customer)

Exercise 2

View where are films have length more than 150:

```
CREATE VIEW public.movies_with_lenght_more_than_150 AS
SELECT film_id, title, length FROM film
        WHERE length>150;
```

```
SELECT title, length
FROM film
WHERE film.length > 150;
```

Trigger:

```
CREATE FUNCTION trigger_function()
  RETURNS TRIGGER
  LANGUAGE PLPGSQL
AS $$
BEGIN
  NEW.amount = NEW.amount * 1.2;
  RETURN NEW;
END;
$$
----------------------------------------------------------------------------------------------------
CREATE TRIGGER Before_Update_payment_amount
BEFORE UPDATE ON payment
FOR EACH ROW
EXECUTE PROCEDURE trigger_function();
```