

Data Transformation

There will be situations where you need to perform data transformation like column level data to row level data and vice versa.

Below are some of the methods which will aid us in this data transformation.

Rows to Columns using Case statement

	SALES_MONTH	PRODUCT_ID	SUM(TOTAL_AMOUNT)
1	01-FEB-15	200	4400
2	01-MAR-15	106	18568
3	01-JAN-15	100	374
4	01-APR-15	100	29282
5	01-MAR-15	101	16192
6	01-APR-15	101	33528
7	01-FEB-15	105	7986
8	01-APR-15	106	1386
9	01-MAR-15	100	16852
10	01-FEB-15	100	5016
11	01-FEB-15	101	5456
12	01-MAY-15	100	1254
13	01-JAN-15	101	2706
14	01-FEB-15	106	5126
15	01-MAR-15	105	17754
16	01-APR-15	105	880



	SALES_MONTH	TOTAL_100	TOTAL_101	TOTAL_105	TOTAL_106	TOTAL_200
1	01-JAN-15	374	2706	0	0	0
2	01-FEB-15	5016	5456	7986	5126	4400
3	01-MAR-15	16852	16192	17754	18568	0
4	01-APR-15	29282	33528	880	1386	0
5	01-MAY-15	1254	0	0	0	0

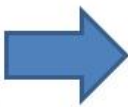
```
SELECT
    TRUNC(SALES_DATE, 'MON') AS SALES_MONTH,
    SUM(CASE WHEN PRODUCT_ID = 100 THEN TOTAL_AMOUNT ELSE 0 END) AS TOTAL_100,
    SUM(CASE WHEN PRODUCT_ID = 101 THEN TOTAL_AMOUNT ELSE 0 END) AS TOTAL_101,
    SUM(CASE WHEN PRODUCT_ID = 105 THEN TOTAL_AMOUNT ELSE 0 END) AS TOTAL_105,
    SUM(CASE WHEN PRODUCT_ID = 106 THEN TOTAL_AMOUNT ELSE 0 END) AS TOTAL_106,
    SUM(CASE WHEN PRODUCT_ID = 200 THEN TOTAL_AMOUNT ELSE 0 END) AS TOTAL_200
FROM SALES
GROUP BY TRUNC(SALES_DATE, 'MON')
ORDER BY TRUNC(SALES_DATE, 'MON')
```

Rows to Columns using PIVOT Analytic function

```
SELECT *  
  
FROM  
(  
    SELECT TRUNC (SALES_DATE, 'MON') AS SALES_MONTH, PRODUCT_ID,  
           TOTAL_AMOUNT  
    FROM SALES  
)  
    PIVOT (SUM (TOTAL_AMOUNT) FOR (PRODUCT_ID) IN (100, 101, 105, 106, 200))  
ORDER BY SALES_MONTH
```

Rows to Columns using LISTAGG Analytic function

	REGION	LAST_NAME
1	SOUTH	AMIRTHRAJ
2	SOUTH	JOSEPH
3	NORTH	MANN



	REGION	LAST_NAME
1	NORTH	MANN
2	SOUTH	AMIRTHRAJ, JOSEPH

```
SELECT REGION,  
       LISTAGG (LAST_NAME, ',') WITHIN GROUP (ORDER BY LAST_NAME) AS LAST_NAME  
FROM CUSTOMER  
GROUP BY REGION
```

Columns to Rows using UNION

	SALES_MONTH	TOTAL_100	TOTAL_101	TOTAL_105	TOTAL_106	TOTAL_200
1	01-JAN-15	374	2706	0	0	0
2	01-FEB-15	5016	5456	7986	5126	4400
3	01-MAR-15	16852	16192	17754	18568	0
4	01-APR-15	29282	33528	880	1386	0
5	01-MAY-15	1254	0	0	0	0



	SALES_MONTH	PRODUCT_ID	SUM(TOTAL_AMOUNT)
1	01-FEB-15	200	4400
2	01-MAR-15	106	18568
3	01-JAN-15	100	374
4	01-APR-15	100	29282
5	01-MAR-15	101	16192
6	01-APR-15	101	33528
7	01-FEB-15	105	7986
8	01-APR-15	106	1386
9	01-MAR-15	100	16852
10	01-FEB-15	100	5016
11	01-FEB-15	101	5456
12	01-MAY-15	100	1254
13	01-JAN-15	101	2706
14	01-FEB-15	106	5126
15	01-MAR-15	105	17754
16	01-APR-15	105	880

```

SELECT SALES_MONTH, 100 AS PRODUCT_ID, TOTAL_100 AS TOTAL_AMOUNT FROM SALES_PIVOT
UNION ALL
SELECT SALES_MONTH, 101 AS PRODUCT_ID, TOTAL_101 AS TOTAL_AMOUNT FROM SALES_PIVOT
UNION ALL
SELECT SALES_MONTH, 105 AS PRODUCT_ID, TOTAL_105 AS TOTAL_AMOUNT FROM SALES_PIVOT
UNION ALL
SELECT SALES_MONTH, 106 AS PRODUCT_ID, TOTAL_106 AS TOTAL_AMOUNT FROM SALES_PIVOT
UNION ALL
SELECT SALES_MONTH, 200 AS PRODUCT_ID, TOTAL_200 AS TOTAL_AMOUNT FROM SALES_PIVOT

```

Columns to Rows using UNPIVOT Analytical function

```

SELECT SALES_MONTH,
       PRODUCT_ID,
       TOTAL_AMOUNT
FROM SALES_PIVOT
UNPIVOT (TOTAL_AMOUNT FOR PRODUCT_ID IN (
        TOTAL_100 AS '100',
        TOTAL_101 AS '101',
        TOTAL_105 AS '105',
        TOTAL_106 AS '106',
        TOTAL_200 AS '200')
)

```