The HAVING Clause

In this lesson, we will learn about the HAVING clause.

We'll cover the following

The HAVING clause

The **HAVING** clause is utilized in SQL as a conditional clause with the `GROUP BY` clause. This conditional clause only returns rows where aggregate function results are matched with given conditions.

The `HAVING` clause was added to SQL because the `WHERE` keyword could not be used with aggregate functions.

Syntax

The basic syntax of the `HAVING` clause is as follows:

```
SELECT column1, column2, ... columnN

FROM table_name

WHERE [ conditions ]

GROUP BY column1, column2, ... columnN

HAVING [ conditions ]

ORDER BY column1, column2, ... columnN;
```

As you can see, the `HAVING` clause must follow the `GROUP BY` clause in a query and must also precede the `ORDER BY` clause if used.

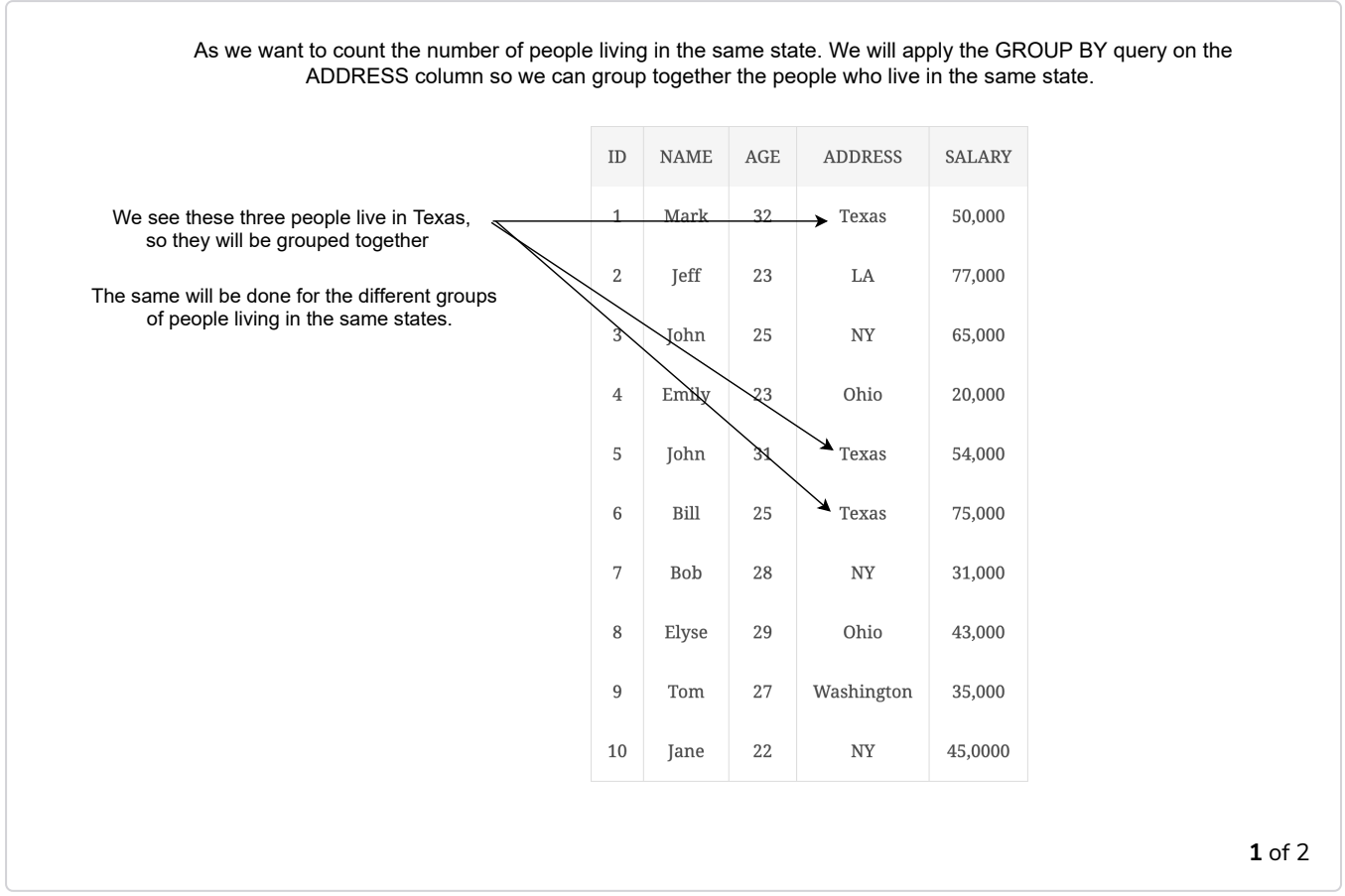
Example

Consider the CUSTOMERS table below but with a few changes:

ID	NAME	AGE	ADDRESS	SALARY
1	Mark	32	Texas	50,000
2	Jeff	23	LA	77,000
3	John	25	NY	65,000
4	Emily	23	Ohio	20,000
5	John	31	Texas	54,000
6	Bill	25	Texas	75,000
7	Bob	28	NY	31,000
8	Elyse	29	Ohio	43,000
9	Tom	27	Washington	35,000
10	Jane	22	NY	45,0000

As you can see, there are many customers that live at the same `ADDRESS` (i.e. live in the same state).

We want to write a SQL statement that returns the number of customers in each state, but only if that state has more than 2 customers:



The code for the above query is written below:

```
SELECT ADDRESS, COUNT(ID)
FROM CUSTOMERS
GROUP BY ADDRESS
HAVING COUNT(ID) > 2;
```

In **line 3**, the GROUP BY statement groups the customers based on their `ADDRESS` and then the `HAVING` clause in **line 4** checks to see if the number of customers living in this state is greater than two using the `COUNT()` function.

Quick quiz!

Q

Will the following SQL statement will return those ADDRESS (i.e. states) that have customers who earn a combine total greater than

80000?

```
SELECT ADDRESS, SUM(SALARY)
FROM CUSTOMERS
GROUP BY ADDRESS
HAVING SUM(SALARY) > 80000;
```

A) True

B) False

COMPLETED 0%

1 of 1

In the next lesson, we will learn to assign aliases to columns and tables.