

Question 1: What is the most popular movie genre by the total number of rentals?

Ans 1:

```
select
  name as genre,
  count(distinct f.film_id) as movie_cnt
  from
    cineflix.category c JOIN cineflix.film_category fc ON c.category_id =
    fc.category_id
                    JOIN cineflix.film f ON f.film_id = fc.film_id
                    JOIN cineflix.inventory i ON i.film_id = f.film_id
                    JOIN cineflix.rental r ON r.inventory_id = i.inventory_id
  group by name
  order by movie_cnt desc
  limit 1;
```

Question 2: Get a list of all customers who have rented more films than the average number of rentals.

Ans 2:

```
select t1.customer_id from
(
  select
    customer_id,
    count(rental_id) as rental_cnt
  from cineflix.rental
  group by customer_id
) t1
JOIN
(
  select round(avg(cnt),2) as avg_rent from
  (
    select customer_id,count(rental_id) as cnt
    from cineflix.rental
    group by customer_id
  )
)t2
```

```
ON t1.rental_cnt > t2.avg_rent;
```

#subquery

```
select
customer_id,
count(rental_id) as rental_cnt
from cineflix.rental
group by customer_id
having rental_cnt >
(
select round(avg(cnt),2) as avg_rent from
(
select customer_id,count(rental_id) as cnt
from cineflix.rental
group by customer_id
)
);
```

Question 3: List all customers and the total amount they have spent on rentals, including customers who have never rented a film.

```
select
c.customer_id,
ifnull(ROUND(sum(amount),2),0.0) as total_amount
from cineflix.customer c LEFT JOIN
cineflix.payment p
ON c.customer_id = p.customer_id
group by c.customer_id
order by 2 desc;
```

Question 4: List all film categories along with the number of films in each category, including categories with no films.

```
select
c.category_id,
name as category_name,
count(f.film_id) as num_of_films
```

```

from cineflix.category c
LEFT JOIN
cineflix.film_category fc ON fc.category_id = c.category_id
LEFT JOIN
cineflix.film f ON fc.film_id = f.film_id
group by category_id,name
order by num_of_films desc;

select ca.category_id, ca.name, COUNT(fc.film_id) as NO_OF_Films from
cineflix.category ca
LEFT JOIN cineflix.film_category fc USING (category_id)
group by ca.name, ca.category_id
order by NO_OF_Films desc;

```

Question 5: List all films with rental rates higher than the average rental rate.

```

select
f.film_id,
f.rental_rate,
tbl.avg_rental
from (
select
avg(rental_rate) as avg_rental
from cineflex.film) tbl
join cineflex.film f
on tbl.avg_rental < f.rental_rate;

```