**ORACLE UNPIVOT**

The Oracle UNPIVOT clause allows you to transpose columns to rows.

The UNPIVOT clause is opposite to the [PIVOT](https://www.oracletutorial.com/oracle-basics/oracle-pivot/) clause except that it does not de-aggregate data during the transposing process.

It’s Very helpful when you are using this under Sub Query .

For example – You can see the image for difference



Syntax for UNPIVOT-

**SELECT select\_list FROM table\_name**

**UNPIVOT [INCLUDE | EXCLUDE NULLS](**

 **unpivot\_clause**

 **unpivot\_for\_clause**

 **unpivot\_in\_clause**

**);**

**Types of UNPIOVT**

* The **unpivot\_clause** helps by allowing the user to specify a name for a column that will represents the unpivoted measure values.
* The unpivot\_for\_clause helps by allowing the user to specify the name for each column which will hold the measure’s values.
* The unpivot\_in\_clause will contain the pivoted columns that will be unpivoted.

**The INCLUDE | EXCLUDE NULLS clause allows user to include or exclude null-valued rows.**

* The INCLUDE NULLS clause will instruct Oracle to include the null-valued rows.
* The EXCLUDE NULLS clause, will eliminates the null-valued rows from the returned result set.

For Code example-

We create one table and make some entry.

**CREATE TABLE company\_prouduct\_test(id number PRIMARY KEY,**

 **fiscal\_year number,**

 **test\_product\_a number,**

 **test\_product\_b number,**

 **test\_product\_c number);**

We done some entry-

### **INSERT INTO company\_prouduct\_test(id, fiscal\_year, test\_product\_a, test\_product\_b, test\_product\_c)**

### **VALUES(4,2019, NULL, 5000, 750000);**

### **INSERT INTO company\_prouduct\_test(id, fiscal\_year, test\_product\_a, test\_product\_b, test\_product\_c)**

### **VALUES(1,2020, NULL, 5000, 750000);**

### **INSERT INTO company\_prouduct\_test(id, fiscal\_year, test\_product\_a, test\_product\_b, test\_product\_c)**

### **VALUES(2,2021, 7500, NULL, 20000);**

### **INSERT INTO company\_prouduct\_test(id, fiscal\_year, test\_product\_a, test\_product\_b, test\_product\_c)**

### **VALUES(3,2022, 7500, 22000, NULL);**



And now I can show you by using unpivot how we can represent the data.

## **SELECT \* FROM company\_prouduct\_test**

## **UNPIVOT INCLUDE NULLS ( quantity**

##  **FOR product\_code**

##  **IN (test\_product\_a AS 'A',**

##  **test\_product\_b AS 'B',**

##  **test\_product\_c AS 'C' ));**

Note:- You can exclude the null also

