

# WITH CLAUSE

The **SQL WITH clause**, also known as Common Table Expressions (CTEs), was introduced by Oracle in the Oracle 9i release 2 database. The SQL WITH clause allows you to give a sub-query block a name (a process also called sub-query refactoring), which can be referenced in several places within the main SQL query.

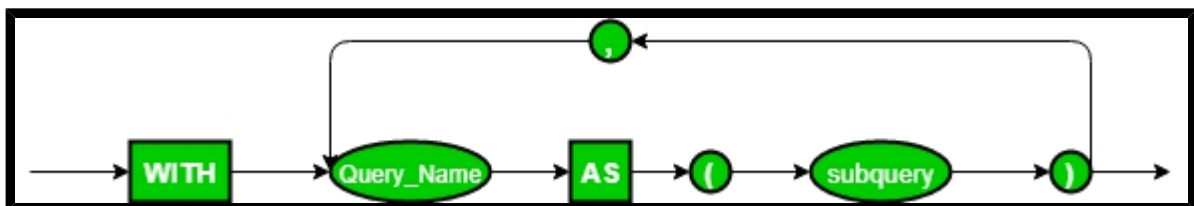
## What is the SQL WITH Clause?

- + The clause is used for defining a temporary relation such that the output of this temporary relation is available and is used by the query that is associated with the WITH clause.
- + Queries that have an associated WITH clause can also be written using nested sub-queries but doing so adds more complexity to read/debug the SQL query.
- + WITH clause is not supported by all database systems.
- + The name assigned to the sub-query is treated as though it were an inline view or table.
- + The SQL WITH clause was introduced by Oracle in the Oracle 9i release 2 database.

**Note:** Not all database systems support the WITH clause.

### Syntax

```
WITH temporaryTable (averageValue) AS (  
    SELECT AVG (Attr1)  
    FROM Table  
)  
SELECT Attr1  
FROM Table, temporaryTable  
WHERE Table.Attr1 > temporaryTable.averageValue;
```



In this query, WITH clause is used to define a temporary relation temporaryTable that has only 1 attribute averageValue. averageValue holds the average value of column Attr1 described in relation Table. The SELECT statement that follows the WITH clause will produce only those tuples where the value of Attr1 in relation Table is greater than the average value obtained from the WITH clause statement.

**Note:** When a query with a WITH clause is executed, first the query mentioned within the clause is evaluated and the output of this evaluation is stored in a temporary relation. Following this, the main query associated with the WITH clause is finally executed that would use the temporary relation produced.

## SQL WITH Clause Examples

Let us look at some of the examples of WITH Clause in SQL:

### Example 1: Finding Employees with Above-Average Salary

Find all the employee whose salary is more than the average salary of all employees. Name of the relation: Employee

EmployeeID	Name	Salary
100011	Smith	50000
100022	Bill	94000
100027	Sam	70550
100845	Walden	80000
115585	Erik	60000
1100070	Kate	69000

#### SQL Query

```
WITH temporaryTable (averageValue) AS (  
    SELECT AVG(Salary)  
    FROM Employee  
)  
SELECT EmployeeID, Name, Salary  
FROM Employee, temporaryTable  
WHERE Employee.Salary > temporaryTable.averageValue;
```

#### Output

EmployeeID	Name	Salary
100022	Bill	94000
100845	Walden	80000

**Explanation:** The average salary of all employees is 70591. Therefore, all employees whose salary is more than the obtained average lies in the output relation.

### Example 2: Finding Airlines with High Pilot Salaries

Find all the airlines where the total salary of all pilots in that airline is more than the average of total salary of all pilots in the database.

Name of the relation: **Pilot**

EmployeeID	Airline	Name	Salary
70007	Airbus 380	Kim	60000
70002	Boeing	Laura	20000

10027	Airbus 380	Will	80050
10778	Airbus 380	Warren	80780
115585	Boeing	Smith	25000
114070	Airbus 380	Katy	78000

## SQL Query

```
WITH totalSalary(Airline, total) AS (
    SELECT Airline, SUM(Salary)
    FROM Pilot
    GROUP BY Airline
),
airlineAverage (avgSalary) AS (
    SELECT avg(Salary)
    FROM Pilot
)
SELECT Airline
FROM totalSalary, airlineAverage
WHERE totalSalary.total > airlineAverage.avgSalary;
```

Airline

Airbus 380

**Explanation:** The total salary of all pilots of **Airbus 380 = 298,830** and that of **Boeing = 45000**. Average salary of all pilots in the table **Pilot = 57305**. Since only the total salary of all pilots of Airbus 380 is greater than the average salary obtained, so Airbus 380 lies in the output relation.

## Important Points About SQL | WITH Clause

+ The SQL WITH clause is good when used with complex SQL statements rather than simple ones. + It also allows you to break down complex SQL queries into smaller ones which make it easy for debugging and processing the complex queries. + The SQL WITH clause is basically a drop-in replacement to the normal sub-query. + The SQL WITH clause can significantly improve query performance by allowing the query optimizer to reuse the temporary result set, reducing the need to re-evaluate complex sub-queries multiple times.