ORACLE 数据库

第 2 章

ORACLE 查询

传智播客.黑马程序员

一、单表查询

（一）简单条件查询

1.精确查询

需求：查询水表编号为 30408 的业主记录

查询语句：

```
select * from T_OWNERS where watermeter='30408'
```

查询结果：

	ID	NAME	ADDRESSID	HOUSENUMBER	WATERMETER	ADDDATE	OWNERTYPEID
1	3	马腾	1	1-3	30408	2015/9/11 星期五	1

2.模糊查询

需求：查询业主名称包含“刘”的业主记录

查询语句：

```
select * from t_owners where name like '%刘%'
```

查询结果：

	ID	NAME	ADDRESSID	HOUSENUMBER	WATERMETER	ADDDATE	OWNERTYPEID
1	5	刘华	2	2-5	30410	2013/9/11 星期三	1
2	6	刘东	2	2-2	30411	2014/9/11 星期四	1

3. and 运算符

需求：查询业主名称包含“刘”的并且门牌号包含 5 的业主记录

查询语句：

```
select * from t_owners where name like '%刘%' and housenumber
like '%5%'
```

查询结果：

	ID	NAME	ADDRESSID	HOUSENUMBER	WATERMETER	ADDDATE	OWNERTYPEID
1	5	刘华	2	2-5	30410	2013/9/11 星期三	1

4. or 运算符

需求：查询业主名称包含“刘”的或者门牌号包含 5 的业主记录

查询语句：

```
select * from t_owners
where name like '%刘%' or housenumber like '%5%'
```

查询结果：

	ID	NAME	ADDRESSID	HOUSENUMBER	WATERMETER	ADDDATE	OWNERTYPEID
1	5	刘华	2	2-5	30410	2013/9/11 星期三	1
2	6	刘东	2	2-2	30411	2014/9/11 星期四	1
3	7	周健	3	2-5	30433	2016/9/11 星期日	1

5. and 与 or 运算符混合使用

需求：查询业主名称包含“刘”的或者门牌号包含 5 的业主记录，并且地址编号为 3 的记录。

语句：

```
select * from t_owners where (name like '%刘%' or housenumber
like '%5%') and addressid=3
```

查询结果：

	ID	NAME	ADDRESSID	HOUSENUMBER	WATERMETER	ADDDATE	OWNERTYPEID
1	7	周健	3	2-5	30433	2016/9/11 星期日	1

因为 and 的优先级比 or 大，所以我们需要用（ ）来改变优先级。

6. 范围查询

需求：查询台账记录中用水字数大于等于 10000，并且小于等于 20000 的记录

我们可以用>= 和<=来实现，语句

```
select * from T_ACCOUNT
where usenum>=10000 and usenum<=20000
```

我们也可以用 between .. and ..来实现

```
select * from T_ACCOUNT
where usenum between 10000  and  20000
```

7. 空值查询

需求：查询 T_PRICETABLE 表中 MAXNUM 为空的记录

语句：

```
select * from T_PRICETABLE t where maxnum is null
```

查询结果：

	ID	PRICE	OWNERTYPEID	MINNUM	MAXNUM
▶ 1	3	4.45	1	10	
2	6	5.87	2	10	
3	9	6.36	3	10	

需求：查询 T_PRICETABLE 表中 MAXNUM 不为空的记录

语句：

```
select * from T_PRICETABLE t where maxnum  is not null
```

查询结果：

	ID	PRICE	OWNERTYPEID	MINNUM	MAXNUM
▶ 1	1	2.45	1	0	5
2	2	3.45	1	5	10
3	4	3.87	2	0	5
4	5	4.87	2	5	10
5	7	4.36	3	0	5
6	8	5.36	3	5	10

（二）去掉重复记录

需求：查询业主表中的地址 ID,不重复显示

语句：

```
select distinct addressid from T_OWNERS
```

（三）排序查询

1.升序排序

需求：对 T_ACCOUNT 表按使用量进行升序排序

语句：

```
select * from T_ACCOUNT order by usenum
```

查询结果：

	ID	OWNERUUID	OWNERTYPE	AREAID	YEAR	MONTH	NUM0	NUM1	USENUM
1	25	100	1	3	2012	12	95076	99324	0
2	8	1	1	1	2012	08	88331	89123	792
3	21	2	1	3	2012	09	89765	90567	802
4	9	1	1	1	2012	09	89123	90122	999
5	11	1	1	1	2012	11	93911	95012	1101
6	23	2	1	3	2012	11	93932	95076	1144
7	6	1	1	1	2012	06	79031	80201	1170
8	18	2	1	3	2012	06	79076	80287	1211
9	20	2	1	3	2012	08	88432	89765	1333

2.降序排序

需求：对 T_ACCOUNT 表按使用量进行降序排序

语句：

```
select * from T_ACCOUNT order by usenum desc
```

查询结果：

	ID	OWNERUUID	OWNERTYPE	AREAID	YEAR	MONTH	NUM0	NUM1	USENUM
1	13	2	1	3	2012	01	30334	50433	20099
2	1	1	1	1	2012	01	30203	50123	13000
3	15	2	1	3	2012	03	60765	74155	13000
4	3	1	1	1	2012	03	60303	74111	13000
5	14	2	1	3	2012	02	50433	60765	10332
6	2	1	1	1	2012	02	50123	60303	10180
7	19	2	1	3	2012	07	80287	88432	8145
8	7	1	1	1	2012	07	80201	88331	8130
9	24	2	1	3	2012	12	95076	99324	4248
10	12	1	1	1	2012	12	95012	99081	4069

（四）基于伪列的查询

在 Oracle 的表的使用过程中，实际表中还有一些附加的列，称为伪列。伪列就像表中的列一样，但是在表中并不存储。伪列只能查询，不能进行增删改操作。

接下来学习两个伪列：ROWID 和 ROWNUM。

1 ROWID

表中的每一行在数据文件中都有一个物理地址，ROWID 伪列返回的就是该行的物理地址。使用 ROWID 可以快速的定位表中的某一行。ROWID 值可以唯一的标识表中的一行。由于 ROWID 返回的是该行的物理地址，因此使用 ROWID 可以显示行是如何存储的。

查询语句：

```
select rowID,t.* from T_AREA t
```

查询结果如下：

	ROWID	ID	NAME
1	AAAM1uAAGAAAAD8AAA	1	海淀
2	AAAM1uAAGAAAAD8AAB	2	昌平
3	AAAM1uAAGAAAAD8AAC	3	西城
4	AAAM1uAAGAAAAD8AAD	4	东城
5	AAAM1uAAGAAAAD8AAE	5	朝阳
6	AAAM1uAAGAAAAD8AAF	6	玄武

我们可以通过指定 ROWID 来查询记录

```
select rowID,t.*
from T_AREA t
where ROWID='AAAM1uAAGAAAAD8AAC';
```

查询结果如下：

	ROWID	ID	NAME
1	AAAM1uAAGAAAAD8AAC	3	西城

2 ROWNUM

在查询的结果集中，ROWNUM 为结果集中每一行标识一个行号，第一行返回 1，第二行返回 2，以此类推。通过 ROWNUM 伪列可以限制查询结果集中返回的行

数。

查询语句：

```
select rownum,t.* from T_OWNERTYPE t
```

查询结果如下：

	ROWNUM	ID	NAME
1	1	1	居民
2	2	2	行政事业单位
3	3	3	商业

我们的分页查询需要用到此伪列，在本章第四小节详细讲解。

（五）聚合统计

ORACLE 的聚合统计是通过分组函数来实现的，与 MYSQL 一致。

1. 聚合函数

（1）求和 sum

需求：统计 2012 年所有用户的用水量总和

```
select sum(usenum) from t_account where year='2012'
```

查询结果如下：

	SUM(USENUM)
1	137868

（2）求平均 avg

需求：统计 2012 年所有用水量（字数）的平均值

```
select avg(usenum) from T_ACCOUNT where year='2012'
```

查询结果如下：

	AVG(USENUM)
1	5744.5

（3）求最大值 max

需求：统计 2012 年最高用水量（字数）

```
select max(usenum) from T_ACCOUNT where year='2012'
```

查询结果如下：

	MAX(USENUM)
1	20099

（4）求最小值 min

需求：统计 2012 年最低用水量（字数）

```
select min(usenum) from T_ACCOUNT where year='2012'
```

查询结果如下：

	MIN(USENUM)
1	792

（5）统计记录个数 count

需求：统计业主类型 ID 为 1 的业主数量

```
select count(*) from T_OWNERS t where ownertypeid=1
```

查询结果如下：

	COUNT(*)
1	8

2. 分组聚合 Group by

需求：按区域分组统计水费合计数

语句：

```
select areaid,sum(money) from t_account group by areaid
```

查询结果：

	AREAID	SUM(MONEY)
1	1	168751.1
2	3	169025.5

3. 分组后条件查询 having

需求：查询水费合计大于 16900 的区域及水费合计

语句：

```
select areaid,sum(money)  from t_account group by areaid
having sum(money)>169000
```

查询结果：

	AREAID	SUM(MONEY)
1	3	169025.5

二、连接查询

（一）多表内连接查询

（1）需求：查询显示业主编号，业主名称，业主类型名称，如下图：

	业主编号	业主名称	业主类型
1	1	范冰	居民
2	2	王强	居民
3	3	马腾	居民
4	4	林小玲	居民
5	5	刘华	居民
6	6	刘东	居民
7	7	周健	居民
8	8	张哲	居民
9	9	昌平区中西医结合医院	行政事业单位
10	10	美廉美超市	商业

查询语句：

```
select o.id 业主编号,o.name 业主名称,ot.name 业主类型
from T_OWNERS o,T_OWNERTYPE ot
where o.ownertypeid=ot.id
```

（2）需求：查询显示业主编号，业主名称、地址和业主类型，如下图

	业主编号	业主名称	地址	业主类型
1	3	马腾	明兴花园	居民
2	2	王强	明兴花园	居民
3	1	范冰	明兴花园	居民
4	6	刘东	鑫源秋墅	居民
5	5	刘华	鑫源秋墅	居民
6	4	林小玲	鑫源秋墅	居民
7	7	周健	华龙苑南里小区	居民
8	8	张哲	河畔花园	居民
9	10	美廉美超市	霍营	商业
10	9	昌平区中西医结合医院	霍营	行政事业单位

分析：此查询需要三表关联查询。分别是业主表，业主分类表和地址表

语句：

```
select o.id 业主编号,o.name 业主名称,ad.name 地址,
ot.name 业主类型
from T_OWNERS o,T_OWNERTYPE ot,T_ADDRESS ad
where o.ownertypeid=ot.id and o.addressid=ad.id
```

（3）需求：查询显示业主编号、业主名称、地址、所属区域、业主分类，如下图：

	业主编号	业主名称	区域	地址	业主类型
1	8	张哲	昌平	河畔花园	居民
2	7	周健	昌平	华龙苑南里小区	居民
3	6	刘东	海淀	鑫源秋墅	居民
4	5	刘华	海淀	鑫源秋墅	居民
5	4	林小玲	海淀	鑫源秋墅	居民
6	3	马腾	海淀	明兴花园	居民
7	2	王强	海淀	明兴花园	居民
8	1	范冰	海淀	明兴花园	居民
9	9	昌平区中西医结合医院	昌平	霍营	行政事业单位
10	10	美廉美超市	昌平	霍营	商业

分析：这里需要四个表关联查询，比上边多了一个区域表（T_AREA）

查询语句：

```
select o.id 业主编号,o.name 业主名称,ar.name 区域,  ad.name 地址, ot.name 业主类型
```

```
from T_OWNERS o ,T_OWNERTYPE ot,T_ADDRESS ad,T_AREA  ar
where   o.ownertypeid=ot.id   and   o.addressid=ad.id   and
ad.areaid=ar.id
```

（4）需求：查询显示业主编号、业主名称、地址、所属区域、收费员、业主分类，如下图：

	业主编号	业主名称	地址	所属区域	收费员	业主类型
▶ 1	6	刘东	鑫源秋墅	海淀	马小云	居民
2	5	刘华	鑫源秋墅	海淀	马小云	居民
3	4	林小玲	鑫源秋墅	海淀	马小云	居民
4	3	马腾	明兴花园	海淀	马小云	居民
5	2	王强	明兴花园	海淀	马小云	居民
6	1	范冰	明兴花园	海淀	马小云	居民

分析：此查询比上边又多了一个表 T_OPERATOR

语句：

```
select ow.id 业主编号,ow.name 业主名称,ad.name 地址,
ar.name 所属区域,op.name 收费员, ot.name 业主类型
from T_OWNERS ow,T_OWNERTYPE ot,T_ADDRESS ad ,
T_AREA ar,T_OPERATOR op
where ow.ownertypeid=ot.id and ow.addressid=ad.id
and ad.areaid=ar.id and ad.operatorid=op.id
```

（二）左外连接查询

需求：查询业主的账务记录，显示业主编号、名称、年、月、金额。如果此业主没有账务记录也要列出姓名。

	ID	NAME	YEAR	MONTH	MONEY
17	2	王强	2012	05	4843.65
18	2	王强	2012	06	2966.95
19	2	王强	2012	07	19955.25
20	2	王强	2012	08	3265.85
21	2	王强	2012	09	1964.90
22	2	王强	2012	10	8244.25
23	2	王强	2012	11	2802.80
24	2	王强	2012	12	10407.60
25	6	刘东			
26	5	刘华			
27	4	林小玲			
28	3	马腾			

分析：我们要查询这个结果，需要用到 T_OWNERS（业主表），T_ACCOUNT（台账表） 按照查询结果，业主表为左表、账务表为右表。

按照 SQL1999 标准的语法，查询语句如下：

```
SELECT ow.id,ow.name,ac.year ,ac.month,ac.money
FROM T_OWNERS ow left join T_ACCOUNT ac
on ow.id=ac.owneruuid
```

按照 ORACLE 提供的语法，就很简单了：

```
SELECT ow.id,ow.name,ac.year ,ac.month,ac.money FROM
T_OWNERS ow,T_ACCOUNT ac
WHERE ow.id=ac.owneruuid(+)
```

如果是左外连接，就在右表所在的条件一端填上（+）

（三）右外连接查询

需求：查询业主的账务记录，显示业主编号、名称、年、月、金额。如果账务记录没有对应的业主信息，也要列出记录。如下图：

20	2	王强	2012	05	4843.65
21	2	王强	2012	04	7212.80
22	2	王强	2012	03	32805.50
23	2	王强	2012	02	25313.40
24	2	王强	2012	01	49242.55
25			2012	12	44.51

SQL1999 标准的语句

```
select ow.id,ow.name,ac.year,ac.month,ac.money from
T_OWNERS ow right join  T_ACCOUNT ac
on ow.id=ac.owneruuid
```

ORACLE 的语法

```
select ow.id,ow.name,ac.year,ac.month,ac.money from
T_OWNERS ow , T_ACCOUNT ac
where ow.id(+) =ac.owneruuid
```

三、子查询

（一）where 子句中的子查询

1. 单行子查询

- 只返回一条记录
- 单行操作符

操作符	含义
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to

需求：查询 2012 年 1 月用水量大于平均值的台账记录

语句：

```
select * from T_ACCOUNT
where year='2012' and month='01' and  usenum>
( select avg(usenum) from T_ACCOUNT where year='2012' and
```

```
month='01' )
```

查询结果：

	ID	OWNERUUID	OWNERTYPE	AREAID	YEAR	MONTH	NUM0	NUM1	USENUM	METERUSER
1	13	2	1	3	2012	01	30334	50433	20099	1

平均值为：

	AVG(USENUM)
1	20009.5

2. 多行子查询

- 返回了多条记录
- 多行操作符

操作符	含义
IN	等于列表中的任何一个
ANY	和子查询返回的任意一个值比较
ALL	和子查询返回的所有值比较

in 运算符

（1）需求：查询地址编号为 1 、3、4 的业主记录

分析：如果我们用 or 运算符编写，SQL 非常繁琐，所以我们用 in 来进行查询

语句如下：

```
select * from T_OWNERS
where addressid in ( 1,3,4 )
```

查询结果如下：

	ID	NAME	ADDRESSID	HOUSENUMBER	WATERMETER	ADDDATE	OWNERTYPEID
1	1	范冰	1	1-1	30406	2015/9/10 星期四	1
2	2	王强	1	1-2	30407	2015/9/11 星期五	1
3	3	马腾	1	1-3	30408	2015/9/11 星期五	1
4	7	周健	3	2-6	30411	2016/9/11 星期日	1
5	8	张哲	4	2-2	30411	2016/9/11 星期日	1

（2）需求：查询地址含有“花园”的业主的信息

语句：

```
select * from T_OWNERS
where addressid in
( select id from t_address where name like '%花园%' )
```

查询结果：

	ID	NAME	ADDRESSID	HOUSENUMBER	WATERMETER	ADDDATE	OWNERTYPEID
1	1	范冰	1	1-1	30406	2015/9/10 星期四	1
2	2	王强	1	1-2	30407	2015/9/11 星期五	1
3	3	马腾	1	1-3	30408	2015/9/11 星期五	1
4	8	张哲	4	2-2	30411	2016/9/11 星期日	1

（3）需求：查询地址不含有“花园”的业主的信息

语句：

```
select * from T_OWNERS
where addressid not in
( select id from t_address where name like '%花园%' )
```

查询结果：

	ID	NAME	ADDRESSID	HOUSENUMBER	WATERMETER	ADDDATE	OWNERTYPEID
1	4	林小玲	2	2-4	30409	2015/6/15 星期一	1
2	5	刘华	2	2-5	30410	2013/9/11 星期三	1
3	6	刘东	2	2-2	30411	2014/9/11 星期四	1
4	7	周健	3	2-5	30433	2016/9/11 星期日	1
5	9	昌平区中西医结合医院	5	2-2	30422	2016/10/11 星期二	2
6	10	美廉美超市	5	4-2	30423	2016/10/12 星期三	3

（二）from 子句中的子查询

from 子句的子查询为多行子查询

需求：查询显示业主编号，业主名称，业主类型名称，条件为业主类型为”居民”，使用子查询实现。

语句：

```
select * from
```

```
(select o.id 业主编号,o.name 业主名称,ot.name 业主类型
from T_OWNERS o,T_OWNERTYPE ot
where o.ownertypeid=ot.id)
where 业主类型='居民'
```

查询结果如下：

	业主编号	业主名称	业主类型
1	1	范冰	居民
2	2	王强	居民
3	3	马腾	居民
4	4	林小玲	居民
5	5	刘华	居民
6	6	刘东	居民
7	7	周健	居民
8	8	张哲	居民

（三）select 子句中的子查询

select 子句的子查询必须为单行子查询

（1）需求：列出业主信息，包括 ID，名称，所属地址。

语句：

```
select id,name,
(select name from t_address where id=addressid) addressname
from t_owners
```

查询结果如下：

	ID	NAME	ADDRESSNAME
1	1	范冰	明兴花园
2	2	王强	明兴花园
3	3	马腾	明兴花园
4	4	林小玲	鑫源秋墅
5	5	刘华	鑫源秋墅
6	6	刘东	鑫源秋墅
7	7	周健	华龙苑南里小区
8	8	张哲	河畔花园
9	9	昌平区中西医结合医院	霍营
10	10	美廉美超市	霍营

（2）需求：列出业主信息，包括 ID，名称，所属地址，所属区域。

语句:

```
select id,name,
( select name from t_address where id=addressid  )
addressname,
( select  (select name from t_area where id=areaid ) from
t_address where id=addressid )
adrename
 from t_owners;
```

查询结果如下：

	ID	NAME	ADDRESSNAME	AREANAME
▶ 1	1	范冰	明兴花园	海淀
2	2	王强	明兴花园	海淀
3	3	马腾	明兴花园	海淀
4	4	林小玲	鑫源秋墅	海淀
5	5	刘华	鑫源秋墅	海淀
6	6	刘东	鑫源秋墅	海淀
7	7	周健	华龙苑南里小区	昌平
8	8	张哲	河畔花园	昌平
9	9	昌平区中西医结合医院	霍营	昌平
10	10	美廉美超市	霍营	昌平

四、分页查询

（一）简单分页

需求：分页查询台账表 T_ACCOUNT，每页 10 条记录

分析：我们在 ORACLE 进行分页查询，需要用到伪列 ROWNUM 和嵌套查询

我们首先显示前 10 条记录，语句如下：

```
select rownum,t.* from T_ACCOUNT t where rownum<=10
```

显示结果如下：

	ROWNUM	ID	OWNERUUID	OWNERTYPE	AREAID	YEAR	MONTH	NUM0	NUM1	USENUM	METERUSER
1	1	1	1	1	1	2012	01	30203	50123	19920	1
2	2	2	1	1	1	2012	02	50123	60303	10180	1
3	3	3	1	1	1	2012	03	60303	74111	13808	1
4	4	4	1	1	1	2012	04	74111	77012	2901	1
5	5	5	1	1	1	2012	05	77012	79031	2019	1
6	6	6	1	1	1	2012	06	79031	80201	1170	1
7	7	7	1	1	1	2012	07	80201	88331	8130	1
8	8	8	1	1	1	2012	08	88331	89123	792	1
9	9	9	1	1	1	2012	09	89123	90122	999	1
10	10	10	1	1	1	2012	10	90122	93911	3789	1

那么我们显示第 11 条到第 20 条的记录呢？编写语句：

```
select rownum,t.* from T_ACCOUNT t
where rownum>10 and rownum<=20
```

查询结果：

	ROWNUM	ID	OWNERUUID	OWNERTYPE	AREAID	YEAR	MONTH	NUM0	NUM1	USENUM

嗯？怎么没有结果？

这是因为 rownum 是在查询语句扫描每条记录时产生的，所以不能使用“大于”符号，只能使用“小于”或“小于等于” ，只用“等于”也不行。

那怎么办呢？我们可以使用子查询来实现

```
select * from
(select rownum r,t.* from T_ACCOUNT t where rownum<=20)
where r>10
```

查询结果如下：

	R	ID	OWNERUUID	OWNERTYPE	AREAID	YEAR	MONTH	NUM0	NUM1	USENUM
1	11	11	1	1	1	2012	11	93911	95012	1101
2	12	12	1	1	1	2012	12	95012	99081	4069
3	13	13	2	1	3	2012	01	30334	50433	20099
4	14	14	2	1	3	2012	02	50433	60765	10332
5	15	15	2	1	3	2012	03	60765	74155	13390
6	16	16	2	1	3	2012	04	74155	77099	2944
7	17	17	2	1	3	2012	05	77099	79076	1977
8	18	18	2	1	3	2012	06	79076	80287	1211
9	19	19	2	1	3	2012	07	80287	88432	8145
10	20	20	2	1	3	2012	08	88432	89765	1333

（二）基于排序的分页

需求：分页查询台账表 T_ACCOUNT，每页 10 条记录，按使用字数降序排序。

我们查询第 2 页数据，如果基于上边的语句添加排序，语句如下：

```
select * from
(select rownum r,t.* from T_ACCOUNT t where rownum<=20 order
by usenum desc)
where r>10
```

查询结果如下：

	R	ID	OWNERUUID	OWNERTYPE	AREAID	YEAR	MONTH	NUM0	NUM1	USENUM	METERUSER
1	13	13	2	1	3	2012	01	30334	50433	20099	1
2	15	15	2	1	3	2012	03	60765	74155	13390	1
3	14	14	2	1	3	2012	02	50433	60765	10332	1
4	19	19	2	1	3	2012	07	80287	88432	8145	1
5	12	12	1	1	1	2012	12	95012	99081	4069	1
6	16	16	2	1	3	2012	04	74155	77099	2944	1
7	17	17	2	1	3	2012	05	77099	79076	1977	1
8	20	20	2	1	3	2012	08	88432	89765	1333	1
9	18	18	2	1	3	2012	06	79076	80287	1211	1
10	11	11	1	1	1	2012	11	93911	95012	1101	1

经过验证，我们看到第 2 页的结果应该是下列记录

11	16	16	2	1	3	2012	04	74155	77099	2944	1
12	4	4	1	1	1	2012	04	74111	77012	2901	1
13	5	5	1	1	1	2012	05	77012	79031	2019	1
14	17	17	2	1	3	2012	05	77099	79076	1977	1
15	20	20	2	1	3	2012	08	88432	89765	1333	1
16	18	18	2	1	3	2012	06	79076	80287	1211	1
17	6	6	1	1	1	2012	06	79031	80201	1170	1
18	11	11	1	1	1	2012	11	93911	95012	1101	1
19	9	9	1	1	1	2012	09	89123	90122	999	1
20	8	8	1	1	1	2012	08	88331	89123	792	1

所以推断刚才的语句是错误的！那为什么是错误的呢？

我们可以先单独执行嵌套查询里面的那句话

```
select rownum r,t.* from T_ACCOUNT t
where rownum<=20 order by usenum desc
```

你会看到查询结果如下：

	R	ID	OWNERUUID	OWNERTYPE	AREAID	YEAR	MONTH	NUM0	NUM1	USENUM	METERUSER
1	13	13	2	1	3	2012	01	30334	50433	20099	1
2	1	1	1	1	1	2012	01	30203	50123	19920	1
3	3	3	1	1	1	2012	03	60303	74111	13808	1
4	15	15	2	1	3	2012	03	60765	74155	13390	1
5	14	14	2	1	3	2012	02	50433	60765	10332	1
6	2	2	1	1	1	2012	02	50123	60303	10180	1
7	19	19	2	1	3	2012	07	80287	88432	8145	1
8	7	7	1	1	1	2012	07	80201	88331	8130	1
9	12	12	1	1	1	2012	12	95012	99081	4069	1
10	10	10	1	1	1	2012	10	90122	93911	3789	1
11	16	16	2	1	3	2012	04	74155	77099	2944	1
12	4	4	1	1	1	2012	04	74111	77012	2901	1
13	5	5	1	1	1	2012	05	77012	79031	2019	1
14	17	17	2	1	3	2012	05	77099	79076	1977	1
15	20	20	2	1	3	2012	08	88432	89765	1333	1
16	18	18	2	1	3	2012	06	79076	80287	1211	1
17	6	6	1	1	1	2012	06	79031	80201	1170	1
18	11	11	1	1	1	2012	11	93911	95012	1101	1

你会发现排序后的 R 是乱的。这是因为 ROWNUM 伪列的产生是在表记录扫描是产生的，而排序是后进行的，排序时 R 已经产生了，所以排序后 R 是乱的。

那该如何写呢？

很简单，我们只要再嵌套一层循环（一共三层），让结果先排序，然后对排序后的结果再产生 R，这样就不会乱了。

语句如下：

```
select * from
(select rownum r,t.* from
   (select * from T_ACCOUNT order by usenum desc) t
      where rownum<=20 )
where r>10
```

结果如下：

	R	ID	OWNERUUID	OWNERTYPE	AREAID	YEAR	MONTH	NUM0	NUM1	USENUM	METERUSER
1	11	10	1	1	1	2012	10	90122	93911	3789	1
2	12	22	2	1	3	2012	10	90567	93932	3365	1
3	13	16	2	1	3	2012	04	74155	77099	2944	1
4	14	4	1	1	1	2012	04	74111	77012	2901	1
5	15	5	1	1	1	2012	05	77012	79031	2019	1
6	16	17	2	1	3	2012	05	77099	79076	1977	1
7	17	20	2	1	3	2012	08	88432	89765	1333	1
8	18	18	2	1	3	2012	06	79076	80287	1211	1
9	19	6	1	1	1	2012	06	79031	80201	1170	1
10	20	23	2	1	3	2012	11	93932	95076	1144	1

五、单行函数

（一）字符函数

函　数	说　　明
ASCII	返回对应字符的十进制值
CHR	给出十进制返回字符
CONCAT	拼接两个字符串，与 || 相同
INITCAT	将字符串的第一个字母变为大写
INSTR	找出某个字符串的位置
INSTRB	找出某个字符串的位置和字节数
LENGTH	以字符给出字符串的长度
LENGTHB	以字节给出字符串的长度
LOWER	将字符串转换成小写
LPAD	使用指定的字符在字符的左边填充
LTRIM	在左边裁剪掉指定的字符
RPAD	使用指定的字符在字符的右边填充
RTRIM	在右边裁剪掉指定的字符
REPLACE	执行字符串搜索和替换
SUBSTR	取字符串的子串
SUBSTRB	取字符串的子串（以字节）
SOUNDEX	返回一个同音字符串

TRANSLATE	执行字符串搜索和替换
TRIM	裁剪掉前面或后面的字符串
UPPER	将字符串变为大写

常用字符函数讲解：

（1）求字符串长度 LENGTH

语句：

```
select length('ABCD') from dual;
```

显示结果为：

	LENGTH('ABCD')
1	4

（2）求字符串的子串 SUBSTR

语句：

```
select substr('ABCD',2,2) from dual;
```

显示结果为：

	SUBSTR('ABCD',2,2)
1	BC

（3）字符串拼接 CONCAT

语句：

```
select concat('ABC','D') from dual;
```

查询结果如下：

	CONCAT('ABC','D')
1	ABCD

我们也可以用|| 对字符串进行拼接

```
select 'ABC'||'D' from dual;
```

查询结果同上。

（二）数值函数

函数	说明
ABS(value)	绝对值
CEIL(value)	大于或等于 value 的最小整数
COS(value)	余弦
COSH(value)	反余弦
EXP(value)	e 的 value 次幂
FLOOR(value)	小于或等于 value 的最大整数
LN(value)	value 的自然对数
LOG(value)	value 的以 10 为底的对数
MOD(value,divisor)	求模
POWER(value,exponent)	value 的 exponent 次幂
ROUND(value,precision)	按 precision 精度 4 舍 5 入
SIGN(value)	value 为正返回 1;为负返回-1;为 0 返回 0.
SIN(value)	余弦
SINH(value)	反余弦

SQRT(value)	value 的平方根
TAN(value)	正切
TANH(value)	反正切
TRUNC(value,按 precision)	按照 precision 截取 value
VSIZE(value)	返回 value 在 ORACLE 的存储空间大小

常用数值函数讲解：

（1）四舍五入函数 ROUND

语句：

```
select round(100.567) from dual
```

查询结果如下：

	ROUND(100.567)
1	101

语句：

```
select round(100.567,2) from dual
```

查询结果如下：

	ROUND(100.567,2)
1	100.57

（2）截取函数 TRUNC

语句：

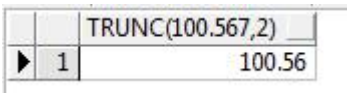
```
select trunc(100.567) from dual
```

查询结果：

	TRUNC(100.567)
1	100

语句：

```
select trunc(100.567,2) from dual
```



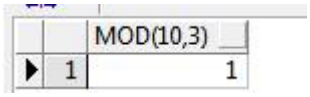
	TRUNC(100.567,2)
1	100.56

（3）取模　MOD

语句：

```
select mod(10,3) from dual
```

结果：



	MOD(10,3)
1	1

（三）日期函数

函　数	描　　述
ADD_MONTHS	在日期 date 上增加 count 个月
GREATEST(date1,date2,. . .)	从日期列表中选出最晚的日期
LAST_DAY(date)	返回日期 date 所在月的最后一天
LEAST(date1, date2, . . .)	从日期列表中选出最早的日期
MONTHS_BETWEEN(date2, date1)	给出 Date2 - date1 的月数(可以是小数)
NEXT_DAY(date,’ day’)	给出日期 date 之后下一天的日期 ,这里的 day 为星期，如： MONDAY,Tuesday 等。
NEW_TIME(date,’ this’ ,’	给出在 this 时区=Other 时区的日期和时间

other’）	
ROUND(date,’ format’）	未指定 format 时，如果日期中的时间在中午之前，则将日期中的时间截断为 12 A.M.(午夜，一天的开始),否则进到第二天。时间截断为 12 A.M.(午夜，一天的开始),否则进到第二天。
TRUNC(date,’ format’）	未指定 format 时，将日期截为 12 A.M.(午夜，一天的开始).

我们用 sysdate 这个系统变量来获取当前日期和时间

语句如下：

```
select sysdate from dual
```

查询结果如下：

	SYSDATE
1	2016/10/11 星期二 下午 10:43:46

常用日期函数讲解：

（1）加月函数 ADD_MONTHS ：在当前日期基础上加指定的月

语句：

```
select add_months(sysdate,2) from dual
```

查询结果如下：

	ADD_MONTHS(SYSDATE,2)
1	2016/12/11 星期日 下午 10:46:49

（2）求所在月最后一天 LAST_DAY

语句：

```
select last_day(sysdate) from dual
```

查询结果如下：

	LAST_DAY(SYSDATE)
1	2016/10/31 星期一 下午 10:48:47

（3）日期截取 TRUNC

语句：

```
select TRUNC(sysdate) from dual
```

查询结果如下：

	TRUNC(SYSDATE)
1	2016/10/11 星期二

语句：

```
select TRUNC(sysdate,'yyyy') from dual
```

查询结果如下：

	TRUNC(SYSDATE,'YYYY')
1	2016/1/1 星期五

语句：

```
select TRUNC(sysdate,'mm') from dual
```

查询结果如下：

	TRUNC(SYSDATE,'MM')
1	2016/10/1 星期六

（四）转换函数

函 数	描 述
CHARTOROWID	将 字符转换到 rowid 类型
CONVERT	转换一个字符节到另外一个字符节
HEXTORAW	转换十六进制到 raw 类型

RAWTOHEX	转换 raw 到十六进制
ROWIDTOCHAR	转换 ROWID 到字符
TO_CHAR	转换日期格式到字符串
TO_DATE	按照指定的格式将字符串转换到日期型
TO_MULTIBYTE	把单字节字符转换到多字节
TO_NUMBER	将数字字串转换到数字
TO_SINGLE_BYTE	转换多字节到单字节

常用转换函数讲解：

（1）数字转字符串 TO_CHAR

语句：

```
select TO_CHAR(1024) from dual
```

查询结果：

	TO_CHAR(1024)
1	1024

（2）日期转字符串 TO_CHAR

语句：

```
select TO_CHAR(sysdate,'yyyy-mm-dd') from dual
```

查询结果：

	TO_CHAR(SYSDATE,'YYYY-MM-DD')
1	2016-10-11

语句：

```
select TO_CHAR(sysdate,'yyyy-mm-dd hh:mi:ss') from dual
```

查询结果：

	TO_CHAR(SYSDATE,'YYYY-MM-DDHH:
1	2016-10-11 11:00:59

（3）字符串转日期 TO_DATE

语句：

```
select TO_DATE('2017-01-01','yyyy-mm-dd') from dual
```

查询结果如下：

	TO_DATE('2017-01-01','YYYY-MM-
1	2017/1/1 星期日

（4）字符串转数字 TO_NUMBER

语句：

```
select to_number('100') from dual
```

（五）其它函数

（1）空值处理函数 NVL

用法：

NVL（检测的值，如果为 null 的值）；

语句：

```
select NVL(NULL,0) from dual
```

查询结果如下：

	NVL(NULL,0)
1	0

需求：

显示价格表中业主类型 ID 为 1 的价格记录 如果上限值为 NULL,则显示 9999999

语句：

```
select PRICE,MINNUM,NVL(MAXNUM,9999999)
from T_PRICETABLE where OWNERTYPEID=1
```

查询结果：

	PRICE	MINNUM	NVL(MAXNUM,9999999)
1	2.45	0	5
2	3.45	5	10
3	4.45	10	9999999

（2）空值处理函数 NVL2

用法：

NVL2（检测的值，如果不为 null 的值，如果为 null 的值）；

需求 :显示价格表中业主类型 ID 为 1 的价格记录 ,如果上限值为 NULL,显示“不限”.

	PRICE	MINNUM	NVL2(MAXNUM,TO_CHAR(MAXNUM),'?
1	2.45	0	5
2	3.45	5	10
3	4.45	10	不限

语句：

```
select PRICE,MINNUM,NVL2(MAXNUM,to_char(MAXNUM) , '不限')
from T_PRICETABLE where OWNERTYPEID=1
```

（3）条件取值 decode

语法：

```
decode(条件,值 1,翻译值 1,值 2,翻译值 2,...值 n,翻译值 n,缺省值)
【功能】根据条件返回相应值
```

需求：显示下列信息（不要关联查询业主类型表，直接判断 1 2 3 的值）

	NAME	类型
1	张哲	居民
2	昌平区中西医结合医院	行政事业单位
3	范冰	居民
4	王强	居民
5	马腾	居民
6	林小玲	居民
7	刘华	居民
8	周健	居民
9	刘东	居民
10	美廉美超市	商业

语句：

```
select name,decode( ownertypeid,1,'居民',2,'行政事业单位',3,'商业') as 类型 from T_OWNERS
```

上边的语句也可以用 case when then 语句来实现

```
select name ,(case ownertypeid
                when 1 then '居民'
                when 2 then '行政事业单位'
                when 3 then '商业'
                else '其它'
            end
            ) from T_OWNERS
```

还有另外一种写法：

```
select name,(case
            when ownertypeid= 1 then '居民'
            when ownertypeid= 2 then '行政事业'
            when ownertypeid= 3 then '商业'
          end )
```

```
        from T_OWNERS
```

六、行列转换

需求：按月份统计 2012 年各个地区的水费，如下图

	地区	一月	二月	三月	四月	五月	六月	七月	八月	九月	十月	十一月	十二月
1	海淀	48804	24941	33829.6	7107.45	4946.55	2866.5	19918.5	1940.4	2447.55	9283.05	2697.45	9969.05
2	西城	49242.55	25313.4	32805.5	7212.8	4843.65	2966.95	19955.25	3265.85	1964.9	8244.25	2802.8	10407.6

```
select  (select name from T_AREA where id=  areaid ) 区域,
  sum( case when month='01' then money else 0 end) 一月,
  sum( case when month='02' then money else 0 end) 二月,
  sum( case when month='03' then money else 0 end) 三月,
  sum( case when month='04' then money else 0 end) 四月,
  sum( case when month='05' then money else 0 end) 五月,
  sum( case when month='06' then money else 0 end) 六月,
  sum( case when month='07' then money else 0 end) 七月,
  sum( case when month='08' then money else 0 end) 八月,
  sum( case when month='09' then money else 0 end) 九月,
  sum( case when month='10' then money else 0 end) 十月,
  sum( case when month='11' then money else 0 end) 十一月,
  sum( case when month='12' then money else 0 end) 十二月
  from T_ACCOUNT where year='2012' group by areaid
```

需求：按季度统计 2012 年各个地区的水费，如下图

	地区	第一季度	第二季度	第三季度	第四季度
1	海淀 …	107574.6	14920.5	24306.45	21949.55
2	西城 …	107361.45	15023.4	25186	21454.65

语句如下：

```
select  (select name from T_AREA where id=  areaid ) 区域,
   sum( case when month>='01' and month<='03' then money else
0 end) 第一季度,
   sum( case when month>='04' and month<='06' then money else
0 end) 第二季度,
   sum( case when month>='07' and month<='09' then money else
0 end) 第三季度,
   sum( case when month>='10' and month<='12' then money else
0 end) 第四季度
   from T_ACCOUNT where year='2012' group by areaid
```

七、分析函数

以下三个分析函数可以用于排名使用。

下图为三种排名方式的举例

分数	三种排名方式		
	相同的值排名相同，排名跳跃	相同的值排名相同，排名连续	排名连续，不论值是否相同
100	1	1	1
98	2	2	2
98	2	2	3
98	2	2	4
95	5	3	5
95	5	3	6
92	7	4	7
92	7	4	8
90	9	5	9
89	10	6	10
87	11	7	11
87	11	7	12

（1） RANK　　相同的值排名相同，排名跳跃

需求：对 T_ACCOUNT 表的 usenum 字段进行排序，相同的值排名相同，排名跳

跃

语句：

```
select rank() over(order by usenum desc ),usenum from T_ACCOUNT
```

结果：

	RANK()OVER(ORDERBYUSENUMDESC)	USENUM
1	1	20099
2	2	13000
3	2	13000
4	2	13000
5	5	10332
6	6	10180
7	7	8145
8	8	8130
9	9	4248
10	10	4069
11	11	3789
12	12	3365
13	13	2944
14	14	2901
15	15	2019

（2） DENSE_RANK 相同的值排名相同，排名连续

需求：对 T_ACCOUNT 表的 usenum 字段进行排序，相同的值排名相同，排名连续

语句：

```
select dense_rank() over(order by usenum desc ),usenum from T_ACCOUNT
```

结果：

	DENSE_RANK()OVER(ORDERBYUSENUM	USENUM
1	1	20099
2	2	13000
3	2	13000
4	2	13000
5	3	10332
6	4	10180
7	5	8145
8	6	8130
9	7	4248
10	8	4069
11	9	3789
12	10	3365
13	11	2944
14	12	2901
15	13	2019
16	14	1977

（3） ROW_NUMBER 返回连续的排名，无论值是否相等

需求：对 T_ACCOUNT 表的 usenum 字段进行排序，返回连续的排名，无论值是否相等

语句：

```
select row_number() over(order by usenum desc ),usenum
from T_ACCOUNT
```

	ROW_NUMBER()OVER(ORDERBYUSENUM	USENUM
1	1	20099
2	2	13000
3	3	13000
4	4	13000
5	5	10332
6	6	10180
7	7	8145
8	8	8130
9	9	4248
10	10	4069
11	11	3789
12	12	3365
13	13	2944
14	14	2901
15	15	2019
16	16	1977
17	17	1333
18	18	1211
19	19	1170
20	20	1144

用 row_number()分析函数实现的分页查询相对三层嵌套子查询要简单的多:

```
select * from
(select row_number() over(order by usenum desc )
rownumber,usenum from T_ACCOUNT)
where rownumber>10 and rownumber<=20
```

查询结果如下：

	ROWNUMBER	USENUM
1	11	3789
2	12	3365
3	13	2944
4	14	2901
5	15	2019
6	16	1977
7	17	1333
8	18	1211
9	19	1170
10	20	1144

八、集合运算

（一）什么是集合运算

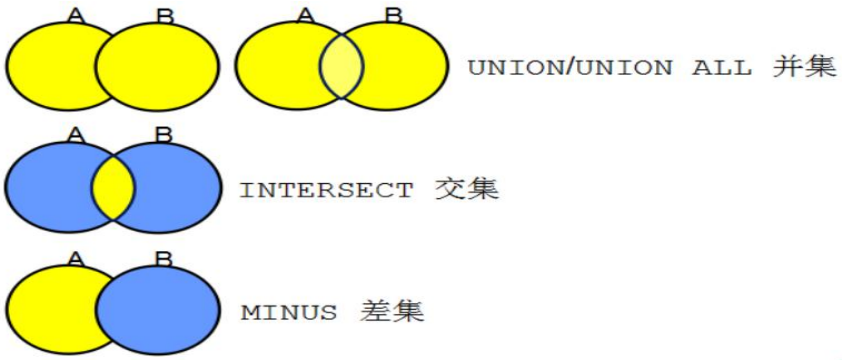
集合运算，集合运算就是将两个或者多个结果集组合成为一个结果集。集合运算包括：

··UNION ALL(并集)，返回各个查询的所有记录，包括重复记录。

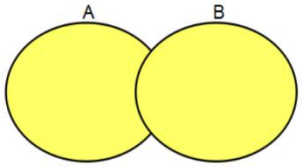
··UNION(并集)，返回各个查询的所有记录，不包括重复记录。

··INTERSECT(交集)，返回两个查询共有的记录。

··MINUS(差集)，返回第一个查询检索出的记录减去第二个查询检索出的记录之后剩余的记录。



（二）并集运算



UNION运算符返回两个集合去掉重复元素后的所有记录。

UNION ALL 不去掉重复记录

```
select * from t_owners where id<=7
union all
select * from t_owners where id>=5
```

结果如下：

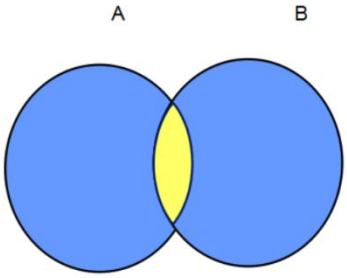
	ID	NAME	ADDRESSID	HOUSENUMBER	WATERMETER	ADDDATE	OWNERTYPEID
1	1	范冰	1	1-1	30406	2015/4/12 星期日	1
2	2	王强	1	1-2	30407	2015/2/14 星期六	1
3	3	马腾	1	1-3	30408	2015/3/18 星期三	1
4	4	林小玲	2	2-4	30409	2015/6/15 星期一	1
5	5	刘华	2	2-5	30410	2013/9/11 星期三	1
6	6	刘东	2	2-2	30411	2014/9/11 星期四	1
7	7	周健	3	2-5	30433	2016/9/11 星期日	1
8	5	刘华	2	2-5	30410	2013/9/11 星期三	1
9	6	刘东	2	2-2	30411	2014/9/11 星期四	1
10	7	周健	3	2-5	30433	2016/9/11 星期日	1
11	8	张哲	4	2-2	30455	2016/9/11 星期日	1
12	9	昌平区中西医结合医院	5	2-2	30422	2016/10/11 星期二	2
13	10	美廉美超市	5	4-2	30423	2016/10/12 星期三	3

UNION 去掉重复记录

```
select * from t_owners where id<=7
union
select * from t_owners where id>=5
```

	ID	NAME	ADDRESSID	HOUSENUMBER	WATERMETER	ADDDATE	OWNERTYPEID
1	1	范冰	1	1-1	30406	2015/4/12 星期日	1
2	2	王强	1	1-2	30407	2015/2/14 星期六	1
3	3	马腾	1	1-3	30408	2015/3/18 星期三	1
4	4	林小玲	2	2-4	30409	2015/6/15 星期一	1
5	5	刘华	2	2-5	30410	2013/9/11 星期三	1
6	6	刘东	2	2-2	30411	2014/9/11 星期四	1
7	7	周健	3	2-5	30433	2016/9/11 星期日	1
8	8	张哲	4	2-2	30455	2016/9/11 星期日	1
9	9	昌平区中西医结合医院	5	2-2	30422	2016/10/11 星期二	2
10	10	美廉美超市	5	4-2	30423	2016/10/12 星期三	3

（三）交集运算



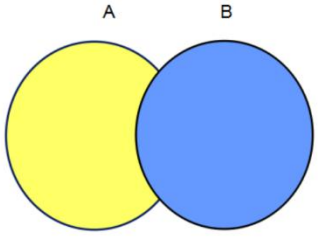
INTERSECT 运算符返回同时属于两个集合的记录

```
select * from t_owners where id<=7
intersect
select * from t_owners where id>=5
```

结果：

	ID	NAME	ADDRESSID	HOUSENUMBER	WATERMETER	ADDDATE	OWNERTYPEID
1	5	刘华	2	2-5	30410	2013/9/11 星期三	1
2	6	刘东	2	2-2	30411	2014/9/11 星期四	1
3	7	周健	3	2-5	30433	2016/9/11 星期日	1

（四）差集运算



MINUS返回属于第一个集合，但不属于第二个集合的记录。

```
select * from t_owners where id<=7
```

```
minus
select * from t_owners where id>=5
```

结果:

	ID	NAME	ADDRESSID	HOUSENUMBER	WATERMETER	ADDDATE	OWNERTYPEID
1	1	范冰	1	1-1	30406	2015/4/12 星期日	1
2	2	王强	1	1-2	30407	2015/2/14 星期六	1
3	3	马腾	1	1-3	30408	2015/3/18 星期三	1
4	4	林小玲	2	2-4	30409	2015/6/15 星期一	1

如果我们用 minus 运算符来实现分页，语句如下：

```
select rownum,t.* from T_ACCOUNT t where rownum<=20
minus
select rownum,t.* from T_ACCOUNT t where rownum<=10
```

	ROWNUM	ID	OWNERUUID	OWNERTYPE	AREAID	YEAR	MONTH	NUM0	NUM1	USENUM
1	11	11	1	1	1	2012	11	93911	95012	1101
2	12	12	1	1	1	2012	12	95012	99081	4069
3	13	13	2	1	3	2012	01	30334	50433	20099
4	14	14	2	1	3	2012	02	50433	60765	10332
5	15	15	2	1	3	2012	03	60765	74155	13000
6	16	16	2	1	3	2012	04	74155	77099	2944
7	17	17	2	1	3	2012	05	77099	79076	1977
8	18	18	2	1	3	2012	06	79076	80287	1211
9	19	19	2	1	3	2012	07	80287	88432	8145
10	20	20	2	1	3	2012	08	88432	89765	1333

九、总结

（一）知识点总结

（二）上机任务布置

为《自来水收费系统》开发统计模块相关的功能

1.收费日报单（总）

统计某日的收费，按区域分组汇总，效果如下：

	区域	用水量(吨)	金额
▶ 1	海淀 …	2.901	7107.45

2.收费日报单（收费员）

统计某收费员某日的收费，按区域分组汇总，效果如下：

	区域	用水量(吨)	金额
▶ 1	海淀 …	2.901	7107.45

3.收费月报表（总）

统计某年某月的收费记录，按区域分组汇总

	区域	用水量(吨)	金额
▶ 1	海淀 …	2.901	7107.45

4.收费月报表（收费员）

统计某收费员某年某月的收费记录，按区域分组汇总

5.收费年报表（分区域统计）

统计某年收费情况，按区域分组汇总，效果如下：

	区域	用水量(吨)	金额
▶ 1	海淀 …	64.809	158782.05

6.收费年报表（分月份统计）

统计某年收费情况，按月份分组汇总，效果如下

	月份	使用吨数	金额
▶ 1	01	4.069	9969.05
2	02	20.099	49242.55
3	03	10.332	25313.4
4	04	13.39	32805.5
5	05	2.944	7212.8
6	06	1.977	4843.65
7	07	1.211	2966.95
8	08	8.145	19955.25
9	09	1.333	3265.85
10	10	0.802	1964.9
11	11	3.365	8244.25
12	12	1.144	2802.8

7.收费年报表（分月份统计）

统计某年收费情况，按月份分组汇总，效果如下

	统计项	一月	二月	三月	四月	五月	六月	七月	八月	九月	十月	十一月	十二月
▶ 1	用水量(吨)	4.069	20.099	10.332	13.39	2.944	1.977	1.211	8.145	1.333	0.802	3.365	1.144
2	金额(元)	9969.05	49242.55	25313.4	32805.5	7212.8	4843.65	2966.95	19955.25	3265.85	1964.9	8244.25	2802.8

8.统计用水量，收费金额（分类型统计）

根据业主类型分别统计每种居民的用水量（整数，四舍五入）及收费金额 ，如果该类型在台账表中无数据也需要列出值为 0 的记录 ， 效果如下：

	NAME	用水量(吨)	金额
▶ 1	居民 ···	138	337821.11
2	商业 ···	0	0
3	行政事业单位 ···	0	0

分析：这里所用到的知识点包括左外连接、sum()、分组 group by 、round() 和 nvl()

9.统计每个区域的业主户数，并列出合计

	区域	业主户数
▶ 1	昌平 ···	4
2	海淀 ···	6
3	合计 ···	10

10.统计每个区域的业主户数，如果该区域没有业主户数也要列出 0

如图:

	区域	业主户数
▶ 1	昌平 ···	4
2	朝阳 ···	0
3	东城 ···	0
4	西城 ···	0
5	海淀 ···	6
6	玄武 ···	0