**In this video we will discuss**  
1. GROUPING\_ID function in SQL Server  
2. Difference between GROUPING and GROUPING\_ID functions  
3. Use of GROUPING\_ID function  
  
   
  
GROUPING\_ID function computes the level of grouping.  
  
**Difference between GROUPING and GROUPING\_ID**  
  
**Syntax :** GROUPING function is used on single column, where as the column list for GROUPING\_ID function must match with GROUP BY column list.

GROUPING(Col1)

GROUPING\_ID(Col1, Col2, Col3,...)

GROUPING indicates whether the column in a GROUP BY list is aggregated or not. Grouping returns 1 for aggregated or 0 for not aggregated in the result set.   
  
GROUPING\_ID() function concatenates all the GOUPING() functions, perform the binary to decimal conversion, and returns the equivalent integer. In short  
GROUPING\_ID(A, B, C) =  GROUPING(A) + GROUPING(B) + GROUPING(C)  
  
**Let us understand this with an example.** 

SELECT   Continent, Country, City, SUM(SaleAmount) AS TotalSales,

         CAST(GROUPING(Continent) AS NVARCHAR(1)) +

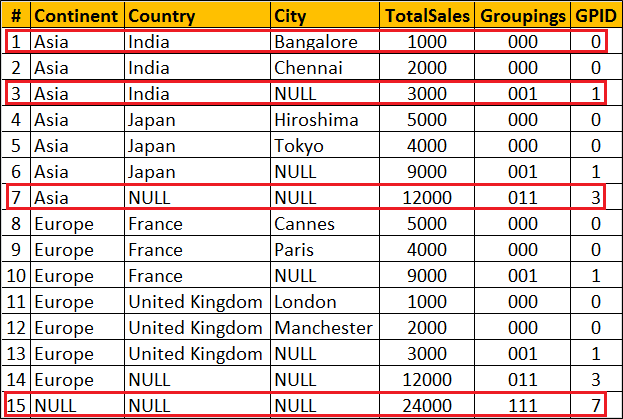
         CAST(GROUPING(Country) AS NVARCHAR(1)) +

         CAST(GROUPING(City) AS NVARCHAR(1)) AS Groupings,

         GROUPING\_ID(Continent, Country, City) AS GPID

FROM Sales

GROUP BY ROLLUP(Continent, Country, City)

**Query result :**  
  
  
**Row Number 1 :** Since the data is not aggregated by any column GROUPING(Continent), GROUPING(Country) and GROUPING(City) return 0 and as result we get a binar string with all ZEROS (000). When this converted to decimal we get 0 which is displayed in GPID column.  
  
**Row Number 7 :** The data is aggregated for Country and City columns, so GROUPING(Country) and GROUPING(City) return 1 where as  GROUPING(Continent) return 0. As result we get a binar string (011). When this converted to decimal we get 10 which is displayed in GPID column.  
  
**Row Number 15 :** This is the Grand total row. Notice in this row the data is aggregated by all the 3 columns. Hence all the 3 GROUPING functions return 1. So we get a binary string with all ONES (111). When this converted to decimal we get 7 which is displayed in GPID column.  
  
**Use of GROUPING\_ID function :** GROUPING\_ID function is very handy if you want to sort and filter by level of grouping.  
  
**Sorting by level of grouping :** 

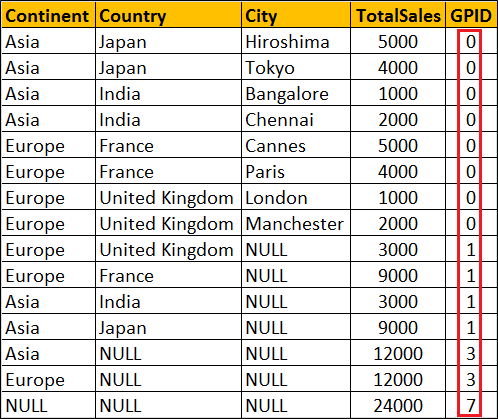
SELECT   Continent, Country, City, SUM(SaleAmount) AS TotalSales,

         GROUPING\_ID(Continent, Country, City) AS GPID

FROM Sales

GROUP BY ROLLUP(Continent, Country, City)

ORDER BY GPID

**Result :**   
   
  
**Filter by level of grouping :** The following query retrieves only continent level aggregated data

SELECT   Continent, Country, City, SUM(SaleAmount) AS TotalSales,

         GROUPING\_ID(Continent, Country, City) AS GPID

FROM Sales

GROUP BY ROLLUP(Continent, Country, City)

HAVING GROUPING\_ID(Continent, Country, City) = 3

**Result :**   
