SQL Model Clause: A Gentle Introduction

Alex Nuijten

Ordina

The Netherlands

Keywords: SQL, Model Clause

Even though the title is "A Gentle Introduction", with the Model clause there is no such thing as a *gentle* introduction. In the same Oracle book where the Model Clause is explained, the "database data warehousing guide", Analytic Functions are explained as well.

Analytic Functions, as opposed to the Model Clause, are relatively easy to learn. The syntax might take some time to get used to, but once you get it, you get it.

The syntax of the Model Clause is complicated, and in my opinion, you will never get used to it.

In 2003 the Oracle Database 10g was released and this was the release where the Model clause was introduced. Whatever you can do with MicroSoft Excel, you can do with the Model Clause, like applying formulas, calculate new values and make forecasts. All this works on the query results, so even if the values are manipulated they are not persisted in the database.

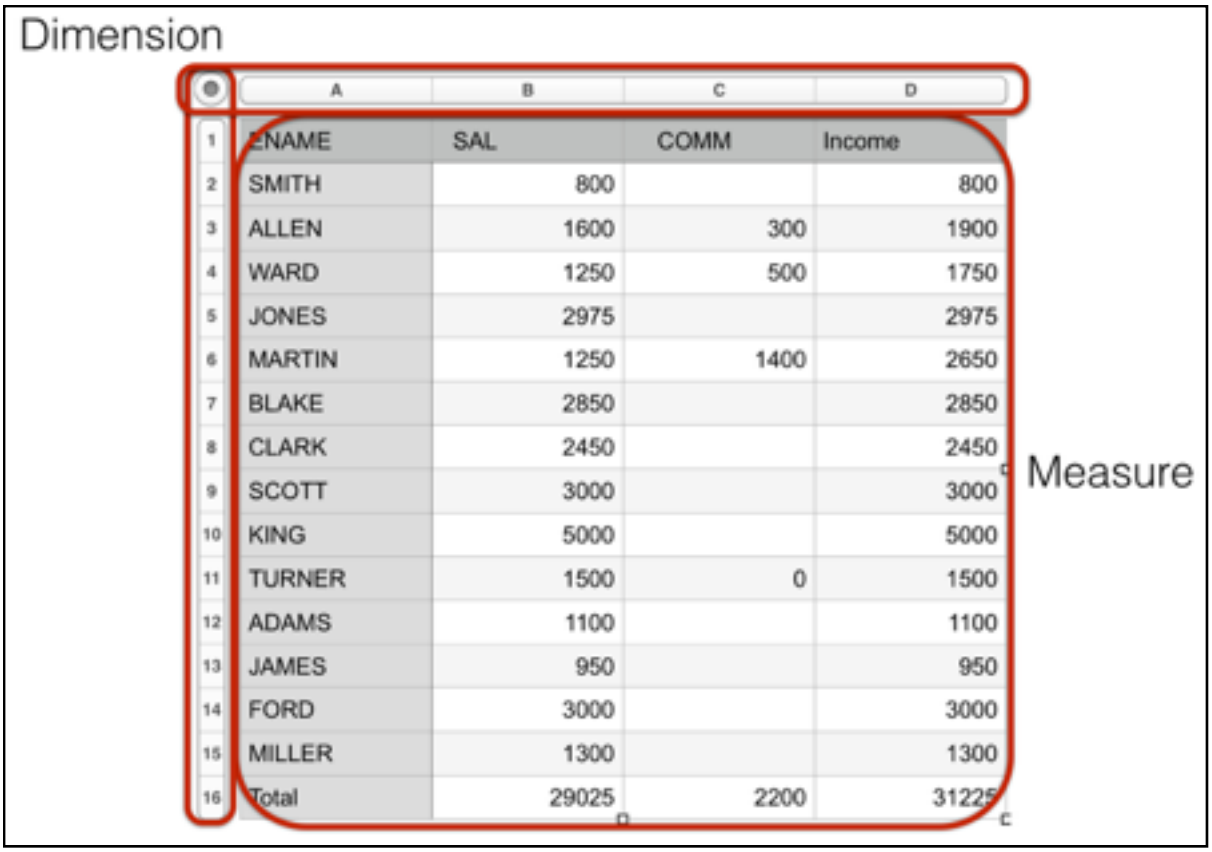
The syntax diagram is shown below, the different sections are described later.

```
MODEL [main] [RETURN {ALL|UPDATED} ROWS]
 [reference models]
 [PARTITION BY (<cols>)]
DIMENSION BY (<cols>)
MEASURES (<cols>)
 [IGNORE NAV] | [KEEP NAV]
RULES
 [UPSERT | UPDATE]
 [AUTOMATIC ORDER | SEQUENTIAL ORDER]
 [ITERATE (n) [UNTIL <condition>] ]
( <cell_assignment> = <expression> ... )
```

The MODEL keyword signals the begin of the Model Clause.

In the section DIMENSION BY you specify how you want to identify the individual cells. This is comparable to how you identify cells in Excel or the key of a relational table. It must produce a unique key for the result set. Dimensions can not be manipulated.

The MEASURES clause is where you specify the cells. The Measures are the values that can be



	A	B	C	D
1	ENAME	SAL	COMM	Income
2	SMITH	800		800
3	ALLEN	1600	300	1900
4	WARD	1250	500	1750
5	JONES	2975		2975
6	MARTIN	1250	1400	2650
7	BLAKE	2850		2850
8	CLARK	2450		2450
9	SCOTT	3000		3000
10	KING	5000		5000
11	TURNER	1500	0	1500
12	ADAMS	1100		1100
13	JAMES	950		950
14	FORD	3000		3000
15	MILLER	1300		1300
16	Total	29025	2200	31225

Illustration 1: Dimensions and Measures in an Excel Sheet

manipulated. Measures could be any value like quantity, price or length. Columns or Expressions could also be used as a measure.

In the RULES section you specify how the measures are manipulated, this is where all the action is. The rules are mainly assignment statements, where the left side represents a cell or range of cells and the right side is the value that you want to set. The right side can contain constants, expressions, bind variables, other cell values, or even aggregates on a range of cells.

Let's take a look at a simple example:

```
select *
  from emp
 model
dimension by (empno)
 measures (ename
          ,sal
          ,comm
          ,0 as income
          )
 rules ();
```

```
     EMPNO ENAME             SAL       COMM    INCOME
---------- ---------- --------- ----------  --------
      7369 SMITH            800                    0
      7499 ALLEN           1600        300         0
      7521 WARD            1250        500         0
      7566 JONES           2975                    0
      7654 MARTIN          1250       1400         0
      7698 BLAKE           2850                    0
      7782 CLARK           2450                    0
```

As you can see in the sample above, only dimensions and measures are shown in the result. There is one measure, with the alias of "income", that doesn't come from the EMP table. This extra measure is shown as a column in the result set.

When the following rule is added to the RULES section, the income for employee 7499 (Allen) will get a value in the income column:

```
income[7499] = sal [7499] + comm [7499]
```

This rule reads in plain English: "The Measure INCOME, identified by DIMENSION 7499 should get the value of the Measure SAL, identified by DIMENSION 7499 and add to that the value of the Measure COMM, identified by DIMENSION 7499".

The Income measure for employee 7499 will then contain the value of 1900 (sal: 1600 comm: 300).

If you want a rule to calculate the total income for each of the employees, it is possible to add a rule for each of the employees 7369, 7521, and so on. This is not really flexible, because when an employee is added to the table, you will also have to add a rule to the statement. A more flexible way is to add the following rule:

```
income[any] = sal [cv()] + comm [cv()]
```

The any-wildcard identifies all the different DIMENSIONs. The CV() function returns the "current value" of the DIMENSION that is being processed by the statement. Now the income-column will be filled with the calculated values.

```
     EMPNO ENAME             SAL       COMM   INCOME
---------- ---------- --------- ---------- --------
      7369 SMITH            800
      7499 ALLEN           1600        300     1900
      7521 WARD            1250        500     1750
      7566 JONES           2975
      7654 MARTIN          1250       1400     2650
      7698 BLAKE           2850
      7782 CLARK           2450
```

You may notice that not all employees have a value for the income column. This is, of course, caused by the missing values for the COMM column in the EMP table.

If this result is not what you want, it can be easily resolved with a NVL or COALESCE in the rule, but a more elegant way is to alter the MODEL keyword as follows:

```
model ignore nav
```

The "ignore nav" will instruct that all NULL are to be ignored in the statement. Now the income value is filled out for all employees.

EMPNO	ENAME	SAL	COMM	INCOME
7369	SMITH	800		800
7499	ALLEN	1600	300	1900
7521	WARD	1250	500	1750
7566	JONES	2975		2975
7654	MARTIN	1250	1400	2650
7698	BLAKE	2850		2850
7782	CLARK	2450		2450

And this was a very gentle introduction to the model clause, the presentation has a lot more examples which are explained in great detail.

Thank you.

Contact address:

Alex Nuijten

Ordina

Ringwade 1

3439 LM, Nieuwegein

The Netherlands

Phone: +31(0)6 10 39 56 54

Email: alex.nuijten@ordina.nl

Blog: nuijten.blogspot.com