# Devived table (select 块)

在复杂的sql操作中，经常需要用到select块，用来代替实际的表或者view。

## 演示数据库

演示数据库是toturial中的sample\_mkt表

## Select中用来代替真实table的derived table(临时表)

### Select中使用Devived table

select datetime, h from

(

select datetime, code, open as o, high as h, low, close from sample\_mkt

) as test

注意在select中的derived table要用alias，这可能是MySql中设定。

### Select中使用Derived table，带where

select datetime, h as haha from

(

select datetime, code, open as o, high as h, low, close from sample\_mkt

) as test

where h >= 11 /\* where中是原表的列名，不是alias的列名

### Select中使用Devived table，带order by

select datetime, h as haha from

(

select datetime, code, open as o, high as h, low, close from sample\_mkt

) as test

where h >= 11

order by haha asc /\* order by是针对alias，不是原表

## insert into derived table 代替values

insert into sample\_insert

(select \* from sample\_mkt where high > 11) –这里不用alias

## Update @ todo

## create

CREATE TABLE 新表

AS

SELECT \* FROM 旧表

create table new\_mkt

as

select \* from sample\_mkt where high > 11

## select块返回标量的情况，作为标量

SELECT site\_id, count FROM access\_log  
WHERE count > (SELECT AVG(count) FROM access\_log);

# 函数

分为aggregate函数和scalar函数，

## 在select中列的计算

### 如果列中只有一个聚合函数，返回标量

SELECT AVG(count) AS CountAverage FROM access\_log;

SELECT site\_id, count FROM access\_log  
WHERE count > (SELECT AVG(count) FROM access\_log);

### 如果列中有聚合函数和原本列名，还是返回一行

SELECT site\_id, SUM(access\_log.count) AS nums FROM access\_log

向量返回第一个，

如果有group by，那么返回该组中第一个。

# on where groupby having字句顺序和执行顺序

**SELECT (列名区) FROM (目标表) WHERE (筛选区) GROUP BY (分组区) HAVING (组后筛选) ORDER BY (排序)**

1. **join on字句作用在目标表里面，首先执行，返回目标表；**
2. **执行where xx对目标表数据做筛选，返回第1个结果集，where里面不能有聚合函数；**
3. **对第1个结果集的数据进行进行分组，返回第2个结果集，group by也不能有聚合函数**
4. **针对第2个结果集中的每1组数据执行select xx，有几组就执行几次，返回第3个结果集；**
5. **针对第3个结集执行having xx进行筛选，返回第4个结果集，having必有group by**
6. **针对第4个结果集排序**

## groupby的例子

SELECT site\_id, SUM(access\_log.count) AS nums

FROM access\_log GROUP BY site\_id;

多表连接

SELECT Websites.name,COUNT(access\_log.aid) AS nums FROM access\_log  
LEFT JOIN Websites  
ON access\_log.site\_id=Websites.id  
GROUP BY Websites.name;

## having的例子

SELECT Websites.name, Websites.url, SUM(access\_log.count) AS nums FROM (access\_log  
INNER JOIN Websites  
ON access\_log.site\_id=Websites.id)  
GROUP BY Websites.name  
HAVING SUM(access\_log.count) > 200;

### having和where混合

SELECT Websites.name, SUM(access\_log.count) AS nums FROM Websites  
INNER JOIN access\_log  
ON Websites.id=access\_log.site\_id  
WHERE Websites.alexa < 200   
GROUP BY Websites.name  
HAVING SUM(access\_log.count) > 200;