

# Advanced SQL Queries

## (1) CTE:

CTE stands for common table expression. A CTE allows you to define a temporary named result set that available temporarily in the execution scope of a statement such as SELECT, INSERT, UPDATE, DELETE, or MERGE.

```
WITH CTE_Employee as
(
SELECT FirstName, Lastname, Gender, Salary
, COUNT(gender) OVER (PARTITION BY Gender) as TotalGender
,AVG(Salary) OVER (PARTITION BY Gender) as AvgSalary
FROM [SQL Queries]..EmployeeDemographics dem
JOIN [SQL Queries]..EmployeeSalary sal
ON dem.EmployeeID = sal.EmployeeID
WHERE Salary > '36000'
)
SELECT *
FROM CTE_Employee
```

Result:

Results		Messages				
	FirstName	Lastname	Gender	Salary	TotalGender	AvgSalary
1	Gary	Jimenez	Male	48000	2	43000
2	Marty	Ross	Male	38000	2	43000

## (2) STRING Functions:

We will create a new table for this and cover the following functions - TRIM,LTRIM and replace

```

CREATE TABLE EmployeeErrors(
EmployeeID varchar(50)
,FirstName varchar(50)
,LastName varchar(50)
)

Insert into EmployeeErrors Values
('1', 'Alex', 'Furgerson')
, (' 2', 'Ryan', 'Giggs')
,('9', 'DaVid', 'Beckham - Fired')

Select *
From EmployeeErrors

```

100 %

Results Messages

	EmployeeID	FirstName	LastName
1	1	Alex	Furgerson
2	2	Ryan	Giggs
3	9	DaVid	Beckham - Fired

TRIM:

```

Select EmployeeID, TRIM(EmployeeID) as IDTRIM
FROM EmployeeErrors

```

Results Messages

	EmployeeID	IDTRIM
1	1	1
2	2	2
3	9	9

LTRIM:

```

Select EmployeeID, LTRIM(EmployeeID) as IDTRIM
FROM EmployeeErrors

```

Results	
EmployeeID	IDTRIM
1	1
2	2
9	9
(3 rows affected)	

Replace:

```
Select LastName, REPLACE(LastName, '- Fired','') as LastNameFixed
FROM EmployeeErrors
```

Results	
LastName	LastNameFixed
Furgerson	Furgerson
Giggs	Giggs
Beckham - Fired	Beckham
(3 rows affected)	
Completion time: 2022-02-15T19:48:23.5148661-05:00	