# SQL语句与OB语句转化示例

By 晓楚([xiaochu.yh@taobao.com](mailto:xiaochu.yh@taobao.com))

本文给出了将SQL语句转化成OB C++接口的例子，着重阐述了groupby接口的用法。

首先建立了一个SQL表格，然后针对这个表格进行一系列的查询。表格记录了Los、San、Bos三个商店的每日销售汇总状况，其中sales表示商店当日销售总额，counts表示商店当日售出的总物品数。

下面所有查询结果都是在mysql中实验得到的。表格信息如下：

mysql> select \* from store\_information;

+------------+-------+--------+------------+

| store\_name | sales | counts | sale\_date |

+------------+-------+--------+------------+

| Los | 1500 | 1 | 2000-01-03 |

| Los | 300 | 1 | 2000-01-01 |

| San | 250 | 2 | 2000-01-01 |

| Bos | 700 | 3 | 2000-01-05 |

+------------+-------+--------+------------+

ObScanParam scan\_param;

scan\_param.add\_column("store\_name");

scan\_param.add\_column("sales");

scan\_param.add\_column("counts");

scan\_param.add\_column("sale\_date");

查询每个商店的总营业额

select store\_name, sum(sales)

from store\_information

group by store\_name;

+------------+------------+

| store\_name | sum(sales) |

+------------+------------+

| Bos | 700 |

| Los | 1800 |

| San | 250 |

+------------+------------+

ObGroupByParam groupby\_param;

groupby\_param.add\_groupby\_column("store\_name");

groupby\_param.add\_aggregate\_column("sales", "s", SUM);

ObScanParam scan\_param;

scan\_param.add\_column("store\_name");

scan\_param.set\_group\_by\_param(groupby\_param);

查询总营业额大于1500的商店

select store\_name, sum(sales)

from store\_information

group by store\_name

having sum(sales) > 1500;

+------------+------------+

| store\_name | sum(sales) |

+------------+------------+

| Los | 1800 |

+------------+------------+

ObGroupByParam groupby\_param;

groupby\_param.add\_groupby\_colum("store\_name");

groupby\_param.add\_aggregate\_column("sales", "s", SUM);

groupby\_param.add\_filter("s > 1500");

ObScanParam scan\_param;

scan\_param.add\_column("store\_name");

scan\_param.set\_group\_by\_param(groupby\_param);

查询售出总物件数等于2的的商店及其总营业额

select store\_name, sum(sales)

from store\_information

group by store\_name

having sum(counts) = 2;

+------------+------------+

| store\_name | sum(sales) |

+------------+------------+

| Los | 1800 |

| San | 250 |

+------------+------------+

ObGroupByParam groupby\_param;

groupby\_param.add\_groupby\_colum("store\_name");

groupby\_param.add\_aggregate\_column("sales", "s", SUM);

//false参数表示查询但并不作为结果返回

groupby\_param.add\_aggregate\_column("counts", "c", SUM, false);

groupby\_param.add\_filter("c = 2");

ObScanParam scan\_param;

scan\_param.add\_column("store\_name");

scan\_param.set\_group\_by\_param(groupby\_param);

查询售出总物件数等于2的的商店及其平均每个物件的售价

select store\_name, sum(sales)/sum(counts) as sale

from store\_information

group by store\_name

having sum(counts) = 2;

+------------+----------+

| store\_name | sale |

+------------+----------+

| Los | 900.0000 |

| San | 125.0000 |

+------------+----------+

ObGroupByParam groupby\_param;

groupby\_param.add\_groupby\_colum("store\_name");

// add\_aggregate\_column只负责聚合运算

groupby\_param.add\_aggregate\_column("sales", "s", SUM, false); groupby\_param.add\_aggregate\_column("counts", "c", SUM, false);

// add\_column只负责数学运算，其参数可以是聚合运算的as\_name

groupby\_param.add\_column("s/c", "sale");

groupby\_param.add\_filter("c = 2");

ObScanParam scan\_param;

scan\_param.add\_column("store\_name");

scan\_param.set\_group\_by\_param(groupby\_param);

查询2000-01-03前售出总物件数等于2的的商店及其平均每个物件的售价

select store\_name, sum(sales)/sum(counts) as sale

from store\_information where sale\_date < '2000-01-03'

group by store\_name

having sum(counts) = 2;

+------------+----------+

| store\_name | sale |

+------------+----------+

| San | 125.0000 |

+------------+----------+

ObGroupByParam groupby\_param;

groupby\_param.add\_groupby\_colum("store\_name");

// add\_aggregate\_column只负责复合运算

groupby\_param.add\_aggregate\_column("sales", "s", SUM, false);

groupby\_param.add\_aggregate\_column("counts", "c", SUM, false);

// add\_column只负责数学运算，其参数可以是复合运算的as\_name

groupby\_param.add\_column("s/c", "sale");

groupby\_param.add\_filter("c = 2");

ObScanParam scan\_param;

scan\_param.add\_column("store\_name");

scan\_param.add\_column("sale\_date", false);

scan\_param.add\_filter("sale\_date < '2000-01-03'");

scan\_param.set\_group\_by\_param(groupby\_param);

查询2000-01-03前售出了物品的商店及其每日平均每个物件的售价

select store\_name, sales/counts as sale

from store\_information where sale\_date < '2000-01-03';

+------------+----------+

| store\_name | sale |

+------------+----------+

| Los | 300.0000 |

| San | 125.0000 |

+------------+----------+

ObScanParam scan\_param;

scan\_param.add\_column("store\_name");

scan\_param.add\_column("sales", false);

scan\_param.add\_column("counts", false);

scan\_param.add\_column("sales/counts", "sale");

scan\_param.add\_filter("sale\_date < '2000-01-03'");

附件：创建表格SQL语句

DROP TABLE IF EXISTS `store\_information`;

CREATE TABLE `store\_information` (

`store\_name` varchar(30) NOT NULL,

`sales` int(11) NOT NULL,

`counts` int(11) NOT NULL,

`sale\_date` date DEFAULT NULL

) ENGINE=MyISAM DEFAULT CHARSET=utf8;

--

-- Dumping data for table `store\_information`

--

LOCK TABLES `store\_information` WRITE;

INSERT INTO `store\_information` VALUES

('Los',1500,1,'2000-01-03'),

('Los',300,1,'2000-01-01'),

('San',250,2,'2000-01-01'),

('Bos',700,3,'2000-01-05');

UNLOCK TABLES;

-EOF-