In SQL Server we have got lot of aggregate functions. Examples  
1. Count()  
2. Sum()  
3. avg()  
4. Min()  
5. Max()  
  
**Group by** clause is used to group a selected set of rows into a set of summary rows by the values of one or more columns or expressions. It is always used in conjunction with one or more aggregate functions.

Create table tblDepartment

(

     ID int primary key,

     DepartmentName nvarchar(50),

     Location nvarchar(50),

     DepartmentHead nvarchar(50)

)

Go

Insert into tblDepartment values (1, 'IT', 'London', 'Rick')

Insert into tblDepartment values (2, 'Payroll', 'Delhi', 'Ron')

Insert into tblDepartment values (4, 'Other Department', 'Sydney', 'Cindrella')

Go

Create table tblEmployee

(

     ID int primary key,

     Name nvarchar(50),

     Gender nvarchar(50),

     Salary int,

     DepartmentId int foreign key references tblDepartment(Id)

)

Go

Insert into tblEmployee values (1, 'Tom', 'Male', 4000, 1)

Insert into tblEmployee values (2, 'Pam', 'Female', 3000, 3)

Insert into tblEmployee values (3, 'John', 'Male', 3500, 1)

Insert into tblEmployee values (4, 'Sam', 'Male', 4500, 2)

Insert into tblEmployee values (5, 'Todd', 'Male', 2800, 2)

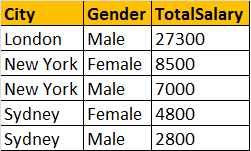
Insert into tblEmployee values (6, 'Ben', 'Male', 7000, 1)

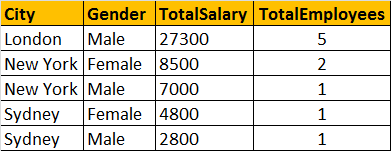
Insert into tblEmployee values (7, 'Sara', 'Female', 4800, 3)

Insert into tblEmployee values (8, 'Valarie', 'Female', 5500, 1)

Insert into tblEmployee values (9, 'James', 'Male', 6500, NULL)

Insert into tblEmployee values (10, 'Russell', 'Male', 8800, NULL)

**Query for retrieving total salaries by city**:  
We are applying SUM() aggregate function on Salary column, and grouping by city column. This effectively adds, all salaries of employees with in the same city.  
**Select City, SUM(Salary) as TotalSalary**  
**from tblEmployee**  
**Group by City**  
  
**Note:** If you omit, the group by clause and try to execute the query, you get an error - Column 'tblEmployee.City' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.   
  
Now, I want an sql query, which gives total salaries by City, by gender. The output should be as shown below.  
  


**Query for retrieving total salaries by city and by gender**: It's possible to group by multiple columns. In this query, we are grouping first by city and then by gender.   
**Select City, Gender, SUM(Salary) as TotalSalary**  
**from tblEmployee**  
**group by City, Gender**  
  
Now, I want an sql query, which gives total salaries and total number of employees by City, and by gender. The output should be as shown below.  
  


**Query for retrieving total salaries and total number of employees by City, and by gender**: The only difference here is that, we are using Count() aggregate function.  
**Select City, Gender, SUM(Salary) as TotalSalary,  
COUNT(ID) as TotalEmployees**  
**from tblEmployee**  
**group by City, Gender**  
  
**Filtering Groups:**  
WHERE clause is used to filter rows before aggregation, where as HAVING clause is used to filter groups after aggregations. The following 2 queries produce the same result.  
  
Filtering rows using WHERE clause, before aggrgations take place:  
**Select City, SUM(Salary) as TotalSalary**  
**from tblEmployee**  
**Where City = 'London'**  
**group by City**  
  
Filtering groups using HAVING clause, after all aggrgations take place:  
**Select City, SUM(Salary) as TotalSalary**  
**from tblEmployee**  
**group by City**  
**Having City = 'London'**  
  
From a performance standpoint, you cannot say that one method is less efficient than the other. Sql server optimizer analyzes each statement and selects an efficient way of executing it. As a best practice, use the syntax that clearly describes the desired result. Try to eliminate rows that   
you wouldn't need, as early as possible.  
  
**It is also possible to combine WHERE and HAVING**  
**Select City, SUM(Salary) as TotalSalary**  
**from tblEmployee**  
**Where Gender = 'Male'**  
**group by City**  
**Having City = 'London'**  
  
**Difference between WHERE and HAVING clause:**  
1. WHERE clause can be used with - Select, Insert, and Update statements, where as HAVING clause can only be used with the Select statement.  
2. WHERE filters rows before aggregation (GROUPING), where as, HAVING filters groups, after the aggregations are performed.