The "HAVING" clause constraints are written the same way as the "WHERE" clause constraints, and are applied to the grouped rows. With our examples, this might not seem like a particularly useful construct, but if you imagine data with millions of rows with different properties, being able to apply additional constraints is often necessary to quickly make sense of the data.

Did you know?

If you aren't using the `GROUP BY` clause, a simple `WHERE` clause will suffice.

Exercise

For this exercise, you are going to dive deeper into **Employee** data at the film studio. Think about the different clauses you want to apply for each task.

Table: Employees

Role	COUNT(*)
Artist	5
Engineer	5
Manager	3

```
SELECT role, COUNT(*)
FROM employees
GROUP BY role;
```

RESET

Exercise 11 — Tasks

1. Find the number of Artists in the studio (without a **HAVING** clause) ✓
2. Find the number of Employees of each role in the studio ✓
3. Find the total number of years employed by all Engineers

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next – SQL Lesson 12: Order of execution of a Query
Previous – SQL Lesson 10: Queries with aggregates (Pt. 1)

The "HAVING" clause constraints are written the same way as the "WHERE" clause constraints, and are applied to the grouped rows. With our examples, this might not seem like a particularly useful construct, but if you imagine data with millions of rows with different properties, being able to apply additional constraints is often necessary to quickly make sense of the data.

Did you know?

If you aren't using the `GROUP BY` clause, a simple `WHERE` clause will suffice.

Exercise

For this exercise, you are going to dive deeper into **Employee** data at the film studio. Think about the different clauses you want to apply for each task.

Table: Employees

Role	COUNT(*)
Artist	5
Engineer	5
Manager	3

```
SELECT role, COUNT(*)
FROM employees
GROUP BY role;
```

RESET

Exercise 11 — Tasks

1. Find the number of Artists in the studio (without a **HAVING** clause) ✓
2. Find the number of Employees of each role in the studio ✓
3. Find the total number of years employed by all Engineers

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next – SQL Lesson 12: Order of execution of a Query
Previous – SQL Lesson 10: Queries with aggregates (Pt. 1)

The "HAVING" clause constraints are written the same way as the "WHERE" clause constraints, and are applied to the grouped rows. With our examples, this might not seem like a particularly useful construct, but if you imagine data with millions of rows with different properties, being able to apply additional constraints is often necessary to quickly make sense of the data.

Did you know?

If you aren't using the `GROUP BY` clause, a simple `WHERE` clause will suffice.

Exercise

For this exercise, you are going to dive deeper into **Employee** data at the film studio. Think about the different clauses you want to apply for each task.

Table: Employees

Role	SUM(years_employed)
Engineer	17

Exercise 11 — Tasks

1. Find the number of Artists in the studio (without a **HAVING** clause) ✓
2. Find the number of Employees of each role in the studio ✓
3. Find the total number of years employed by all Engineers ✓

```
SELECT role, SUM(years_employed)
FROM employees
GROUP BY role
HAVING role = "Engineer";
```

RESET

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

Next – SQL Lesson 12: Order of execution of a Query
Previous – SQL Lesson 10: Queries with aggregates (Pt. 1)