

# Course 4 : 11 october 2021

---

## Last course

### Cubic model

We have seen that data warehouses can be conceptually seen as n-dimensions cubes

#### Browsing a cube

- Slicing : choosing a level of a dimension
- Roll-up : aggregating, ...

## This course

We will use many languages to implement those ideas

# Languages for Data Warehouses and DW operations

---

SQL has extensions for DW

## SQL

### Aggregations

#### Group by cube

- it basically computes union of all the possible groupings as we see in the example
- we are not considering the `null` values !

```
select item-name, color, size, sum(number)
from sales
group by cube(item-name, color, size)
```

Computes the union of eight different groupings of the *sales* relation:

```
{ (item-name, color, size), (item-name, color),
  (item-name, size), (color, size), (item-name),
  (color), (size), ( ) }
```

Note that there is always an internal optimization that the tools handle for us. We might have the impression that computing this grouping is better for the future for

### Group by roll-up

Roll up : showing level details from the top level to the bottom level of a same dimension.

Cube : only remove or add null values from the top right to the first dimensions

**what is rollup ? why use it ?**

## MDX (Multidimensional Expressions)

It is a query language for cubes that is used by many data warehousing systems. It is specially designed for DW. It is easy for simple queries but can be complex afterwards : be careful!

### Orthogonality

Orthogonality : one dimension is independant of another. SQL is almost an orthogonal system. A table can be stored "table" in the database. What results of a query ? a query result : a **table**. We can use this result as input for another query etc. But for any query, as complex as it is, the result will always have the same look : a table. **This proves SQL is a very regular language.**

If we have an orthogonal definition, the output is easy to manipulate.

Example : french is not an orthogonal language. Recteur devient rectrice par exemple.

Many languages are orthogonal : we can replace many variable types with other, for example make functions of functions, and still have the same output, etc. Here in MDX we don't have this orthogonality.

## On fait un exo

---