Lab I SQL Self and cross join

```
-- In this lab, you will be using the [Sakila](https://dev.mysql.com/doc/sakila/en/)
database of movie rentals.


use sakila;

select
a.actor_id, a.film_id, b.actor_id,
dense_rank() over(partition by a.actor_id order by a.film_id) as Ranks
from sakila.film_actor as a
join sakila.film_actor as b
on a.film_id = b.film_id
group by b.actor_id
order by Ranks;

-- 2. Get all pairs of customers that have rented the same film more than 3 times.
-- solution from brett
select  in2.film_id as FilmID, a1.customer_id as ID1, a2.customer_id as ID2 ,
count(*) as Count
from sakila.customer a1

inner join rental re1
on re1.customer_id = a1.customer_id

inner join inventory in1
on re1.inventory_id = in1.inventory_id

inner join film fa1
on in1.film_id=fa1.film_id

inner join inventory in2
on fa1.film_id = in2.film_id

inner join rental re2
on re2.inventory_id=in2.inventory_id

inner join customer a2
on re2.customer_id=a2.customer_id


where a1.customer_id <> a2.customer_id and a2.customer_id>a1.customer_id
group by a1.customer_id, a2.customer_id

having count(*)>3
order by Count desc;
---------- own approach

drop table if exists film_join1;
```

```
create temporary table film_join1
select r.rental_id, r.inventory_id, r.customer_id, i.film_id
from sakila.rental as r
join sakila.inventory as i
on r.inventory_id = i.inventory_id;

drop table if exists film_join2;
create temporary table film_join2
select r.rental_id, r.inventory_id, r.customer_id, i.film_id
from sakila.rental as r
join sakila.inventory as i
on r.inventory_id = i.inventory_id;

select * from film_join;

select * from film_join1;
select
f1.rental_id, f1.customer_id, f1.film_id, f2.film_id
from film_join1 as f1
join film_join2 as f2
on f1.film_id <> f2.film_id;


-- 3. Get all possible pairs of actors and films

select *
from sakila.actor
cross join
sakila.film
limit 100;
```