allAPEX

SQL Model Clause:

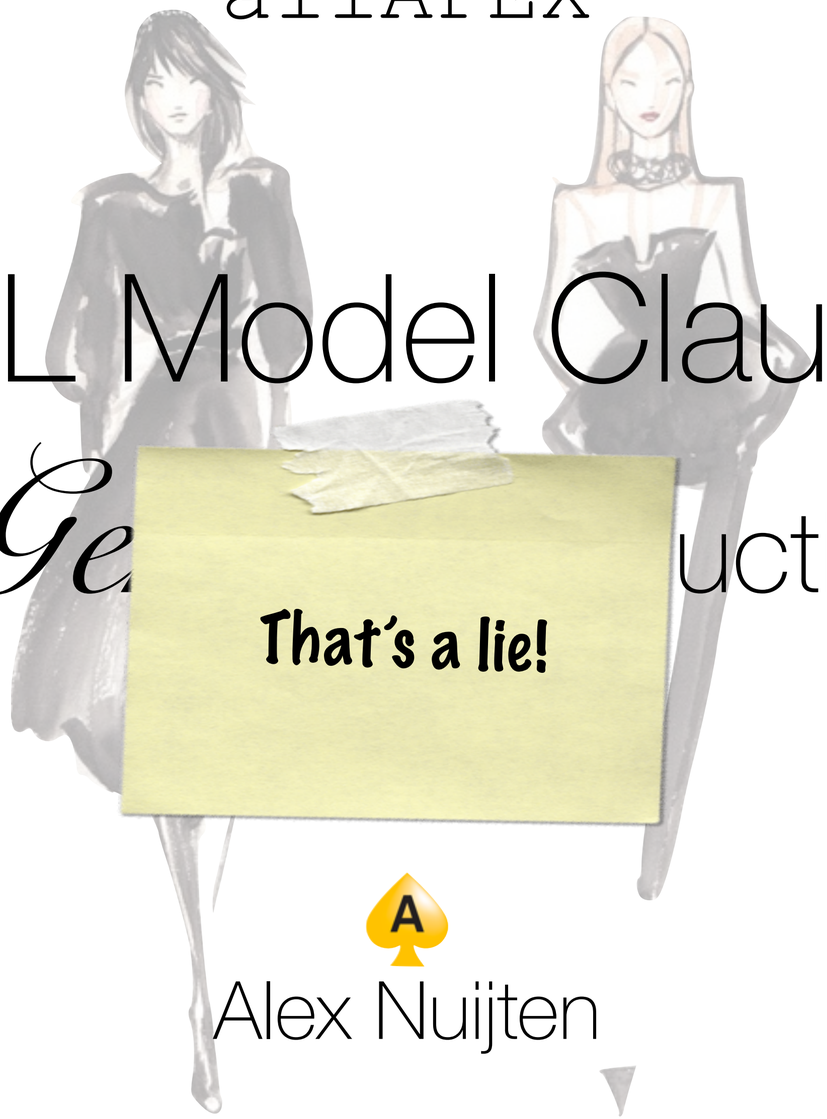
a *Gentle* introduction

Alex Nuijten

allAPEX

SQL Model Clause:

a Ge uction



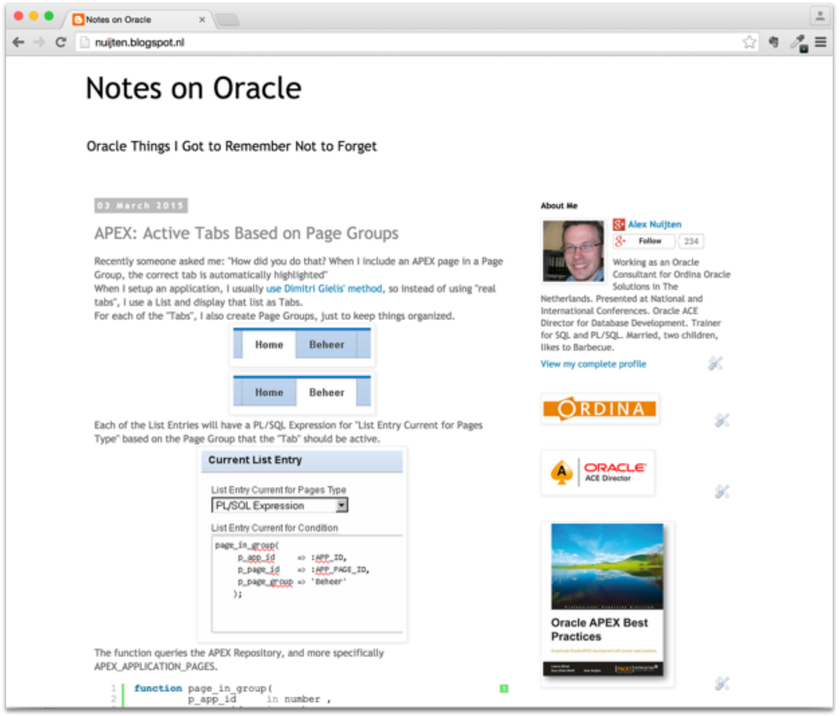
Alex Nuijten

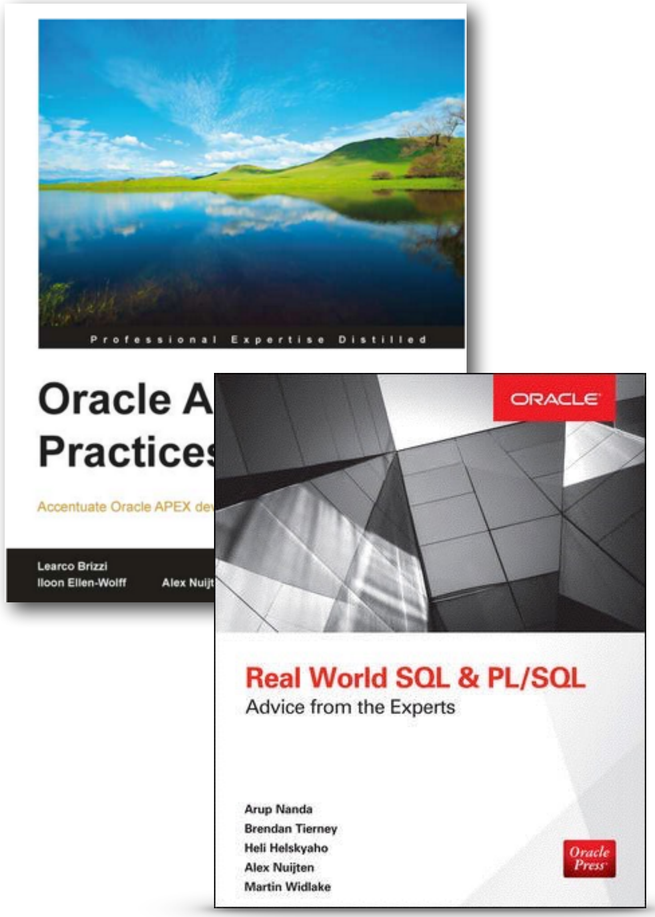
allAPEX

smart4 apex

@alexnuijten

nuijten.blogspot.com





Agenda


ORACLE
DATABASE
10g

2003

Oracle® Database Data Warehousing Guide
10*g* Release 2 (10.2)
B14223-02

- Multidimensional Array from Query Results
- Apply Formulas
- Calculate New Values
- Make Forecasts

Syntax

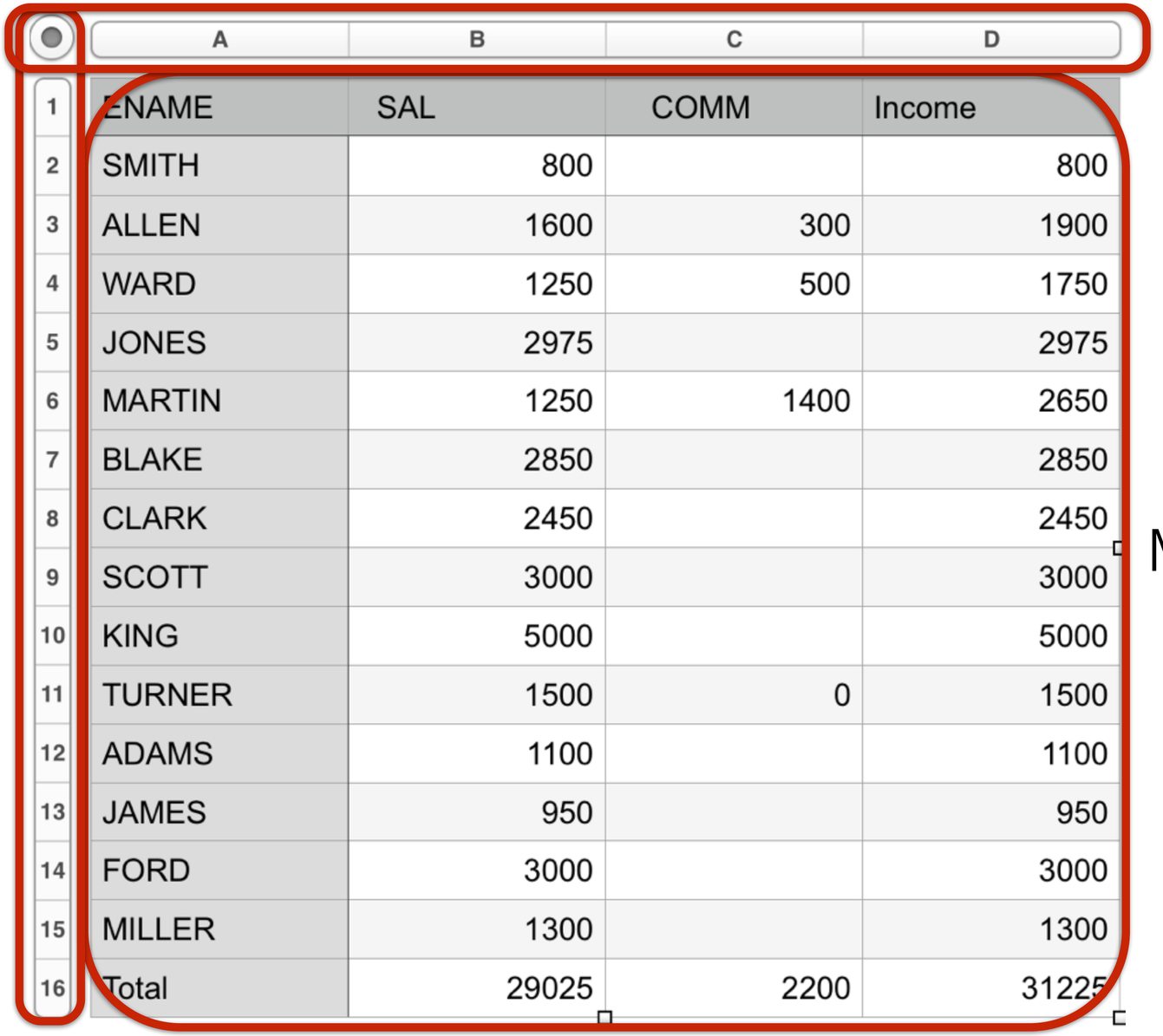
```
MODEL


DIMENSION BY (<cols>)
MEASURES (<cols>)
```

```
MODEL


DIMENSION BY (<cols>)
MEASURES (<cols>)

RULES


( <cell_assignment> = <expression> ... )
```

```
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> ... )
```

Terminology

Dimension



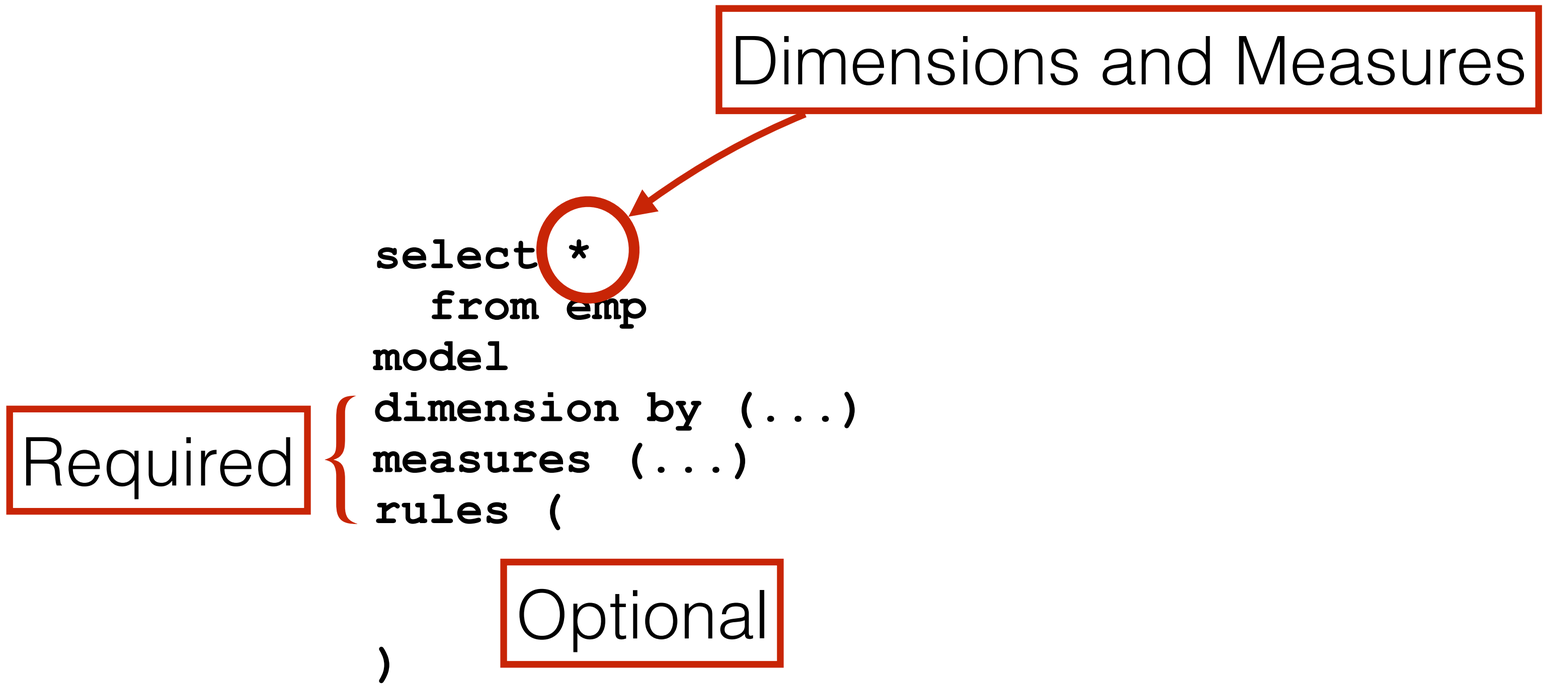
	A	B	C	D
1	ENAME	SAL	COMM	Income
2	SMITH	800		800
3	ALLEN	1600	300	1900
4	WARD	1250	500	1750
5	JONES	2975		2975
6	MARTIN	1250	1400	2650
7	BLAKE	2850		2850
8	CLARK	2450		2450
9	SCOTT	3000		3000
10	KING	5000		5000
11	TURNER	1500	0	1500
12	ADAMS	1100		1100
13	JAMES	950		950
14	FORD	3000		3000
15	MILLER	1300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

Measure

Rules to manipulate Measures

	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

Rules are like Spreadsheet Functions


Dimensions and Measures
select *
from emp
model
dimension by (...)
measures (...)
rules (
)
Required
Optional

```
MODEL


DIMENSION BY (<cols>)
MEASURES (<cols>)

RULES


( <cell_assignment> = <expression> ... )
```

Dimension

- “Key of a Relational Table”
- Must produce Unique Key for Result Set

```
select *
  from emp
model
dimension by (empno)
measures ()
rules ()
```

```
MODEL


DIMENSION BY (<cols>)
MEASURES (<cols>)

RULES


( <cell_assignment> = <expression> ... )
```

Measures

- Measurable Quantity like price or length
- Columns
- Expressions
-  Implicit Datatype Conversions

Measures

```
select *
  from emp
model
dimension by (empno)
measures (ename
          ,sal
          ,sal * 1.1 new_sal
          ,sysdate   today
          ,'some remarks' notes
          )
rules ()
```


```
 EMPNO      ENAME                SAL     NEW_SAL TODAY      NOTES
---------- ---------- ---------- ---------- --------- ------------
      7782 CLARK              2450       2695 23-JUN-15 some remarks
      7839 KING               5000       5500 23-JUN-15 some remarks
...
```

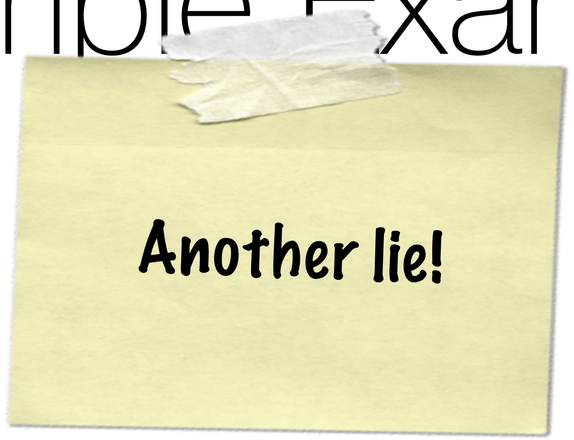
ORA-25137: Data value out of range

```
MODEL


DIMENSION BY (<cols>)
MEASURES (<cols>)

RULES


( <cell_assignment> = <expression> ... )
```

Rules

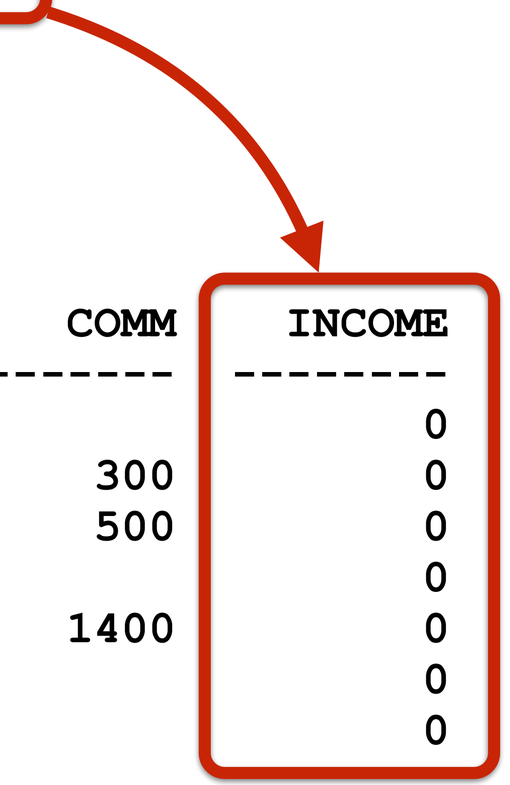
- Assignment Statement
- Left Side:
  - Represents cell or range of cells
- Right Side:
  - Expression involving constants
  - Bind variables
  - Individual cells
  - Aggregate function on range of cells

Simple Example



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

```
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

```
select *
  from emp
 model
dimension by (empno)
 measures (ename
           ,sal
           ,comm
           ,0 as income
           )
 rules (
  income[7499] = sal [7499] + comm [7499]
 );
```

EMPNO	ENAME	SAL	COMM	INCOME
7369	SMITH	800		0
7499	ALLEN	1600	300	1900
7521	WARD	1250	500	0
7566	JONES	2975		0
7654	MARTIN	1250	1400	0
7698	BLAKE	2850		0
7782	CLARK	2450		0

```
select *
  from emp
 model
dimension by (empno)
 measures (ename
           ,sal
           ,comm
           ,0 as income
           )
 rules (
  income[any] = sal [cv()] + comm [cv()]
 );
```

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		


```
select *
  from emp
 model ignore nav
dimension by (empno)
 measures (ename
           ,sal
           ,comm
           ,0 as income
           )
 rules (
  income[any] = sal [cv()] + comm [cv()]
 );
```

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

```
select *
  from emp
 model ignore nav
dimension by (ename)
 measures (sal
           ,comm
           ,0 income
           )
 rules (income [any]      = sal [cv()] + comm [cv()]
       ,sal    ['Total'] = sum (sal)[any]
       ,comm   ['Total'] = sum (comm)[any]
       ,income ['Total'] = sum (income)[any]
 );
```

```
ENAME             SAL       COMM   INCOME
-------- ---------- --------- -------
SMITH            800                800
ALLEN           1600        300    1900
WARD            1250        500    1750
JONES           2975               2975
MARTIN          1250       1400    2650
BLAKE           2850               2850
CLARK           2450               2450
sal ['Total'] = sum (sal)[any]     3000
KING            5000               5000
TURNER          1500          0    1500
ADAMS           1100               1100
JAMES            950                950
FORD            3000               3000
MILLER          1300               1300
Total          29025       2200   31225

15 rows selected.
```

Same Result without MODEL

```
select empno
      ,ename
      ,sal
      ,comm
      ,sal + coalesce (comm,0) as income
  from emp;
```

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

Same Result without MODEL

```
select case grouping (ename)
         when 1
         then 'Total'
         else ename
       end          as ename
      ,sum (sal)    as sal
      ,sum (comm)   as comm
      ,sum (sal + coalesce (comm,0)) as income
  from emp
 group by grouping sets ((ename), ());
```

ENAME	SAL	COMM	INCOME
ADAMS	1100		1100
ALLEN	1600	300	1900
BLAKE	2850		2850
CLARK	2450		2450
FORD	3000		3000
JAMES	950		950
JONES	2975		2975
KING	5000		5000
MARTIN	1250	1400	2650
MILLER	1300		1300
SCOTT	3000		3000
SMITH	800		800
TURNER	1500	0	1500
WARD	1250	500	1750
Total	29025	2200	31225

15 rows selected.


TIDAL WAVE

mei

MA	DI	WO	DO	VR	ZA	ZO
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

juni

MA	DI	WO	DO	VR	ZA	ZO
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

juli

MA	DI	WO	DO	VR	ZA	ZO
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

augustus

MA	DI	WO	DO	VR	ZA	ZO
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

september

MA	DI	WO	DO	VR	ZA	ZO
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

oktober

MA	DI	WO	DO	VR	ZA	ZO
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Rustig Gezellige bedrijvigheid Druk

```
create table themepark_visits
(id              number primary key
,visit_date      date
,no_of_visitors number
);
```

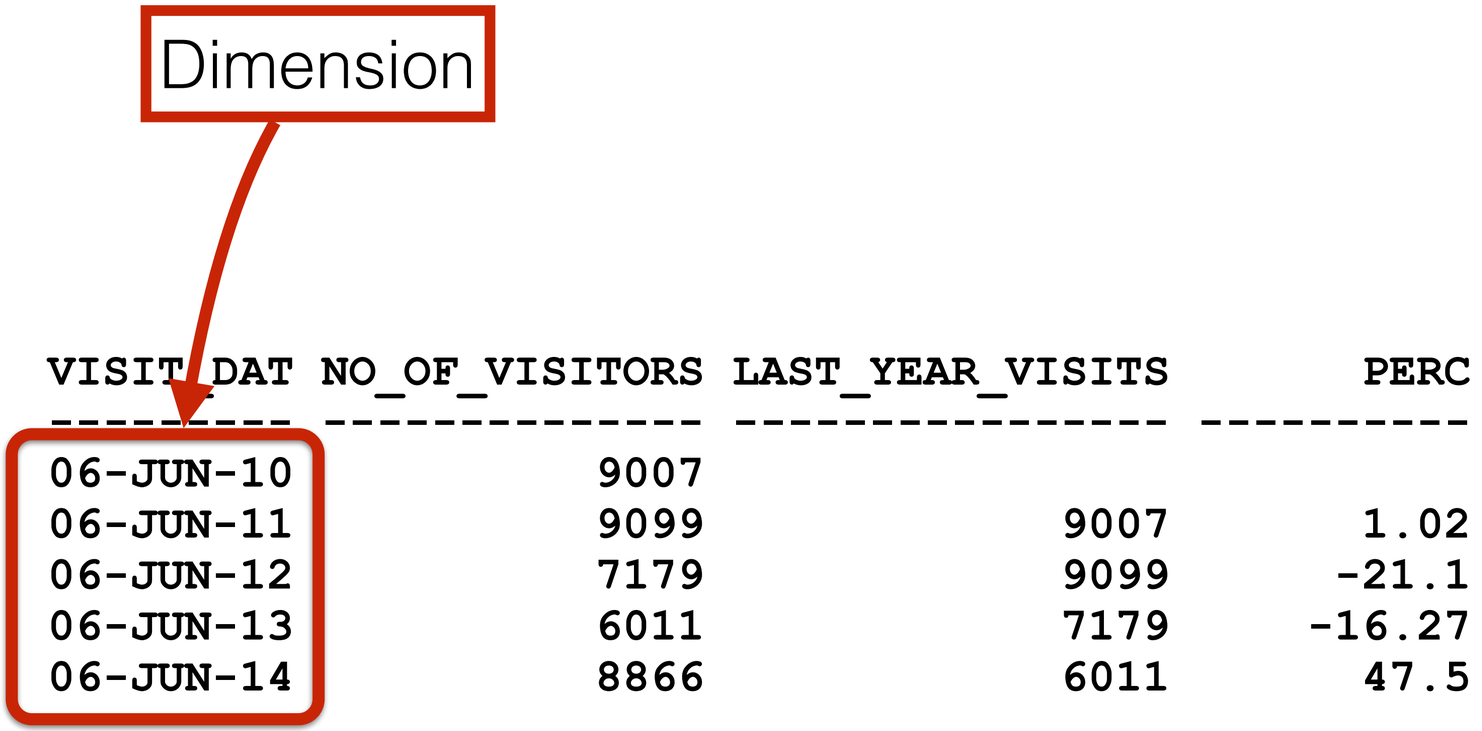
```
insert into themepark_visits
select rownum -- id
       ,to_date ('06-06-'||to_char (2010 + rownum -1)
                 , 'dd-mm-yyyy') -- visit_date
       ,trunc (dbms_random.value(1000,10000)) -- no_of_visitors
  from dual
 connect by level <= 5;
```

ID	VISIT_DAT	NO_OF_VISITORS
1	06-JUN-10	9007
2	06-JUN-11	9099
3	06-JUN-12	7179
4	06-JUN-13	6011
5	06-JUN-14	8866

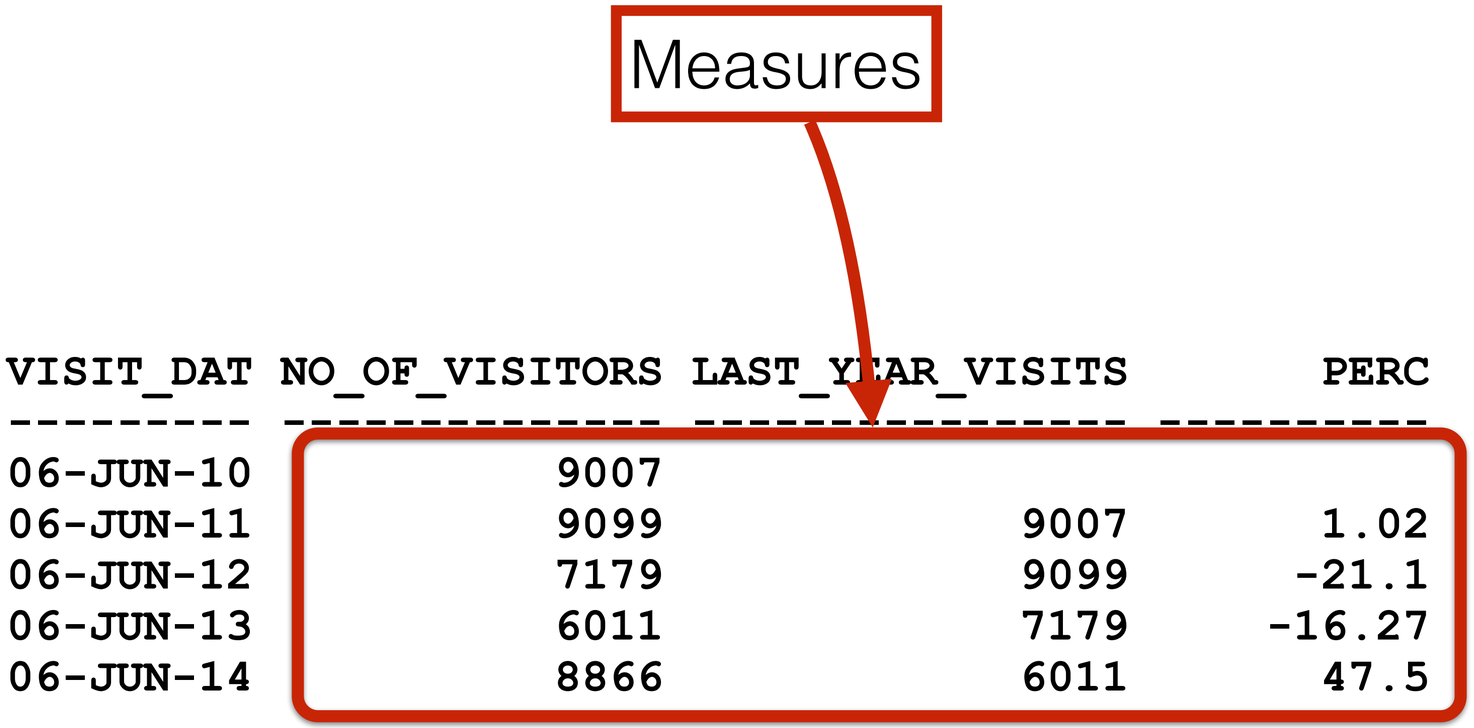
```
with visitors as
(
select tpv.visit_date
      ,tpv.no_of_visitors
      ,lag (tpv.no_of_visitors)
           over (order by tpv.visit_date) last_year_visits
  from themepark_visits tpv
)
```

```
with visitors as
(
select tpv.visit_date
      ,tpv.no_of_visitors
      ,lag (tpv.no_of_visitors)
           over (order by tpv.visit_date) last_year_visits
  from themepark_visits tpv
),
visits as
(
select visit_date
      ,no_of_visitors
      ,last_year_visits
      ,round (
          (no_of_visitors - last_year_visits)
          / last_year_visits * 100
       ,2) perc
  from visitors
)
```

```
select *
  from visits
 model
   dimension by (visit_date)
   measures (no_of_visitors
            ,last_year_visits
            ,perc
            )
   rules ();
```



VISIT_DAT	NO_OF_VISITORS	LAST_YEAR_VISITS	PERC
06-JUN-10	9007		
06-JUN-11	9099	9007	1.02
06-JUN-12	7179	9099	-21.1
06-JUN-13	6011	7179	-16.27
06-JUN-14	8866	6011	47.5



VISIT_DAT	NO_OF_VISITORS	LAST_YEAR_VISITS	PERC
06-JUN-10	9007		
06-JUN-11	9099	9007	1.02
06-JUN-12	7179	9099	-21.1
06-JUN-13	6011	7179	-16.27
06-JUN-14	8866	6011	47.5

```
no_of_visits [ date '2015-06-06'] = 0
```

VISIT_DAT	NO_OF_VISITORS	LAST_YEAR_VISITS	PERC
06-JUN-10	9007		
06-JUN-11	9099	9007	1.02
06-JUN-12	7179	9099	-21.1
06-JUN-13	6011	7179	-16.27
06-JUN-14	8866	6011	47.5
06-JUN-15	0		

“Number of Visitors for the current value of the Dimension minus 1 year”

```
no_of_visitors [ date '2015-06-06'] =
       no_of_visitors [cv() - interval '1' year] +
        ((no_of_visitors [cv() - interval '1' year] *
          perc [cv() -interval '1' year])
          /100)
```

VISIT_DAT	NO_OF_VISITORS	LAST_YEAR_VISITS	PERC
06-JUN-10	9007		
06-JUN-11	9099	9007	1.02
06-JUN-12	7179	9099	-21.1
06-JUN-13	6011	7179	-16.27
06-JUN-14	8866	6011	47.5
06-JUN-15	8866		

“Percentage for the current value of the Dimension minus 1 year”

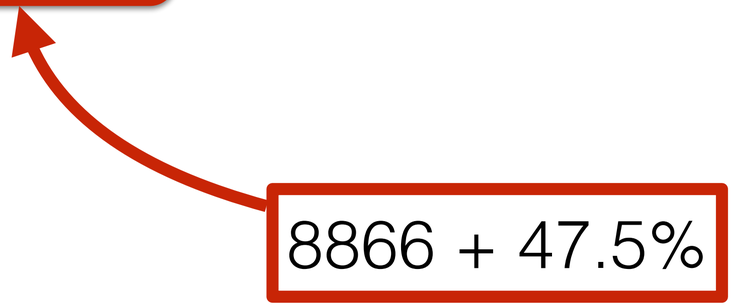
```
no_of_visitors [ date '2015-06-06'] =
       no_of_visitors [cv() - interval '1' year] +
        ((no_of_visitors [cv() - interval '1' year] *
          perc [cv() -interval '1' year])
          /100)
```

“Number of Visitors for next year, based on the Number of Visitors last year, take into account the growth percentage ”

```
no_of_visitors [ date '2015-06-06'] =
       no_of_visitors [cv() - interval '1' year] +
        ((no_of_visitors [cv() - interval '1' year] *
          perc [cv() -interval '1' year])
          /100)
```

VISIT_DAT	NO_OF_VISITORS	LAST_YEAR_VISITS	PERC
06-JUN-10	9007		
06-JUN-11	9099	9007	1.02
06-JUN-12	7179	9099	-21.1
06-JUN-13	6011	7179	-16.27
06-JUN-14	8866	6011	47.5
06-JUN-15	13077.35		

VISIT_DAT	NO_OF_VISITORS	LAST_YEAR_VISITS	PERC
06-JUN-10	9007		
06-JUN-11	9099	9007	1.02
06-JUN-12	7179	9099	-21.1
06-JUN-13	6011	7179	-16.27
06-JUN-14	8866	6011	47.5
06-JUN-15	13077.35		



```
no_of_visitors [ date '2015-06-06'] =
       no_of_visitors [cv() - interval '1' year] +
        ((no_of_visitors [cv() - interval '1' year] *
          perc [cv() -interval '1' year])
          /100)
```

“Average Percentage for all values of the Dimension”

```
no_of_visitors [ date '2015-06-06'] =
       no_of_visitors [cv() - interval '1' year] +
        ((no_of_visitors [cv() - interval '1' year] *
          avg (perc) [any])
          /100)
```

VISIT_DAT	NO_OF_VISITORS	LAST_YEAR_VISITS	PERC
06-JUN-10	9007		
06-JUN-11	9099	9007	1.02
06-JUN-12	7179	9099	-21.1
06-JUN-13	6011	7179	-16.27
06-JUN-14	8866	6011	47.5
06-JUN-15	9113.13975		

Average: 2.7875%

VISIT_DAT	NO_OF_VISITORS	LAST_YEAR_VISITS	PERC
06-JUN-10	9007		
06-JUN-11	9099	9007	1.02
06-JUN-12	7179	9099	-21.1
06-JUN-13	6011	7179	-16.27
06-JUN-14	8866	6011	47.5
06-JUN-15	9113.13975		


JHA TOURIST
DL 2CAD 2729
CNG

```
create table traffic_jams
(id         number primary key
,road       varchar2(5)
,jamdate    date
,jamlength number
);
```

```
insert into traffic_jams
select rn      -- id
      ,'A-27' -- road
      ,days   --jamdate
      ,trunc (dbms_random.value (1,100)) -- jamlength
  from dual
 model
   dimension by (rownum rn)
   measures (cast (null as date ) days)
   rules iterate (90)
   (
      days [iteration_number] =
       date '2014-01-01' + iteration_number
   );
```

```
SQL> select *
  2    from traffic_jams
  3   order by jamdate;

    ID ROAD  JAMDATE    JAMLENGTH
------ ----- --------- ----------
     0 A-27  01-JAN-14          63
    90 B-52  01-JAN-14          25
     1 A-27  02-JAN-14          15
    91 B-52  02-JAN-14          96
     2 A-27  03-JAN-14          44
    92 B-52  03-JAN-14          39
    93 B-52  04-JAN-14          53
     3 A-27  04-JAN-14           3
    94 B-52  05-JAN-14          95
...
```

```
select road
       ,to_char (jamdate
                 ,'fmday'
                 ) weekday
       ,avg (jamlength) avg_length
  from traffic_jams
 group by road
          ,to_char (jamdate
                    ,'fmday'
                    );
```

ROAD	WEEKDAY	AVG_LENGTH
A-27	sunday	62.0769231
A-27	monday	42.2307692
A-27	tuesday	48.5
B-52	monday	41.8461538
B-52	tuesday	47.6666667
B-52	saturday	38.6923077
A-27	friday	42.3846154
B-52	wednesday	35.9230769
A-27	saturday	46.4615385
B-52	friday	60.3076923
B-52	thursday	55.4615385
B-52	sunday	50.4615385
A-27	thursday	48
A-27	wednesday	46.5384615

14 rows selected.

```
select road
       ,to_char (jamdate
                ,'fmday'
                ,'nls_date_language=dutch'
                ) weekday
       ,avg (jamlength) avg_length
  from traffic_jams
 group by road
          ,to_char (jamdate
                   ,'fmday'
                   ,'nls_date_language=dutch'
                   );
```

ROAD	WEEKDAY	AVG_LENGTH
B-52	maandag	41.8461538
B-52	dinsdag	47.6666667
A-27	dinsdag	48.5
B-52	woensdag	35.9230769
B-52	vrijdag	60.3076923
A-27	zaterdag	46.4615385
A-27	zondag	62.0769231
B-52	donderdag	55.4615385
B-52	zondag	50.4615385
A-27	donderdag	48
A-27	vrijdag	42.3846154
A-27	woensdag	46.5384615
A-27	maandag	42.2307692
B-52	zaterdag	38.6923077

14 rows selected.

```
SQL> select to_char (sysdate + rownum
  2                 ,'fmday'
  3                 ,'nls_date_language=slovenian'
  4                 ) weekday
  5    from dual
  6  connect by level <= 7
  7  /
```

```
SQL> select road
  2       ,jamdate
  3       ,jamlength
  4    from traffic_jams
  5   where to_char (jamdate, 'fmday') = 'tuesday';

ROAD  JAMDATE   JAMLENGTH
----- --------- ---------
A-27  07-JAN-14        93
A-27  14-JAN-14        48
A-27  21-JAN-14        30
A-27  28-JAN-14        34
A-27  04-FEB-14        45
A-27  11-FEB-14        59
A-27  18-FEB-14        73
A-27  25-FEB-14         2
A-27  04-MAR-14        29
A-27  11-MAR-14        94
A-27  18-MAR-14         4
A-27  25-MAR-14        71
B-52  07-JAN-14        65
B-52  14-JAN-14        97
...
```

```
select road
       ,jamdate
       ,jamlength
    from traffic_jams
   where to_char (jamdate, 'fmday') = 'tuesday'
model
   partition by (road)
   dimension by (jamdate)
   measures (jamlength)
   rules (
      jamlength [date '2014-06-10']
          = avg (jamlength) [any]
   );
```

ROAD	JAMDATE	JAMLENGTH
B-52	07-JAN-14	65
B-52	14-JAN-14	97
B-52	21-JAN-14	61
B-52	28-JAN-14	27
B-52	04-FEB-14	47
B-52	11-FEB-14	15
B-52	18-FEB-14	26
B-52	25-FEB-14	81
B-52	04-MAR-14	38
B-52	11-MAR-14	15
B-52	18-MAR-14	27
B-52	25-MAR-14	73
B-52	10-JUN-14	47.6666667

```
A-27  07-JAN-14          93
A-27  14-JAN-14          48
A-27  21-JAN-14          30
A-27  28-JAN-14          34
A-27  04-FEB-14          45
A-27  11-FEB-14          59
A-27  18-FEB-14          73
A-27  25-FEB-14           2
A-27  04-MAR-14          29
A-27  11-MAR-14          94
A-27  18-MAR-14           4
A-27  25-MAR-14          71
A-27  10-JUN-14        48.5

26 rows selected.
```

```
select road
      ,jamdate
      ,jamlength
   from traffic_jams
  where to_char (jamdate, 'fmday') = 'tuesday'
model return updated rows
  partition by (road)
  dimension by (jamdate)
  measures (jamlength)
  rules (
     jamlength [date '2014-06-10']
         = avg (jamlength) [any]
  );
```

ROAD	JAMDATE	JAMLENGTH
B-52	10-JUN-14	47.6666667
A-27	10-JUN-14	48.5

```
select road
       ,jamdate
       ,weekday
       ,jamlength
   from traffic_jams
model return updated rows
  partition by (road)
  dimension by (jamdate
                ,to_char (jamdate, 'fmday') weekday)
  measures (jamlength)
  rules (jamlength [date '2014-06-02', 'monday']
                = avg (jamlength) [any, 'monday']
        ,jamlength [date '2014-06-03', 'tuesday']
                = avg (jamlength) [any, 'tuesday']
        ,jamlength [date '2014-06-04', 'wednesday']
                = avg (jamlength) [any, 'wednesday']
        ,jamlength [date '2014-06-05', 'thursday']
                = avg (jamlength) [any, 'thursday']
        ,jamlength [date '2014-06-06', 'friday']
                = avg (jamlength) [any, 'friday']
        ,jamlength [date '2014-06-07', 'saturday']
                = avg (jamlength) [any, 'saturday']
        ,jamlength [date '2014-06-08', 'sunday']
                = avg (jamlength) [any, 'sunday']
  )
```

```
jamlength [date '2014-06-02', 'monday']
      = avg (jamlength) [any, 'monday']
```

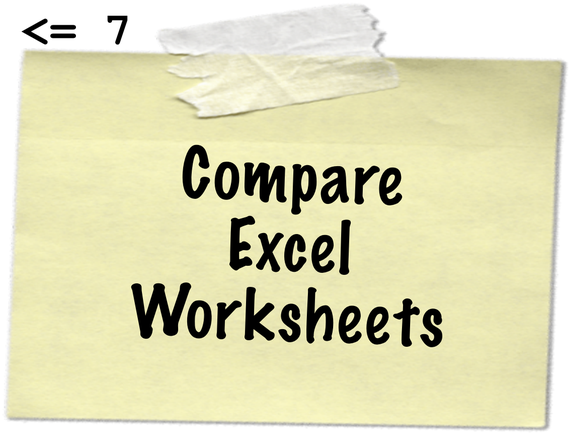
ROAD	JAMDATE	WEEKDAY	JAMLENGTH
A-27	02-JUN-14	monday	42.2307692
A-27	03-JUN-14	tuesday	48.5
A-27	04-JUN-14	wednesday	46.5384615
A-27	05-JUN-14	thursday	48
A-27	06-JUN-14	friday	42.3846154
A-27	07-JUN-14	saturday	46.4615385
A-27	08-JUN-14	sunday	62.0769231
B-52	02-JUN-14	monday	41.8461538
B-52	03-JUN-14	tuesday	47.6666667
B-52	04-JUN-14	wednesday	35.9230769
B-52	05-JUN-14	thursday	55.4615385
B-52	06-JUN-14	friday	60.3076923
B-52	07-JUN-14	saturday	38.6923077
B-52	08-JUN-14	sunday	50.4615385

14 rows selected.

```
select rownum rn
       ,trunc (sysdate + 7, 'iw') + rownum - 1
            next_week
  from dual
connect by level <= 7
```

```
        RN NEXT_WEEK
---------- ---------
         1 02-JUN-14
         2 03-JUN-14
         3 04-JUN-14
         4 05-JUN-14
         5 06-JUN-14
         6 07-JUN-14
         7 08-JUN-14

7 rows selected.
```

```
...
reference weekdays
        on (select rownum rn
                  ,trunc (sysdate + 7, 'iw') + rownum - 1
                     next_week
              from dual
             connect by level <= 7
            )
      dimension by (rn)
      measures (next_week)
...
```



```
...
main result
partition by (road)
dimension by (jamdate, to_char (jamdate, 'fmday') dd)
measures (jamlength)
rules iterate (7)(
   ...
)
...
```

Reference (“Worksheet”)

```
...
jamlength [
   weekdays.next_week [iteration_number + 1]
    ,to_char (weekdays.next_week [iteration_number + 1], 'fmday')]
  = avg (jamlength)
     [any, to_char (weekdays.next_week [iteration_number + 1]
                    , 'fmday')
                  ]
...
```

Iterate Built-In

```
select road
       ,jamdate
       ,to_char (jamdate, 'fmday') weekday
       ,jamlength
  from traffic_jams
model return updated rows
 reference weekdays
       on (select rownum rn
                  ,trunc (sysdate + 7, 'iw') + rownum - 1
                    next_week
             from dual
           connect by level <= 7
          )
      dimension by (rn)
      measures (next_week)
main result
partition by (road)
dimension by (jamdate, to_char (jamdate, 'fmday') dd)
measures (jamlength)
rules iterate (7)(
     jamlength [weekdays.next_week [iteration_number + 1]
          ,to_char (weekdays.next_week [iteration_number + 1], 'fmday')
          ] = avg (jamlength) [any
                               ,to_char (weekdays.next_week [iteration_number + 1]
                               ,'fmday')
                               ]
 )
```

ROAD	JAMDATE	WEEKDAY	JAMLENGTH
A-27	02-JUN-14	monday	42.2307692
A-27	03-JUN-14	tuesday	48.5
A-27	04-JUN-14	wednesday	46.5384615
A-27	05-JUN-14	thursday	48
A-27	06-JUN-14	friday	42.3846154
A-27	07-JUN-14	saturday	46.4615385
A-27	08-JUN-14	sunday	62.0769231
B-52	02-JUN-14	monday	41.8461538
B-52	03-JUN-14	tuesday	47.6666667
B-52	04-JUN-14	wednesday	35.9230769
B-52	05-JUN-14	thursday	55.4615385
B-52	06-JUN-14	friday	60.3076923
B-52	07-JUN-14	saturday	38.6923077
B-52	08-JUN-14	sunday	50.4615385

14 rows selected.


DANGER
WARNING
GENERATOR ON LOAD
MAINS ON LOAD
IMPORTANT!

Generate Rows

```
select r
  from dual
 model
   dimension by (rownum rn)
   measures (cast (null as number ) r)
   rules iterate (7)
   (
      r [iteration_number] = iteration_number
   );
```

Generate Rows

```
select r                                                    R
  from dual                                                 -----
 model                                                        1
   dimension by (rownum rn)                                   0
   measures (cast (null as number ) r)                        2
   rules iterate (7)                                          3
   (                                                          4
      r [iteration_number] = iteration_number                 5
   );                                                         6
```

Generate Rows

```
select to_char (days, 'fmday') weekdays
  from dual
 model
   dimension by (rownum rn)
   measures (cast (null as date ) days)
   rules iterate (7)
   (
      days [iteration_number] =
         date '2014-06-02' + iteration_number
   )
```

Generate Rows

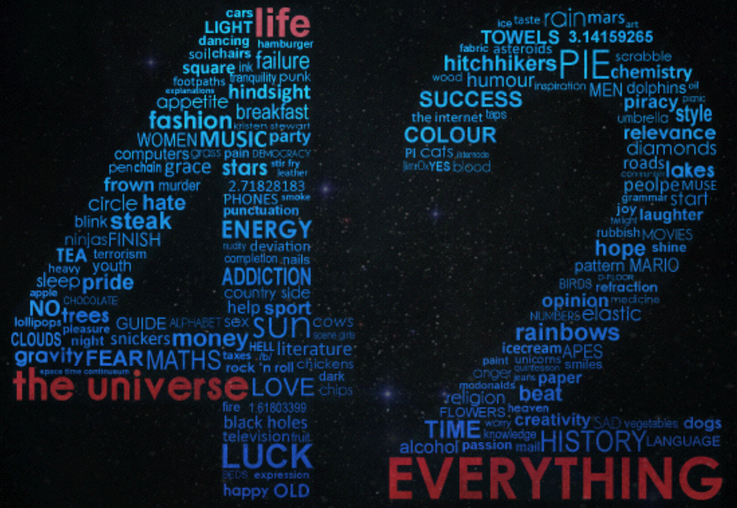
```
select to_char (days, 'fmday') weekdays
  from dual
 model
   dimension by (rownum rn)
   measures (cast (null as date ) days)
   rules iterate (7)
   (
      days [iteration_number] =
         date '2014-06-02' + iteration_number
   )
```

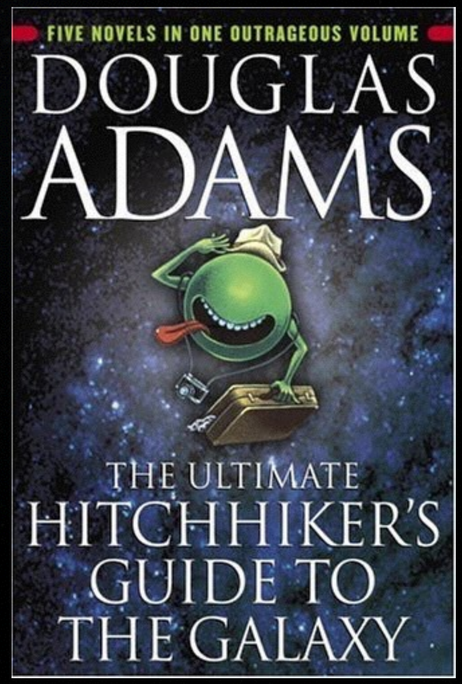
```
WEEKDAYS
----------
tuesday
monday
wednesday
thursday
friday
saturday
sunday
```

```
select *
  from dual
  model
  dimension by (rownum rn)
  measures (cast (null as varchar2(2)) as letter)
  rules iterate (26) (
    letter [iteration_number + 1]= chr(65 + iteration_number)
  )
  order by letter
```

RN	LE
1	A
2	B
3	C
4	D
5	E
6	F
7	G
8	H
9	I
10	J
11	K
12	L
13	M
14	N
15	O
16	P
17	Q
18	R
19	S
20	T
21	U
22	V
23	W
24	X
25	Y
26	Z

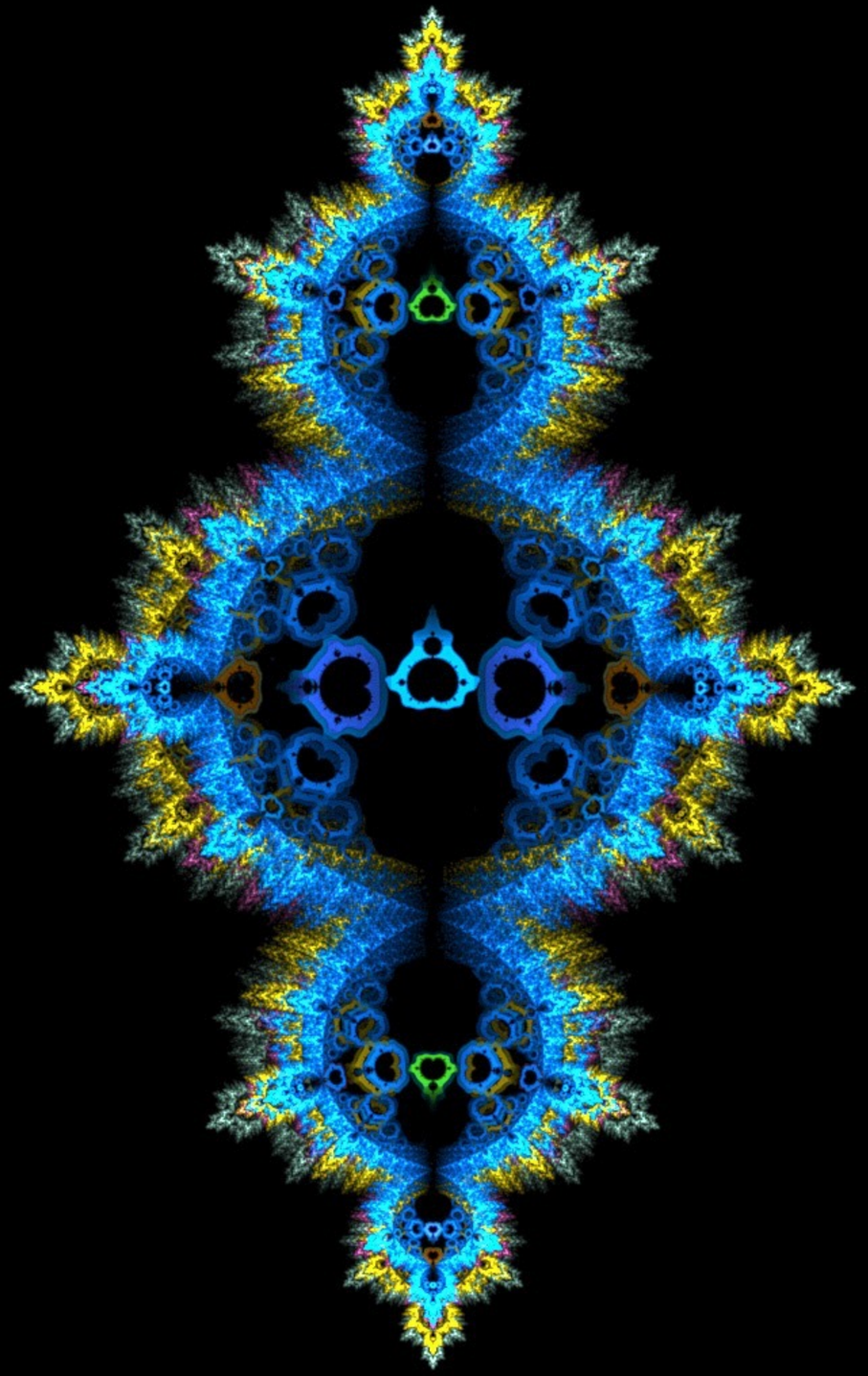
26 rows selected.


cars
life
LIGHT
dancing
hamburger
soil
chairs
failure
square
ink
footpaths
tranquility
punk
explanations
hindsight
appetite
breakfast
kristen stewart
fashion
WOMEN
MUSIC
party
computers
grass
pain
DEMOCRACY
pen
chain
grace
stars
stir fry
leather
frown
murder
2.71828183
PHONES
smoke
punctuation
circle
hate
blink
steak
ENERGY
ninjas
FINISH
rusty
deviation
TEA
terrorism
completion
nails
heavy
youth
sleep
pride
ADDICTION
apple
CHOCOLATE
country
side
NO
help
sport
trees
lollipops
pleasure
GUIDE
ALPHABET
sex
sun
cows
CLOUDS
night
snickers
money
HELL
literature
gravity
FEAR
MATHS
taxes
rock
'n roll
chickens
space time continuum
the universe
LOVE
dark
chips
fire
1.61803399
black holes
television
fruit
LUCK
BEDS
expression
happy
OLD
ice
taste
rain
mars
art
TOWELS
3.14159265
fabric
asteroids
hitchhikers
PIE
scrabble
wood
humour
inspiration
chemistry
MEN
dolphins
elf
SUCCESS
piracy
picnic
the internet
taps
umbrella
style
COLOUR
relevance
PI
cats
diamonds
staircase
roads
lakes
blood
peolpe
MUSE
grammar
start
joy
laughter
rubbish
MOVIES
hope
shine
pattern
MARIO
BIRDS
refraction
opinion
medicine
NUMBERS
elastic
rainbows
icecream
APES
paint
unicorns
smiles
anger
paper
religion
beat
FLOWERS
heaven
creativity
SAD
vegetables
dogs
TIME
knowledge
HISTORY
LANGUAGE
alcohol
passion
mail
EVERYTHING

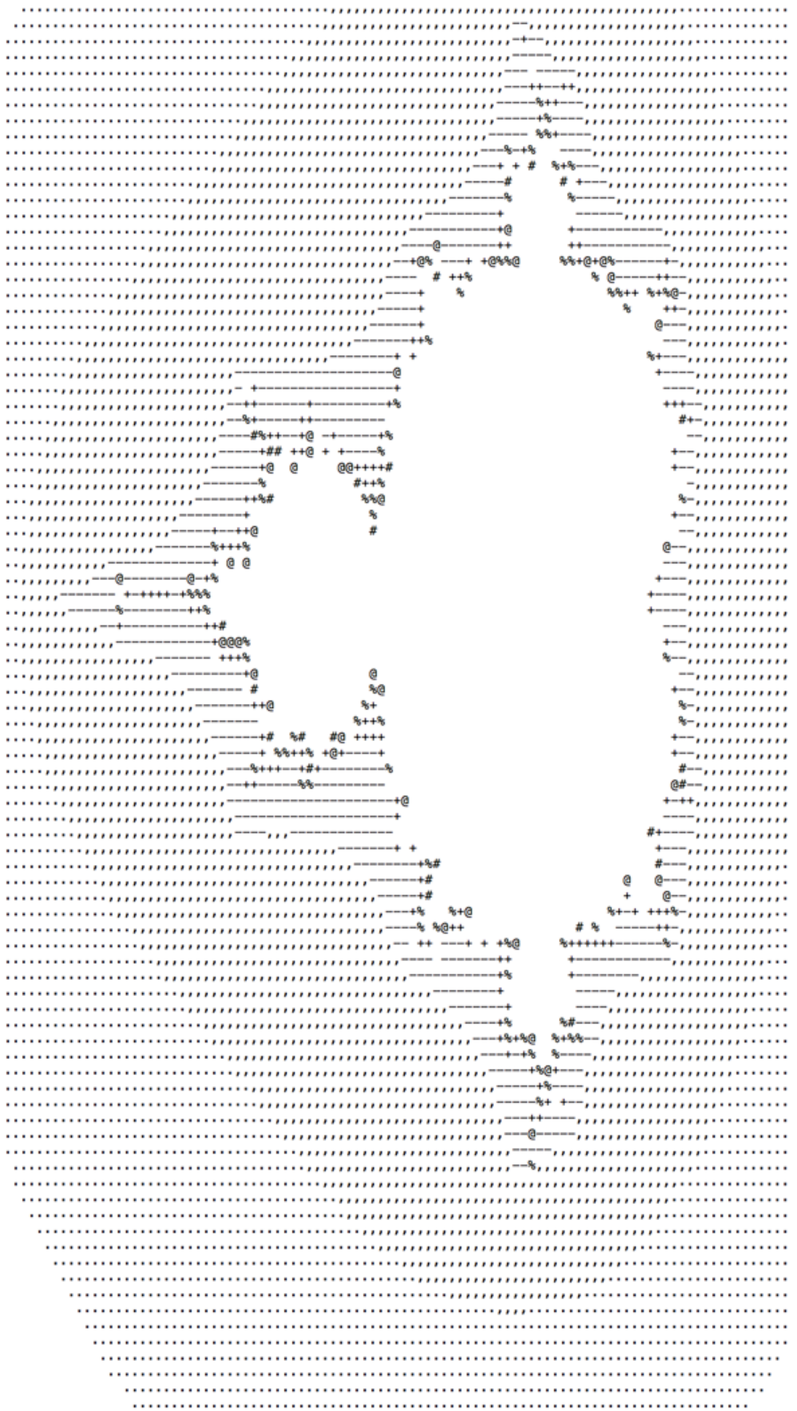

FIVE NOVELS IN ONE OUTRAGEOUS VOLUME
DOUGLAS ADAMS
THE ULTIMATE
HITCHHIKER'S GUIDE TO THE GALAXY

```
select fibo
  from dual
  model
  dimension by (rownum rn)
  measures (cast (null as number) fibo)
  rules (
     fibo [0] = 0
    ,fibo [1] = 1
    ,fibo [for rn from 2 to 10 increment 1] =
         fibo[cv() - 1] + fibo [cv() - 2]
  )
 order by rn
/
```

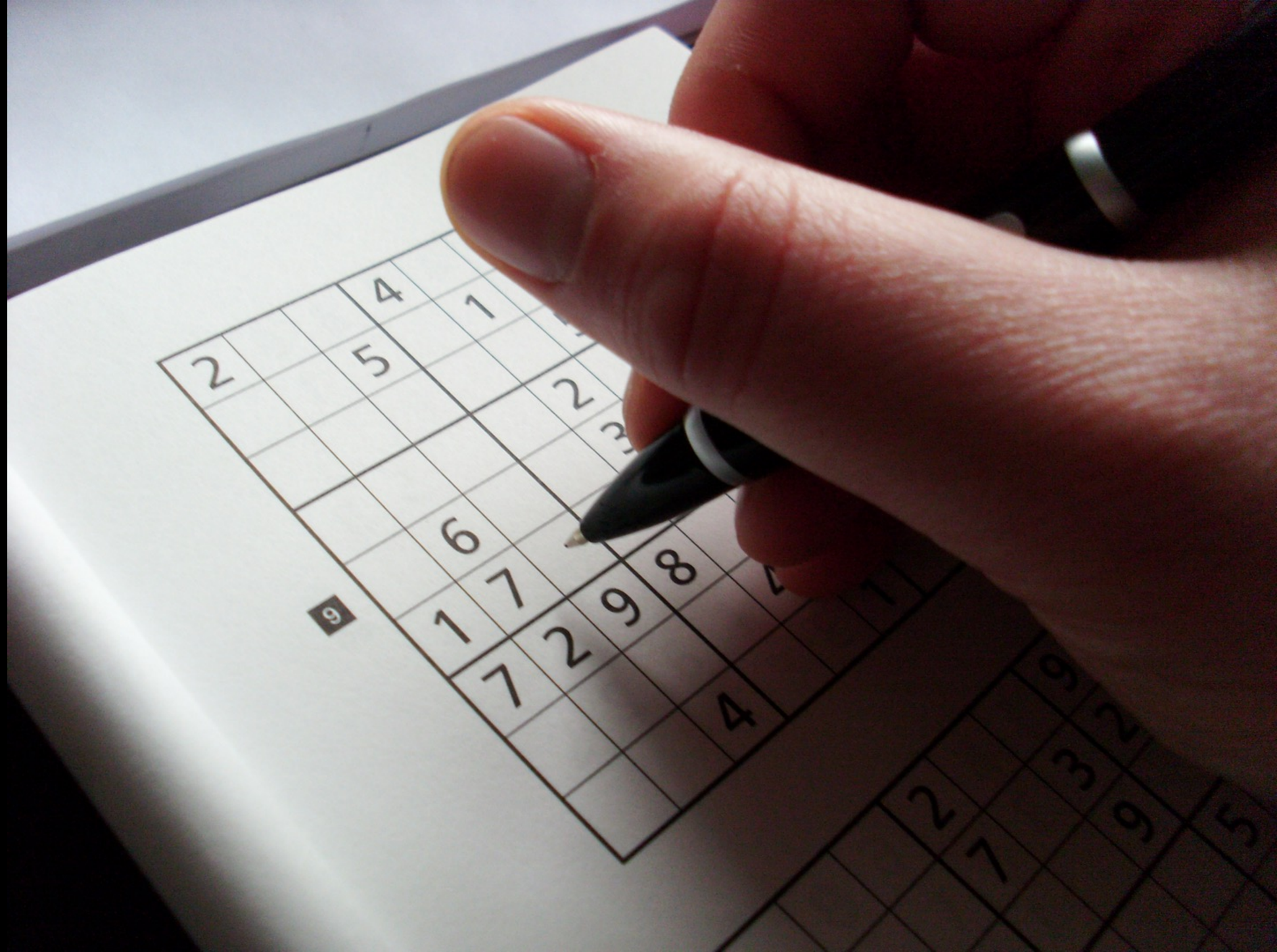
```
      FIBO
----------
         0
         1
         1
         2
         3
         5
         8
        13
        21
        34
        55
```



```
SQL> WITH ORD AS ( SELECT ROWNUM -1 XYZ FROM DUAL CONNECT BY ROWNUM<=100)
  2  , XGEN AS (
  3    SELECT -2.2 + XYZ*0.031 CX, XYZ IX
  4   FROM ORD)
  5  , YGEN AS (
  6    SELECT -1.5 + XYZ*0.031 CY, XYZ IY
  7   FROM ORD)
  8  , Z AS (
  9    SELECT IX, IY,I
 10   FROM XGEN, YGEN
 11  MODEL PARTITION BY (IX, IY)
 12        DIMENSION BY (0 I)
 13        MEASURES (CX, CY
 14          , CX X
 15          , CY Y )
 16        IGNORE NAV
 17        RULES ITERATE (100)
 18          UNTIL (X[ITERATION_NUMBER] * X[ITERATION_NUMBER] +
 19          Y[ITERATION_NUMBER] * Y[ITERATION_NUMBER] > 16) (
 20    CX[ITERATION_NUMBER] = CX[CV()]+CX[CV()-1]
 21        , CY[ITERATION_NUMBER] = CY[CV()]+CY[CV()-1]
 22        , X[ITERATION_NUMBER] = CX[CV()]+X[CV()-1]*X[CV()-1]-Y[CV()-1]*Y[CV()-1]
 23        , Y[ITERATION_NUMBER] = CY[CV()]+Y[CV()-1]*X[CV()-1]*2
 24        )
 25  )
 26  SELECT    string
 27  FROM
```



100 rows selected.


4
1
2
5
2
6
7
8
9
1
7
2
9
4
2
7
9

```
SQL> var s varchar2(81)
SQL>
SQL>
SQL> exec :s := '   56 2  63       2   37 5  173327  14 1  9   6   7    2 38183   '

PL/SQL procedure successfully completed.
```

```
SQL> select substr( s, ( rownum - 1 ) * 9 + 1, 9 ) sudoku
  2  from ( select x, s
  3  from ( select :s s
  4       from dual
  5    )
  6    model
  7   reference xxx on
  8     ( select i, j, r
  9       from dual
 10       model
 11         dimension by ( 1 i, 1 j )
 12         measures ( 1 x, 1 y, 1 r )
 13         rules
 14     ( x[for i from 1 to 81 increment 1, 1] = trunc( ( cv(i) - 1 ) / 9 ) * 9
 15     , y[for i from 1 to 81 increment 1, 1] = mod( cv(i) - 1, 9 ) + 1
 16     , r[for i from 1 to 81 increment 1, for j from 1 to 8 increment 1] = case when
x[ cv(i), 1 ] + cv(j) < cv(i)
 17                     then x[ cv(i), 1 ] + cv(j)
 18                     else x[ cv(i), 1 ] + cv(j) + 1
 19                         end
 20     , r[for i from 1 to 81 increment 1, for j from 9 to 16 increment 1] = case when
y[ cv(i), 1 ] + ( cv(j) - 9 ) * 9 < cv(i)
 21                      then y[ cv(i), 1 ] + ( cv(j) - 9 ) * 9
 22                      else y[ cv(i), 1 ] + ( cv(j) - 8 ) * 9
 23                          end
 24     , r[for i from 1 to 81 increment 1, 17] = case mod( x[ cv(i), 1 ] / 9, 3 )
 25                   when 0 then x[ cv(i), 1 ] + 9
 26                   when 1 then x[ cv(i), 1 ] - 9
```

```
SUDOKU
------------------------------------
174563829
263897514
598214637
956482173
327651498
481739256
612378945
745926381
839145762

9 rows selected.
```

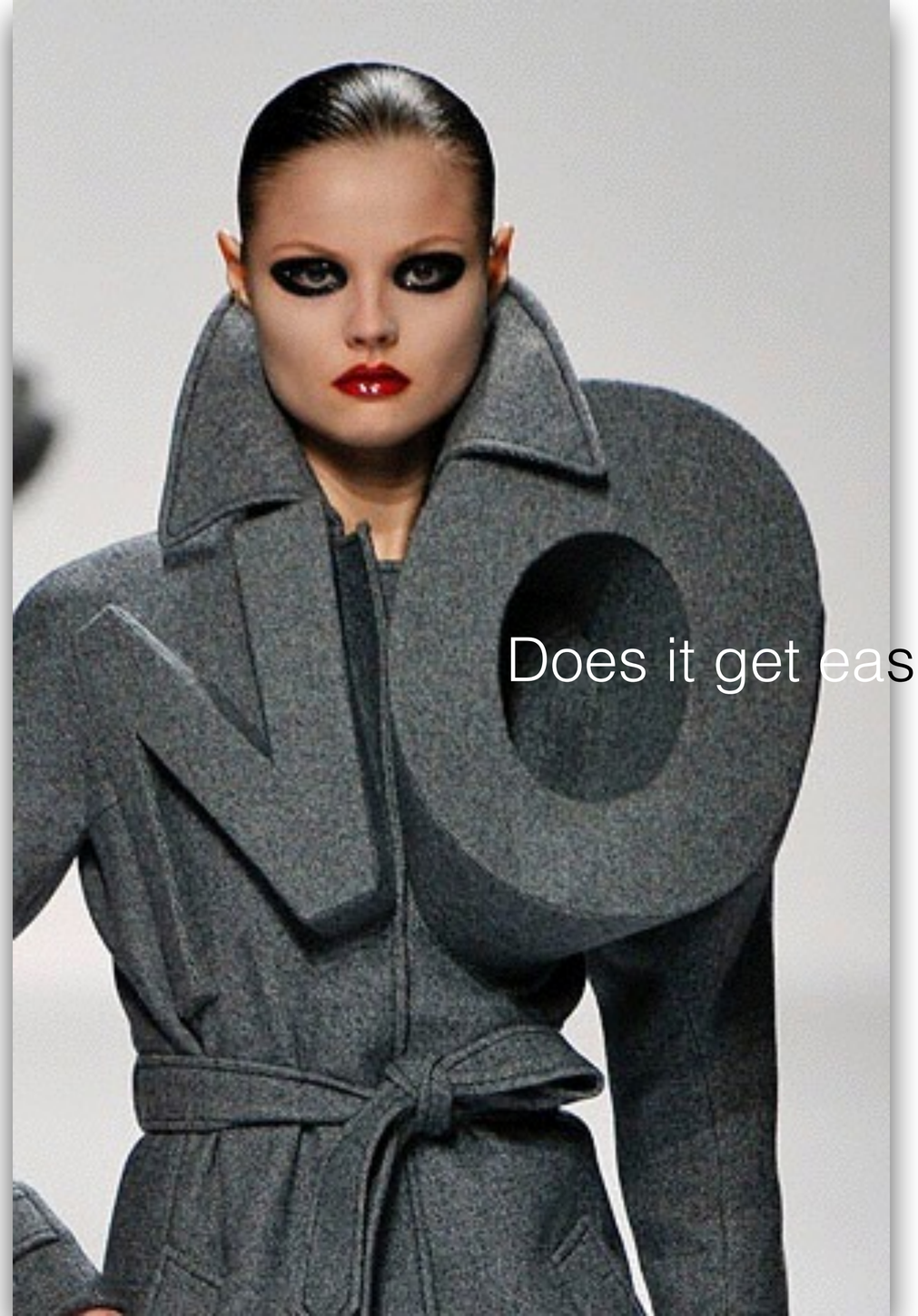
Final Thoughts

Is it worth
getting out of bed for?

es!

for certain Use Cases

Does it get easier over time?



Does it get easier over time?


8 HOUR
8 HOUR
8 HOUR
8 HOUR

match_recognize

Example by Stewart Ashton
http://stewashton.wordpress.com/

STUDY_SITE	CNT
1001	3407
1002	4323
1004	1623
1008	1991
1011	885
1012	11597
1014	1989
1015	5282
1017	2841
1018	5183
1020	6176
1022	2784
1023	25865
1024	3734
1026	137
1028	6005
1029	76
1031	4599
1032	1989
1034	3427
1036	879
1038	6485
1039	3
1040	1105
1041	6460
1042	968
1044	471
1045	3360

Divide in Groups with total CNT <= 65000

STUDY_SITE	CNT
1001	3407
1002	4323
1004	1623
1008	1991
1011	885
1012	11597
1014	1989
1015	5282
1017	2841
1018	5183
1020	6176
1022	2784
1023	25865
1024	3734
1026	137
1028	6005
1029	76
1031	4599
1032	1989
1034	3427
1036	879
1038	6485
1039	3
1040	1105
1041	6460
1042	968
1044	471
1045	3360

FIRST_SITE	LAST_SITE	SUM_CNT
1001	1022	48081
1023	1044	62203
1045	1045	3360

```
SQL> select s first_site
  2      ,max(e) last_site
  3      ,max(sm) sum_cnt
  4     from (
  5     select s, e, cnt, sm from t
  6     model
  7        dimension by (row_number() over (order by study_site) rn)
  8    measures (study_site s, study_site e, cnt, cnt sm)
  9    rules (
 10      sm[ rn  > 1] =
 11        case when sm[cv() - 1] + cnt[cv()] > 65000 or cnt[cv()] > 65000
 12          then cnt[cv()]
 13        else sm[cv() - 1] + cnt[cv()]
 14        end,
 15      s[ rn  > 1] =
 16        case when sm[cv() - 1] + cnt[cv()] > 65000 or cnt[cv()] > 65000
 17          then s[cv()]
 18        else s[cv() - 1]
 19        end
 20    )
 21  )
 22  group by s;
```

FIRST_SITE	LAST_SITE	SUM_CNT
1001	1022	48081
1023	1044	62203
1045	1045	3360

```
SQL> select *
  2    from t
  3    match_recognize (
  4    order by study_site
  5    measures
  6      first(study_site) first_site
  7     ,last(study_site)  last_site
  8     ,sum(cnt)          sum_cnt
  9    pattern (a+)
 10    define a as sum(cnt) <= 65000
 11  );
```

FIRST_SITE	LAST_SITE	SUM_CNT
1001	1022	48081
1023	1044	62203
1045	1045	3360

Excel

Learn about
MATCH_RECOGNIZE
!!!!

Sorry that I lied...

http://finalfashion.ca/wp-content/uploads/2012/07/chanel-dior-couture-fall-2012.jpg

© Austin Kirk
https://flic.kr/p/f4U7nV

© LINGARAJ G J
https://flic.kr/p/4FpWpn

© ukCWCS
https://flic.kr/p/n6k5U6

© Debs (ò_ó)♪
https://flic.kr/p/6eJ3uD

© Sarah
https://flic.kr/p/6yE6wa

http://www.efteling.com/

Viktor and Rolf

Numbers by Apple

Excel by MicroSoft

Calc by Apache OpenOffice

© moffoys
https://flic.kr/p/6pZH62

© Anua22a
https://flic.kr/p/4xZrCf

© Tim Psych
https://flic.kr/p/4hyGoc

allAPEX

Alex Nuijten

www.allAPEX.nl alex@allAPEX.nl @alexnuijten nuijten.blogspot.com