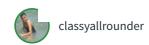


Trending Now DSA Data Structures Algorithms Interview Preparation Data Science Topic-wise Practice J

SQL | With Ties Clause



Read Discuss Courses Practice

This post is a continuation of <u>SQL Offset-Fetch Clause</u>

Now, we understand that how to use the Fetch Clause in Oracle Database, along with the Specified Offset and we also understand that Fetch clause is the newly added clause in the Oracle Database 12c or it is the new feature added in the Oracle database 12c.

Now consider the below example:

Suppose we a have a table named **myTable** with below data:

AD

ID	NAME	SALARY	
1	Geeks	10000	
4	Finch	10000	
2	RR	6000	
3	Dhoni	16000	
5	Karthik	7000	
6	Watson	10000	

Now, suppose we want the first three rows to be Ordered by Salary in descending order, then the below query must be executed:

Query:

SELECT * from myTable
order by salary desc
fetch first 3 rows only;

Output:

We got only first 3 rows order by Salary in Descending Order

ID	NAME	SALARY
3	Dhoni	16000
1	Geeks	10000
4	Finch	10000

Note: In the above result we got first 3 rows, ordered by Salary in Descending Order, but we have one more row with same salary i.e, the row with name **Watson** and Salary **10000**, but it didn't came up, because we restricted our output to first three rows only. But this is not optimal, because most of the time in live applications we will be required to display the tied rows also.

Real Life Example – Suppose we have 10 Racers running, and we have only 3 prizes i.e, first, second, third, but suppose, Racers 3 and 4 finished the race together in same time, so in this case we have a tie between 3 and 4 and that's why both are holder of Position 3.

With Ties

So, to overcome the above problem, Oracle introduces a clause known as **With Ties** clause. Now, let's see our previous example using With Ties clause.

Query:

```
SELECT * from myTable
order by salary desc
fetch first 3 rows With Ties;
```

Output:

See we get only first 3 rows order by Salary in Descending Order along with **Tied Row** also

ID	NAME	SALARY
3	Dhoni	16000
1	Geeks	10000
6	Watson	10000 // We get Tied Row also
4	Finch	10000

Now, see we got the **tied row** also, which we were not getting previously.

Note: We **get** the tied row in our output, only when we use the **order by** clause in our Select statement. Suppose, if we won't use order by clause, and still we are using **with ties** clause,

then we won't get the tied row in our output and the query behaves same as, if we are using **ONLY** clause **instead** of With Ties clause.

Example – Suppose we execute the below query(without using order by clause) :

Query:

```
SELECT * from myTable
fetch first 3 rows With Ties;
```

Output:

See we won't get the tied row because we didn't use order by clause

ID	NAME	SALARY
1	Geeks	10000
4	Finch	10000
2	RR	6000

In the above result we won't get the tied row and we get only first 3 rows. So **With Ties** is **tied** with **order by** clause, i.e, we get the tied row in output if and only if we use With Ties along with Order by clause.

Note: Please make sure that, you run these queries in Oracle Database 12c, because Fetch clause is the newly added feature in Oracle 12c, also With Ties, runs only in Oracle Database 12c, these queries **won't** run in below versions of 12c like 10g or 11g.

References: About Fetch Clause as well as With Ties Clause, Performing SQL Queries Online
Last Updated: 21 Mar, 2018

Similar Reads

1.	Difference between Having clause and Group by clause
2.	SQL Distinct Clause
3.	SQL WHERE Clause
4.	Top Clause in Microsoft SQL Server
5.	SQL Union Clause
6.	SQL WITH clause
7.	SQL Except Clause