第九章

Model 子句

通过 Model 子句，可以使用SQL语句中的一些很强大的功能，如聚合，并行，以及多维、多变量分析。
传统的SQL语句与Model子句的最核心区别在于Model子句支持跨行引用，多单元格引用以及单元格聚合。

9.1 电子表格

eg: 当前库存为上周库存的综合加上本周收到的数量再减去本周卖出的数量。

9.2 通过 Model 子句进行跨行引用

9.2.1 示例数据

```
drop table sales_fact;
create table sales_fact as
select country_name country, country_subRegion region, prod_name product, calendar_year year, calendar_week_number week,
sum(amount_sold) sale, sum(amount_sold*(case when mod(rownum,10)=0 then 1.4 when mod(rownum,5)=0 then 0.6 when mod(rownum,2)=0
then 0.9 when mod(rownum,2)=1 then 1.2 else 1 end)) receipts from sales, times, customers, countries, products where
sales.time_id = times.time_id and slaes.prod_id = products.prod_id and sales.cust_id = customers.cust_id and customers.country_id
= countries.country_id group by country_name, country_subRegion, prod_name, calendar_year, calendar_week_number;
```

9.2.2 剖析 Model 子句

eg: 使用Model子句进行库存公式计算

```
col product format a30
col country format a10
col region format a10
col year format 999
col week format 99
col sale format 999999
set lines 120 pages 100
select product, country, year, week, inventory, sale, receipts
from sales_fact
model  return updated rows
where country in ('Australia') and product = 'Xtend Memory'
partition by (product, country)
dimension by (year, week)
measures(0 inventory, sale, receipts)
rules automatic order (
inventory[year,week] = nvl(inventory [cv(year), cv(week)-1],0) - sale[cv(year), cv(week)] + receipts [cv(year), cv(week)]
)
order by product, country, year, week;
```

9.2.3 规则

eg:

```
rules (
    inventory [2001, 3] = nvl(inventory [cv(year), cv(week)-1], 0)
    - sale [cv(year), cv(week) ] +
    +receipts [cv(year), cv(week) ]
)
```

9.3 位置和符号引用

eg: 使用位置引用来初始化2002年的值——UPSERT

```
select product, country, year, week, inventory, sale, receipts
from sales_fact
where country in ('Australia') and product = 'Xtend Memory'
model return updated rows
paartition by (product, country)
dimension by (year, week)
```

```
measures (0 inventory, sale, receipts)
rules automatic order (
    inventory [year, week ] =
        nvl(inventory [cv(year), cv(week)-1],0)
        -sale[cv(year), cv(week)] +
        +receipts [cv(year), cv(week)],
    sale [2002,1] = 0,
    receipts [2002,1] =0
)
order by product, country, year, week;
```

9.3.2 符号标记

eg: 符号引用——UPDATE

```
select product, country, year, week, sale
from sales_fact
where country in (’Australia’) and product = ’Xtend Memory’
model return updated rows
partition by (product, country)
dimension by (year, week)
measures(sale)
rules(
    sale [year in (2000,2001), week in (1,52,53)] order by year, week
        =sale [cv(year), cv(week)] * 1.10
)
order by product, country, year, week;
```

9.3.3 FOR 循环

语法：

```
for dimension from <value1> to <value2> [increment | decrement] <value3>
```

eg:

```
select product, country, year, week, inventory, sale, receipts
from sales_fact
where country in (’Australia’) and product = ’Xtend Memory’
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules automatic order (
    inventory [year, week] =
        nvl(inventory [cv(year), cv(week)-1],0)
        - sale[cv(year), cv(week)] +
        + receipts [cv(year), cv(week)],
        sale [2002, for week from 1 to 53 increment1] = 0,
        receipts [2002,for week from 1 to 53 increment1] =0
)
order by product, country, year, week;
```

9.4 返回更新后的行

eg: 没有RETURN UPDATED ROWS 的SQL语句

```
select product, country, year, week, sale
from sales_fact
where country in (’Australia’) and product = ’Xtend Memory’
model
partition by (product, country)
dimension by (year, week)
measures (sale)
rules (
    sale [year in (2000,2001), week in (1,52,53)] order by year, week
```

```
        = sale [cv(year), cv(week)] * 1.10
)
order by product, country, year, week;
```

eg: RETURN UPDATED ROWS 与 UPSERT

```
select product, country, year, week, sale
from sales_fact
where country in ('Australia') and product = 'Xtend Memory'
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (sale)
rules (
    sale(2002,1] = 0
)
order by product, country, year, week;
```

9.5 求解顺序

9.5.1 行求解顺序

eg: 单元格级的求解顺序

```
select product, country, year, week, inventory, sale, receipts
from sales_fact
where country in ('Australia')
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules (
    inventory [year, week] order by year, week =
        nvl(inventory [cv(year), cv(week)-1],0)
        - sale[cv(year), cv(week)] +
        + receipts [cv(year), cv(week) ]
)
order by product, country, year, week;
```

eg: 使用DESC关键字的求解顺序

```
select product, country, year, week, inventory, sale, receipts
from sales_fact
where country in ('Australia') and product in ('Xtend Memory')
model return updated rows
partition by (product, ocuntry)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules (
    inventory [year, week] order by year, week desc =
        nvl(inventory [cv(year), cv(week)-1],0)
        - sale[cv(year), cv(week)] +
        + receipts [cv(year), cv(week)]
)
order by product, country, year, week;
```

9.5.2 规则求解顺序

eg: 规则求值顺序——顺序求值

```
select * from (
seelect product, country, year, week, inventory, sale, receipts
from sales_fact
where country in ('Australia') and product in ('Xtend Memory')
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules sequential order (
    inventory [year, week]] order by year, week =
        nvl(inventory cv(year), cv(week)-1],0)
        - sale[cv(year), cv(week)] +
        + receipts [cv(year), cv(week)],
    receipts [year in (2000,2001), week in (51, 52, 53)]
        order by year, week
        =receipts [cv(year), cv(week)] * 10
    )
    order  by product, country, year, week
)where week >50;
```

eg: 规则求解顺序——自动顺序

```
...
9 rules automatic order (
...
```

9.6 聚合

eg:

```
select product, country, year, week, inventory, avg_infentory, max_sale
from sales_fact
where country in ('Australia) and product = 'Xtend Memory'
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, 0 avg_inventory, 0 max_sale, sale, receipts)
rules automatic order (
    inventory [year, week] =
        nvl(inventory [cv(year), cv(week)-1], 0)
        - sale[cv(year),  cv(week)] +
        + receipts [cv(year), cv(week) ],
        avg_inventory[year, any]=avg(invnetory)[cv(year),week],
        max_sale[year,any] = max(sale)[cv(year),week]
)
order by product, country, year, week;
```

9.7 迭代

语法：

```
[iterate (n) [until <condition] ] (<cell_assignment> = <expression> ... )
```

9.7.1

eg:

```
select year, week, sale, sale_list
from sales_fact
where country in ('Australia') and product = 'Xtend Memory'
model return updated rows partition by (product, country)
dimension by (year, week)
measures(cast(' ' as varchar2(50)) sale_list, sale)
rules iterate(5) (
    sale_list [year, week] order by year, week =
        sale [cv(year), cviteration_number +2] ||
        case when iteration_number=0 then '' else ',' end ||
        sale_list [cv(year), cv(week)]
)
order by year, week;
```

9.7.2 PRESENTV 与空值

语法：

```
presentv (cell_reference, expr1, expr2)
```

Note: 如果cell_reference引用了一个存在的单元格，那么presentv函数返回expr1；
如果cell_reference引用了 一个不存在的但与昂则返回expr2.

eg: 迭代和presentv

```
select year, week, sale, sale_list
from sales_fact
where country in ('Australia') and product = 'Xtend Memory'
model return updated rows partition by product, country)
dimension by (year, week)
measures (cast(' ' as varchar2(120)) sale_list, sale, 0 tmp)
rules iterate(5) (
    sale_list [year, week] order by year, week =
        presentv (sale [cv(year), cv(week)-iteration_number + 2],
            sale[cv(year), cv(week)-iteration_number + 2]
                case when iteration_number=0 then ' ' else ', ' end ||
                sale_list [cv(year), cv(week)])
)
order by year, week;
```

Note:
presentnnv 函数与 presentv 函数类似，但提供了进一步区分所引用的是不存在的单元格还是存在的值为空的单元格的能力。
语法：

```
presentnnv (cell_reference, expr1, expr2)
```

Note: 如果第1个参数cell_reference引用了一个存在的但与昂并且单元格不含控制，那么返回expr1, 否则返回expr2.

9.8 查找表（or 参考表）

eg: 参考 Model

```
select year, week, sale, prod_lsit_proce
from sales_fact
where country in ('Australia') and product = 'Xtend Memory'
model return updated rows
reference ref_prod on
    (select prod_name, max(prod_list_price) prod_list_price from products
    group by prod_name)
    dimension by (prod_name)
    measures (prod_list_price)
main main_section
    partition by (product, country)
    dimension by (year, week)
    measures (sale, receipts, 0 prod_list_price)
    rules (
        prod_list_price[year,week] order by year, week =
            ref_prod.prod_list_price[cv(product)]
    )
order by year, week;
```

eg: 更多的查找表

```
select year, week, sale, prod_list_price, iso_code
from sales_fact
where country in ('Australia') and product = 'Xtend Memory'
model return updated rows
reference ref_prod on
(select prod_name, max(prod_list_price) prod_list_price from
    products group by prod_name)
    dimension by (prod_name)
    measures (prod_list_price)
    reference ref_country on
        (select country_name, country_iso_code from countries)
        dimension by (country_name)
        measures (country_iso_code)
    main main_section
        partition by (product, country)
        dimension by (year, week)
        measures (sale, receipts, 0 prod_list_price, cast(' ' as varchar2(5)) iso_code)
        rules (
            prod_list_price[year,week] order by year, week =
                ref_prod.prod_list_price[cv(product)],
            iso_code[year,week] order by year, week =
                ref_country.country_iso_code [cv(country)]
        )
order by year, week;
```

9.9 空值

eg:

```
select product, country, year, week, sale
from sales_fact
where country in ('Australia') and product = 'Xtend Memory'
model keep nav return updated rows
partition by (product, country)
dimension by (year, week)
measures(sale)
rules sequential order(
    sale[2001,1] order by year, week = sale[2001,1],
    sale[2002,1] order by year, week = sale[2001,1] + sale[2002,1]
    )
order by product, country,  year, week;
```

eg: 忽略NAV

```
select product, country, year, week, sale
from sales_fact
where country in ('Australia') and product = 'Xtend Memory'
model ignore nav return updated rows
partition by (product, country)
dimension by (year, week)
measures(sale)
rules sequential order (
    sale[2001,1] order by year, week = sale[2001,1],
    sale[2001,1] order by year, week = sale[2001,1] + sale[2002,1]
    )
order by product, country, year, week;
```

9.10 使用 Model 子句进行性能调优

在Model子句中，规则求解是关键的步骤。规则求解可以使用下面5中算法之一：acyclic, aacyclic fast, cyclic, ordered, ordered fast.

eg: 自动排序与acyclic

```
select product, country, year, week, inventory, sale, receipts
from sales_fact
where country in ('Australia') and product = 'Xtend Memory'
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules automatic order (
    inventory [year, week] order by year, week =
        nvl(inventory [cv(year), cv(week)-1],0)
        - sale[cv(year), cv(week)] +
        + receipts [cv(year), cv(week) ]
)
```

```
order by product, country, year, week;
```

eg: 自动排序与 acyclic fast

```
select distinct product, country, year, week, sale_first_week
from sales_fact
where country in ('Australia') and products='Xtend Memory'
model return updated rows
partition by (product, country)
dimension by (year, week)
measures ( o sale_first_week, sale)
rules automatic order (
    sale_first_week [2000,1] = 0.12*sale [2000,1]
)
order by product, country, year, week;
```

eg: 自动排序与 cyclic

```
select product, country, year, week, inventory, sale, receipts
from sales_fact
where country in ('Australia') and product = 'Xtend Memory'
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules automatic order (
    inventory [year, week] =
        nvl(inventory[cv(year),cv(week)-1],0)
        - sale[cv(year),cv(week)] +
        + receipts [cv(year), cv(week) ]
)
order by product, country, year, week;
```

eg: sequentia 顺序

```
select product, country, year, week, inventory, sale, receipts
from sales_fact
where country in ('Australia') and product = 'Xtend Memory'
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules sequential order (
    inventory [year, week] order by year, week =
        nvl(inventory [cv(year), cv(week)-1],0)
        - sale[cv(year),cv(week)] +
        + receipts [cv(year), cv(week) ]
)
order by product, country, year, week;
```

9.10.2 谓语前推

eg: 谓语前推

```
select * from (
select product, country, year, week, inventory, sale, receipts
from sales_fact
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules automatic order (
    inventory [year, week] =
        nvl(inventory [cv(year), cv(week)-1],0)
        - sale[cv(year), cv(week)] +
        + receipts [cv(year), cv(week)]
    )
) where country in ('Australia') and product = 'Xtend Memory'
order by product, country, year, week;
```

eg: 未进行谓语前推

```
select * from (
select product, country, year, week, inventory, sale, receipts
from sales_fact
mod el return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules automatic order (
    inventory [year, week] =
        nvl(inventory [cv(year), cv(week)-1],0)
        - sale[cv(year), cv(week)] +
        + receipts [cv(year), cv(week)]
    )
) where year=2000
order by product, country, year, week;
```

9.10.3 物化视图

eg: 物化视图与查询重写

```
create materialized view mv_model_inventory
enable query rewrite as
select product, country, year, week, inventory, sale, receipts
from sales_fact
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules sequential order (
    inventory [year, week] order by year, week =
        nvl(inventory [cv(year), cv(week)-1], 0)
        - sale[cv(year), cv(week)] +
        + receipts [cv(year), cv(week)]
    )
/
select * from (
select product, country, year, week, inventory, sale, receipts
from sales_fact
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules sequential order (
    inventory [year, week] order by year, week =
        nvl(inventory [cv(year), cv(week)-1], 0)
        - sale[cv(year), cv(week)] +
        + receipts [cv(year), cv(week)]
    )
)
where country in ('Australia') and product = 'Xtend Memory'
order by product, country, year, week;
```

9.10.4 并行

eg: Model 与并行查询

```
select /*+ parallel(sf 4) */
```

```
product, country, year, week, inventory, sale, receipts
from sales_fact sf
where country in ('Australia') and product = 'Xtend Memory'
model return updated rows
partition by (product, country)
dimension by (year, week)
measures(0 inventory, sale, receipts)
rules automatic order (
    inventory [year, week] order by year, week =
        nvl(inventory [cv(year), cv(week)-1], 0)
        -sale[cv(year), cv(week)] +
        + receipts [cv(year), cv(week)]
);
```

9.10.5 Model 子句执行中的分区

eg: 分区剪裁

```
select * from (
select product, country, year, week, inventory, sale, receipts
from sales_fact_part sf
model return updated rows partition by (year, country)
dimension by (product, week)
measures (0 inventory, sale, receipts)
rules automatic order (
    inventory [product, week] order by product, week =
        nvl(inventory [cv(product), cv(week)-1], 0)
        - sale[cv(product), cv(week) ] +
        + receipts [cv(product), cv(week)]
    )
) where year=2000 and country='Australia' and product = 'Xtend Memory';
```

eg: 不进行分区剪裁

```
select * from (
select product, country, year, week, inventory, sale, receipts
from sales_fact_part sf
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules automatic order (
    inventory [year, week] order by year, week =
        nvl(inventory [cv(year), cv(week)-1], 0)
        - sale[cv(year), cv(week)] +
        + receipts [cv(year), cv(week)]
    )
) where year = 2000 and country = 'Australia' and product = 'Xtend Memory';
```

9.10.6 索引

eg: 记住SQL数据访问的索引

```
create index sales_fact_part_i1 on sales_fact_part (country, product);
select * from (
select product, country, year, week, inventory, sale, receipts
from sales_fact_part sf
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules automatic order (
    inventory [year, week] order by year, week =
        nvl(inventory [cv(year), cv(week)-1], 0)
        - sale[cv(year), cv(week)] +
        + receipts [cv(year), cv(week)]
    )
) where year = 2000 and country='Australia' and product = 'Xtend Memory';
```

9.11 子查询因子化

eg: 记住更多SQL数据访问索引的内容

```
with t1 as (
select product, country, year, week, inventory, sale, receipts
from sales_fact sf
where country in ('Australia') and product = 'Xtend Memory'
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (0 inventory, sale, receipts)
rules automatic order (
inventory [year, week] order by year, week =
    nvl(inventory [cv(year), cv(week)-1], 0)
    -sale[cv(year), cv(week)] +
    +receipts [cv(year), cv(week)]
)
)
select product, country, year, week, inventory, sale, receipts, prev_sale
from t1
model return updated rows
partition by (product, country)
dimension by (year, week)
measures (inventory, sale, receipts, 0 prev_sale)
rule sequential order (
    prev_sale [year, week] order by year, week =
        nvl(sale [cv(year)-1, cv(week)], 0)
)
order by 1,2,3,4;
```