```
--4-2
SELECT CONCAT (c.first_name, ' ', c.last_name) as full_name , COUNT(c.customer_id)
AS rental_count
FROM public.customer AS c
JOIN public.rental AS r
ON c.customer_id = r.customer_id
GROUP BY full_name
ORDER BY rental_count DESC;
--limit(1); uncomment this if you want just the top 1 customer


--4-3
SELECT c.customer_id, CONCAT (c.first_name, ' ', c.last_name) as full_name,
f.title, COUNT(i.inventory_id) as film_total
FROM customer AS c
JOIN rental AS r
ON c.customer_id = r.customer_id
JOIN inventory AS i
ON r.inventory_id = i.inventory_id
JOIN film AS f
ON i.film_id = f.film_id
GROUP BY c.customer_id, c.first_name, c.last_name, f.title
HAVING COUNT(f.title) >=1 AND c.customer_id = 148
ORDER BY film_total DESC;
--limit(1); this would be useful if customer had rented a movie more then once
--customer never rented the same movie twice

--4-4
SELECT c.customer_id, CONCAT (c.first_name, ' ', c.last_name) AS full_name,
cat.name AS favorite_category, COUNT(*) as total_films
FROM customer AS c
JOIN rental AS r
ON r.customer_id = c.customer_id
JOIN inventory AS i
ON i.inventory_id = r.inventory_id
JOIN film AS f
ON f.film_id = i.film_id
JOIN film_category AS fcat
ON fcat.film_id = f.film_id
JOIN category as cat
ON fcat.category_id = cat.category_id
WHERE c.last_name = 'Hunt'
GROUP BY c.customer_id, full_name, cat.name
ORDER BY total_films DESC
limit(1)

--4-5
CREATE OR REPLACE FUNCTION public.delete_inactive_customer()
  RETURNS trigger
AS
$BODY$
BEGIN
      DELETE FROM customer.active WHERE active = 0;
    RETURN NULL;
END;
$BODY$
LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER delete_inactive_customer
```

```
AFTER INSERT ON public.customer
FOR EACH ROW
EXECUTE FUNCTION public.delete_inactive_customer();
```