```sql
-----------------------------------------
--AVG Function
-----------------------------------------

SELECT
    AVG(length) as avg_length,
    AVG(rental_duration) as avg_rental_duration,
    AVG(replacement_cost) as avg_replacement_cost
FROM film;

SELECT
    AVG(length) as avg1,
    AVG(DISTINCT length) as avg2
FROM film;

-----------------------------------------
--SUM Function
-----------------------------------------

SELECT
    SUM(length) as sum_length,
    SUM(rental_duration) as sum_rental_duration,
    SUM(replacement_cost) as sum_replacement_cost
FROM film;

SELECT
    SUM(salary) as sum_manager_sal,
    ROUND(AVG(salary),2) as avg_manager_sal
FROM employees
WHERE job_id in (2,7,10,14,15,19) -manager positios;


-----------------------------------------
--COUNT Function
-----------------------------------------

SELECT COUNT(*) FROM actor;

SELECT COUNT(*) FROM film
WHERE rating = 'G';

SELECT COUNT(*) FROM payment
WHERE customer_id = 341;

SELECT COUNT(phone_number) FROM employees;

SELECT COUNT(discount) FROM cars;

SELECT COUNT(distinct rating) FROM film;

SELECT COUNT(distinct job_id) FROM employees;
```

```sql
-----------------------------------------
--MIN-MAX Functions
-----------------------------------------

SELECT
    MIN(salary) min_salary,
    MAX(salary) max_salary,
    MIN(hire_date) min_hire_date,
    MAX(hire_date) max_hire_date
FROM employees;

SELECT
    MIN(length) min_length,
    MAX(length) max_length,
    MAX(replacement_cost) max_replacement_cost,
    MAX(rental_duration) - MIN(rental_duration) dif_rental_duration
FROM film;

-----------------------------------------
--GROUP BY Clause
-----------------------------------------

SELECT DISTINCT rating
FROM film
ORDER BY rating;

SELECT rating
FROM film
GROUP BY rating
ORDER BY rating;

SELECT
    customer_id,
    SUM (amount)
FROM payment
GROUP BY customer_id;

SELECT
    rating,
    SUM(length) sum_length,
    SUM(rental_duration) sum_rental_duration,
    SUM(replacement_cost) sum_replacement_cost
FROM film
GROUP BY rating
ORDER BY rating;

SELECT
    job_id,
    COUNT(*) numberof_emp,
    MIN(salary) min_salary,
    MAX(salary) max_salary
FROM employees
GROUP BY job_id
ORDER BY 1;
```

```sql
SELECT
    rating, special_features,
    COUNT(*) numberof_films
FROM film
GROUP BY rating, special_features
ORDER BY rating, special_features;

SELECT
    department_id, manager_id,
    COUNT(*) numberof_emp
FROM employees
GROUP BY department_id, manager_id
ORDER BY 1, 2;


-----------------------------------------
--Having Clause
-----------------------------------------

SELECT
    customer_id,
    SUM(amount) sum_amount
FROM payment
GROUP BY customer_id
HAVING SUM(amount) > 150

SELECT
    department_id,
    COUNT(*) numberof_emps
FROM employees
GROUP BY department_id
HAVING COUNT(*) > 5
```

```sql
-------------------------------------------------------
--EXERCISE ANSWERS
-------------------------------------------------------

-----------------------------------------
--AVG Function
-----------------------------------------

SELECT
      ROUND(AVG(length), 2) as avg_length,
      ROUND(AVG(rental_duration), 2) as avg_rental_duration,
      ROUND(AVG(replacement_cost), 2) as avg_replacement_cost
FROM film;

-----------------------------------------
--SUM Function
-----------------------------------------

SELECT SUM(return_date - rental_date)
FROM rental;

SELECT
    return_date, rental_date,
    return_date::date - rental_date::date
FROM rental;

-----------------------------------------
--COUNT Function
-----------------------------------------

select
    count(*) as number_of_rows,
    count(postal_code) as non_null_postal_code,
    count(distinct district) different_districts
from address;

-----------------------------------------
--MIN-MAX Functions
-----------------------------------------

SELECT
    MIN(length(CONCAT(first_name, last_name))) min_numberof_letters,
    MAX(length(CONCAT(first_name, last_name))) max_numberof_letters,
    ROUND(AVG(length(CONCAT(first_name, last_name))),2) avg_numberof_letters
FROM employees;
```

```sql
-----------------------------------------
--GROUP BY Clause
-----------------------------------------

--Exercise-1

select customer_id,
    count(*) as rental_count,
    min(rental_date)::date as first_rental_date,
    max(rental_date)::date as first_rental_date
from rental
group by customer_id

--Exercise-2

SELECT
    SUM(CASE rating WHEN 'NC-17' THEN 1 ELSE 0 end) numberof_NC17,
    SUM(CASE rating WHEN 'PG' THEN 1 ELSE 0 end) numberof_PG,
    SUM(CASE rating WHEN 'G' THEN 1 ELSE 0 end) numberof_G,
    SUM(CASE rating WHEN 'PG-13' THEN 1 ELSE 0 end) numberof_PG13,
    SUM(CASE rating WHEN 'R' THEN 1 ELSE 0 end) numberof_R
FROM film;


-----------------------------------------
--Having Clause
-----------------------------------------

--Exercise-1

SELECT customer_id, COUNT (*)
FROM rental
GROUP BY customer_id
HAVING COUNT (*) > 35;

--Exercise-2

SELECT department_id,
    COUNT(*) numberof_emps,
    CASE
        WHEN COUNT(*) <= 3 THEN 'Small Room'
        WHEN COUNT(*) BETWEEN 4 AND 6 THEN 'Middle Room'
        WHEN COUNT(*) > 6 THEN 'Big Room'
    END AS room_type
FROM employees
GROUP BY department_id
ORDER BY 2
```