# 6.3 聚合来自树的值

# 对保存于叶节点的值做聚合

**表UNITS**

| ID | NAME | COMMANDER |
|---|---|---|
| 1 | III Corps | Général de Division Dominique Vandamme |
| 2 | 8th Infantry Division | Général de Division Baron Etienne-Nicolas Lefol |
| 3 | 1st Brigade | Général de Brigade Billard |
| 4 | 2nd Brigade | Général de Brigade Baron Corsin |
| 5 | 10th Infantry Division | Général de Division Baron Pierre-Joseph Habert |
| 6 | 1st Brigade | Général de Brigade Baron Gengoult |
| 7 | 2nd Brigade | Général de Brigade Baron Dupeyroux |
| 8 | 11th Infantry Division | Général de Division Baron Pierre Berthézène |
| 9 | 1st Brigade | Général de Brigade Baron Dufour |
| 10 | 2nd Brigade | Général de Brigade Baron Logarde |
| 11 | 3rd Light Cavalry Division | Général de Division Baron Jean-Simon Domont |
| 12 | 1st Brigade | Général de Brigade Baron Dommanget |
| 13 | 2nd Brigade | Général de Brigade Baron Vinot |
| 14 | Reserve Artillery | Général de Division Baron Jérôme Doguereau |

**UNIT_LINKS_ADJACENCY**

| ID | PARENT_ID |
|---|---|
| 2 | 1 |
| 3 | 2 |
| 4 | 2 |
| 5 | 1 |
| 6 | 5 |
| 7 | 5 |
| 8 | 1 |
| 9 | 8 |
| 10 | 8 |
| 11 | 1 |
| 12 | 11 |
| 13 | 11 |
| 14 | 1 |

**UNIT_LINKS_PATH**

| ID | PATH |
|---|---|
| 1 | 1 |
| 2 | 1.1 |
| 3 | 1.1.1 |
| 4 | 1.1.2 |
| 5 | 1.2 |
| 6 | 1.2.1 |
| 7 | 1.2.2 |
| 8 | 1.3 |
| 9 | 1.3.1 |
| 10 | 1.3.2 |
| 11 | 1.4 |
| 12 | 1.4.1 |
| 13 | 1.4.2 |
| 14 | 1.5 |

**UNIT_STRENGTH**

| ID | MEN |
|---|---|
| 3 | 2952 |
| 4 | 2107 |
| 6 | 2761 |
| 7 | 2823 |
| 9 | 2488 |
| 10 | 2050 |
| 12 | 699 |
| 13 | 318 |
| 14 | 152 |

# 计算每一层的人数（邻接模型）

计算第三军的总人数：

```
select sum(men)
   from unit_strength
   where id in (select id
            from unit_links_adjacency
            connect by prior id = parent_id
            start with parent_id = 1)
```

计算每一层的人数

```
select u.name,
      u.commander,
      (select sum(men)
       from unit_strength
       where id in (select id
                from unit_links_adjacency
                connect by parent_id = prior id
                start with parent_id = u.id)
         or id = u.id) men
from units u
```

**Connect by 的过程化本质带来巨大的障碍**

# 计算每一层的人数（物化路径）

SQL> select * from exploded_links_path;

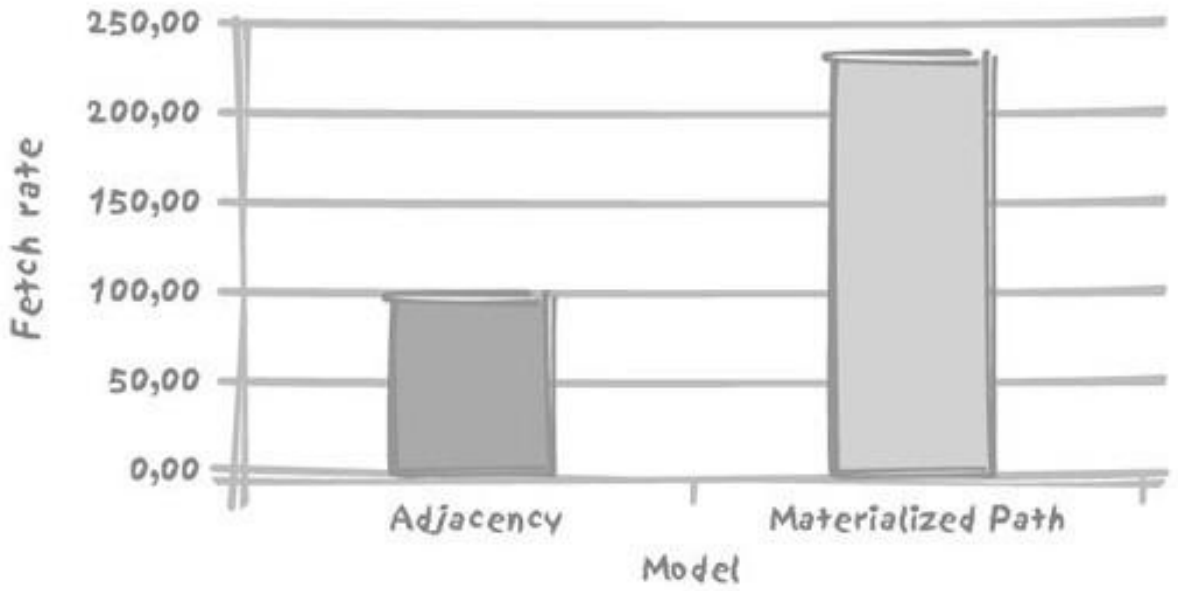
| ID | ANCESTOR | DEPTH |
|---|---|---|
| 14 | 1 | 1 |
| 13 | 1 | 2 |
| 12 | 1 | 2 |
| 11 | 1 | 1 |
| 10 | 1 | 2 |
| 9 | 1 | 2 |
| 8 | 1 | 1 |
| 7 | 1 | 2 |
| 6 | 1 | 2 |
| 5 | 1 | 1 |
| 4 | 1 | 2 |
| 3 | 1 | 2 |
| 2 | 1 | 1 |
| 4 | 2 | 1 |
| 3 | 2 | 1 |
| 7 | 5 | 1 |
| 6 | 5 | 1 |
| 10 | 8 | 1 |
| 9 | 8 | 1 |
| 13 | 11 | 1 |
| 12 | 11 | 1 |

```
select u.name, u.commander, sum(s.men) men
  from units u,
       exploded_links_path el,
       unit_strength s
  where u.id = el.ancestor
    and el.id = s.id
  group by u.name, u.commander
```

| *NAME* | *COMMANDER* | *MEN* |
|---|---|---|
| *III Corps* | *Général de Division Dominique Vandamme* | *16350* |
| *8th Infantry Division* | *Général de Division Baron Etienne-Nicolas Lefol* | *5059* |
| *10th Infantry Division* | *Général de Division Baron Pierre Joseph Habert* | *5584* |
| *11th Infantry Division* | *Général de Division Baron Pierre Berthézène* | *4538* |
| *3rd Light Cavalry Division* | *Général de Division Baron Jean-Simon Domont* | *1017* |

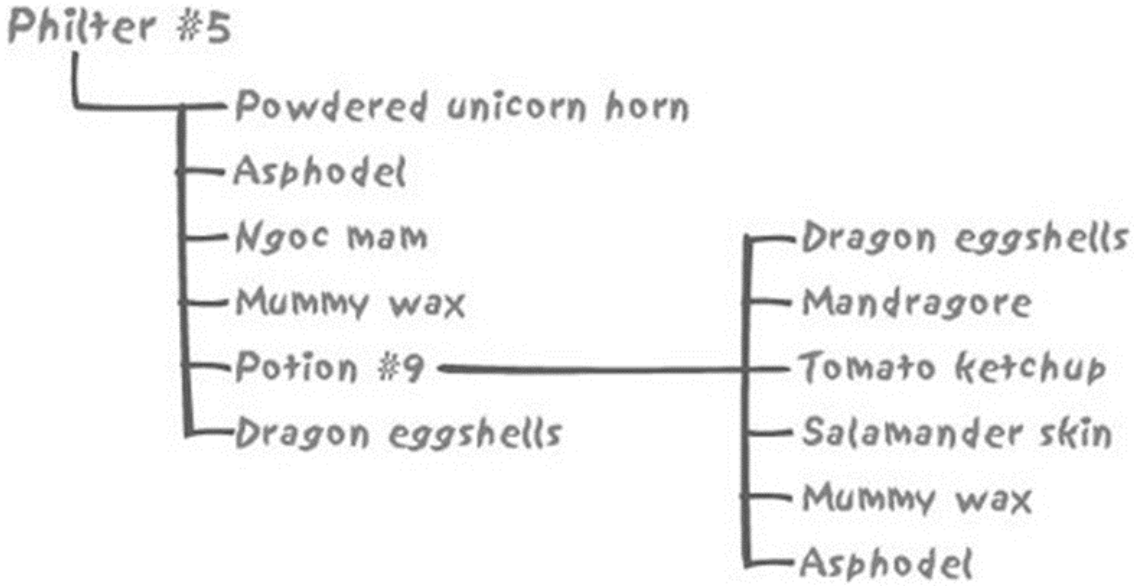
# 计算每一层的人数

- 执行查询5000次，比较单位时间返回的记录数



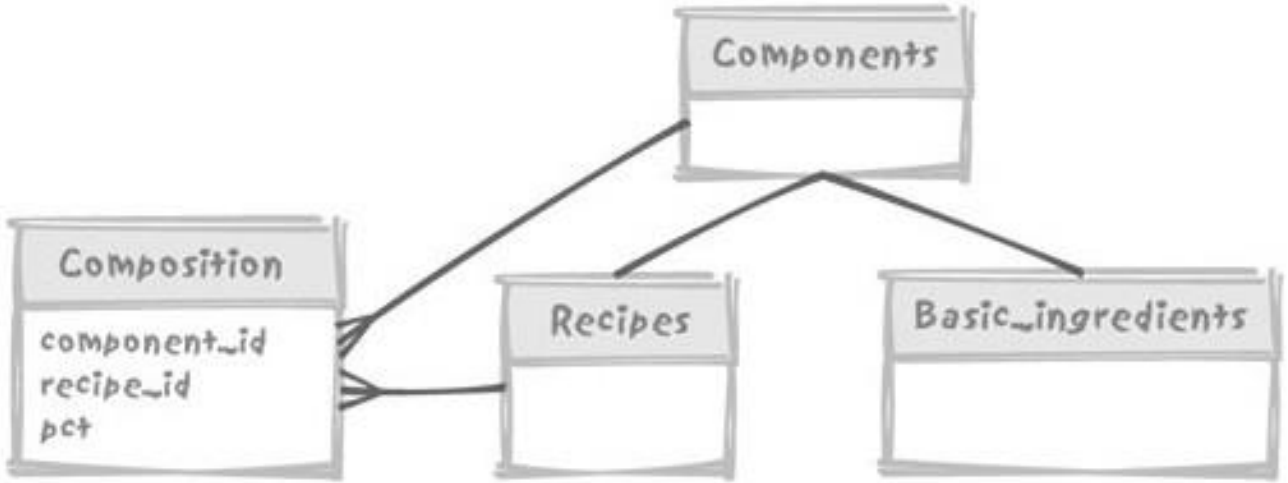
# 散布在各层的百分比

- 假设我们经营魔药。每种魔药由多种成分（ingredient）组成，处方（recipe）列出成分及百分比。处方可以共享某种“基础魔药”，以复合成分（compound ingredient）的形式表示。



# 散布在各层的百分比

- 某一种可以选择的建模方法
- Components表为通用类型
- 它有recipes和basic_ingredients两种子类型
- Composition表保存处方成分（可以是处方或基本成分及其数量）



# 散布在各层的百分比

```
SQL> select connect_by_root recipe_id root_recipe,
  2         recipe_id,
  3         prior pct,
  4         pct
  5         component_id
  6  from composition
  7  connect by recipe_id = prior component_id
  8  /
```

| ROOT_RECIPE | RECIPE_ID | PRIORPCT | PCT | COMPONENT_ID |
|---|---|---|---|---|
| 14 | 14 | | 5 | 3 |
| 14 | 14 | | 20 | 7 |
| 14 | 14 | | 15 | 8 |
| 14 | 14 | | 30 | 9 |
| 14 | 14 | | 20 | 10 |
| 14 | 14 | | 10 | 2 |
| 15 | 15 | | 30 | 14 |
| 15 | 14 | 30 | 5 | 3 |
| 15 | 14 | 30 | 20 | 7 |
| 15 | 14 | 30 | 15 | 8 |
| 15 | 14 | 30 | 30 | 9 |

...

```
with recursive_composition(actual_pct, component_id)
  as (select a.pct,
           a.component_id
      from composition a,
          components b
      where b.component_id = a.recipe_id
       and b.component_name = 'Philter #5'
      union all
      select parent.pct * child.pct,
           child.component_id
      from recursive_composition parent,
          composition child
      where child.recipe_id = parent.component_id)


select x.component_name, sum(y.actual_pct)
  from recursive_composition y,
       components x
  where x.component_id = y.component_id
    and x.component_type = 'I'
  group by x.component_name
```

# 树状结构的问题

- 本章的方法，在数据量很少的情况下效果令人满意
- 对大数据量的处理“像老爷车一样慢”

- 同样可以采用非规范化模型、或基于触发器的扁平化数据模型。

- 不建议对关系模型“屡遭诟病的缓慢本性”反规范化，这很容易遮掩程序设计中的问题。
- 不过，SQL确实缺乏处理树结构的强大的、可伸缩的手段。

# End

下一讲，进入非规范化模型