第四章　关系数据库标准语言SQL

重要考点提示

- SQL语句ALTER为数据库添加、删除或修改字段
- 进行数据的插入和更新操作
- SQL语句进行嵌套查询
- SQL语句进行超联接查询，主要是内部联接
- SQL语句的GROUP短语进行分组及计算查询以及HAVING子句的作用
- SQL语句建立视图，包括视图中字段名的重新定义

一、SQL概述

1、概念

SQL是结构化查询语言Structured Query Langvage的缩写，它包括数据查询、数据定义、数据操纵和数据控制4部分，VF在SQL方面支持数据定义、数据查询和数据操纵功能。另外由于VF自身在安全控件方面的缺陷，所以它没有提供数据控制功能。

2、SQL语言的特点

① 一种一体化的语言

② 一种高度非过程化的语言

③ 语言非常简洁

④ 可以直接以命令方式交互使用，也可以以程序方式使用

二、查询功能

SQL的核心是查询。基本形式由SELECT——FROM——WHERE查询块组成，多个查询块可嵌套执行。

VF的SQL的SELECT命令的语法格式如下：

```
SELECT  [ALL/DISTINCT][TOP  nExpr[PERCENT]]
[Alias.] select_Item[AS  Column_name][,Alias.]Select_Item[AS Column_name…]
FROM [FORCE] [DatabaseName!]Table [[AS] Local_Alias]
    [[INNER/LEFT[OUTER]/RIGHT[OUTER]/FULL[OUTER]JOIN
    DatabaseName!] Table [[AS] Local_Alias]
   [ON joinCondition…]
   [[INTO Destination]
   /[TO  [ADDITIVE]/ TO  PRINTER [PROMPT]/TO SCREEN]]
[PREFERENCE  preferencename]
[NOCONSOLE]
[PLAN]
[NOWAIT]
[WHERE  joincondition[AND Joincondition…]
    [AND/OR filtercondition[AND/OR  filtercondition…]
[GROUP BY GroupColumn[,GroupColumn…]]
     [HAVING  FilterCondition]
[UNION [ALL]  SELECTCommand]
[ORDER BY Order  Item[ASC/DESC][,Order  Item[ASC/DESC]…]]
```

其中主要短语的含义如下:
SELECT:说明要查询的数据.
FROM: 说明要查询的数据来自哪个或哪些表,可
以单个表或多个表进行查询.
WHERE:说明书查询条件即选择元组的条件.
GROUP BY:短语用于对查询结果进行分组,可利用它进行分组汇总.
HAVING:短语必须跟随GROUP BY使用.它用来限定分组必须满足的条件
ORDER BY 短语用于对查询结果进行排序

1、简单查询

这些简单的查询基于单个表,可有简单的查询条件,这样的查询由SELECT和FROM短语构成无条件查询或由SELECT FROM和WHERE短语构成条件查询

eg1 从职工关系中检索所有的工资值

select 工资 from 职工

若去掉重复需指使指定distinct 短语

select distinct 工资 from 职工

distinct短语的作用是去掉查询结课中的重复值

eg2 从职工关系中检索出所有的职工号及其工资值

SELECT 职工号 ,工资 FROM 职工

不同的字段名之间用,隔开

Eg3 检查仓库关系中的所有元组

select * from 仓库

*是通配符,表示所有的属性(字段)这里的命令等同于

select 仓库号,城市,面积 from 仓库

Eg4 检索工资多于1230元的职工号

select 职工号 from 职工 where 工资>1230

用where短语指定查询条件,查询条件可以是任意复杂的逻辑表达式

Eg5 检索哪些仓库有工资多于1210元的职工

select distinct 仓库号 from 职工 where 工资>1210

Eg6 给出在仓库”wh1” 或“wh2”工作,并且工资少于1250的职工号

select 职工号 from 职工;

where 工资<1250 AND (仓库号=“WH1” OR 仓库号=“WH2”)

分号表示续行符号

系统是怎样完成SQL命令的检索要求的呢?
如果有 where子句系统首先根据指定的条件依次检验关系中的每个元组.如果没有指定where 子句则直接选出满足条件的元组(相当于关系的选择操作)并显示select子句中指定属性的值(相当于关系的投影操作)

2、简单的联接查询

联接是关系的基本操作之一,联接查询是一种基于多个关系的查询

Eg7 找出工资多于1230元的职工和他们所在的城市

select 职工号,城市 from 职工, 仓库 where ;

工资>1230 AND 职工.仓库号=仓库.仓库号

如果在检索命令的from之后有两个关系,那么这两个关系之间肯定有一种联系,否则无法构成检索表达式,当from 之后的多个关系中含有相同的属性名时,必需用关系前缀直接指明属性所属关系, “.”前是关系名,后是属性名

Eg8 找出工作在面积大于400的仓库的职工号以及这些职工所在的城市

```
select   职工号,城市   from   职工,仓库;
where 面积>400 AND职工,仓库号=职工,仓库号
```

3.嵌套查询

这类基于多个关系的查询.查询结果出自一个关系,但相关的条件却涉及多个关系,之前的例子中where之后是一个相对独立的条件.这个条件或为真或为假.但是有时需要用另外的方式来表达检索要求.比如当检索关系X中的元组时.它的条件依赖于相关的关系Y中的元组属性值.这时使用SQL的嵌套查询功能非常方便.

其内层基本上也是一个select-from-where查询语句.这种简单的嵌套查询可使用谓词IN或NOT IN来判断在外层的查询条件中是否包含内层查询的结果.

Eg9 哪些城市至少有 一个仓库的职工工资为1250元

select 城市 from 仓库 where 仓库号 IN;

(select 仓库号from职工where工资=1250)

In相当于集合运算符∈

Eg10 查询所有职工的工资都多于1210元的仓库的信息.

这个检索也可描述为没有一个职工的工资少于1210的仓库的信息.

Select * from 仓库 where 仓库号 not in;

(select 仓库号 from 职工 where 工资<1210)

在仓库WH4中还没有职工

如果要排除那些没有职工的仓库,检索要求应描述为检索所有职工的工资都多于1210元的仓库信息并且该仓库至少有一名职工.

Select * from 仓库 where 仓库号 not in (select 仓库号 from 职工 where 工资<1210) AND 仓库号 in ;

(select 仓库号 from 职工)

内层是两个并列的查询,在结果中将不包含没有职工的仓库信息.

能不能写成

select * from 仓库 where 仓库号 in ;
(select 仓库号from 职工where 工资>1210)

此结果为只要仓库中有一个职工的工资大于1210元即被选择出来.

Eg11 .找出和职工E4挣同样工资的所有职工

select 职工号 from 职工 where 工资=;
(select 工资 from 职工 where 职工号="E4")

4.几个特殊的运算符

(1) between…and它表示的是查询的条件是在…和…之间.相当于用and连接的一个逻辑表达式

Eg12 检索出工资在1220元到1240元范围内的职工信息.

Select * from 职工 where 工资 between 1220 and 1240

等价于 select * from 职工 where 工资 >=1220 and ;

工资<=1240

(2) LIKE: 它是一个字符串匹配运算符, ”_”表示一个字符.

通配符”%”表示0个或多个字符

Eg13 从供应商关系中检索出全部公司的信息,不要工厂其他供应商的信息.

Select * from 供应商 where 供应商 LIKE“%公司”

Eg14 从供应商关系中检索出包含”华电子厂”共五个字符的供应商信息.

Select * from 供应商 where 供应商名 like”_华电子厂”

Eg15 找出不在北京的全部供应商的信息

select * from 供应商 where 地址!=“北京”

在SQL中不等于用”!=“表示,也可用NOT写出等价命令

select * from 供应商 where NOT (地址=北京)

NOT应用范围很广,比如有NOT IN，NOT BETWEEN

若提出与eg12相反的要求，找出工资不在1220到1240之间的全部职工的信息可用。

Select * from 职工 where 工资 not between 1220 and 1240 与之等价的还可写成

Select * from 职工 where 工资<1220 OR 工资>1240

5.排序

使用SQL SELECT可将查询结果进行排序,排序的短语是order by

格式: order by ordey_Item [ASC | desc] [,order_Item [ASC | desc]…]

说明:可按升序(ASC)或降序(desc)排序,允许按一列或多列排序

Eg16 按职工的工资值升序(降序)检索出全部职工信息.

Select * from 职工order by 工资 (desc)

升序时可省略, 默认为升序

Eg17 先按仓库升序,再按工资降序输出全部职工信息

Select * from 职工 order by 仓库号,工资 desc

注: order by 是对最终的查询结果进行排序,不可以在子查询中使用该短语.

6.简单的计算查询

SQL不仅具有一般的检索以能力,而且还有计算方式的检索,用于计算检索的函数有

COUNT---计数

SUM---求和

AVG---计算平均值

MAX---求最大值

MIN---求最小值

这些函数可用在SELECT短语中对查询结果进行计算.

Eg18 .找出供应商所在地的数目.

Select count (distinct 地址)from供应商

注:除非对关系中的元组个数进行计数,一般count函数应使用distinct.

Select count(*)from供应商

将给出供应商关系中的记录数.

Eg19 求支付的工资总数.

Select sum (工资) from 职工

包括重复值，不能使用 sum(distinct 工资)

Eg20 求北京和上海仓库职工的工资总和

select sum (工资) from 职工 where 仓库号 IN;

(select 仓库号 from 仓库 where 城市="北京" or 城市="上海")

Eg21 求所有职工的工资都多于1210元的仓库的平均面积

select avg(面积) from 仓库 where 仓库号 not in;

(select 仓库号 from 职工 where 工资<=1210)

在查询结果中包括没有职工的仓库WH4,若要排除可改写成

select avg(面积) from 仓库 where 仓库号 not in;

(select 仓库号 from 职工 where 工资<=1210);

AND 仓库号 IN (select 仓库号 from 职工)

Eg22 求在WH2仓库工作的职工的最高(低)工资值

Select max(min)(工资)from 职工 where 仓库号=“WH2”

7、分组与计算查询

在SQL select中可利用GROUP BY子句进行分组计算查询

格式：GROUP BY GroupColumn [,GroupColumn…]

[HAVING FilterCondition]

说明：可按一列或多列分组还可用Having 进一步限定分组条件

eg23 求每个仓库的职工的平均工资

select 仓库号,avg(工资) from 职工 group by 仓库号

Group by一般跟在where 之句之后，没有 where子句时，跟在from子句之后.

在分组查询时，有时要求分组满足某个条件时才检索，这时可以用HAVING子句来限定分组

Eg24 求至少有两个职工的每个仓库的平均工资

select 仓库号 count(*),AVG(工资) from 职工;

group by 仓库号 having count(*)>=2

注:

HAVING 子句总是跟在GROUP BY子句之后,不可以单独使用,Having 子句和where子句并不矛盾，在查询中先用where子句限定元组，然后进行分组，最后再用Having子句限定分组。

8.利用空值查询

SQL支持空值,同样可以利用空值进行查询,查询空值时要用is null 而= Null是无效的，因为空值不是一个确定的值,所以不能=这样的运算符进行比较

eg25 找出尚未确定供应商的订购号

select * from 订购单 where 供应商号 is null

Eg26 列出已经确定了供应商的订购单信息.

Select * from 订购单 where 供应商号 is not null

9、使用量词和谓词的查询

与嵌套查询或子查询有关的运算符,除了in和not in运算符外还有两类与子查询有关的运算符它们有以下两种格式.

(1)<表达式><比较运算符>[ANY | ALL | some](子查询)

(2)[NOT]EXISTS(子查询)

ANY ALL SOME 是量词,其中any 和some是同义词,在进行比较运算时只要子查询中有一行能使结果为真则结果就为 真,而ALL则要求子查询中的所有行都为真时,结果才为真.

Exists是谓词，Exists或NOT Exists是用来检查在子查询中是否有结果返回，即存在元组或不存在元组，其本身并没有进行任何运算或比较，只用来返回子查询结果。

Eg27 检索那些仓库中还没有职工的仓库信息.

Select * from 仓库 where not exists
(select * from 职工 where 仓库号=仓库.仓库号)

等价于

select * from 仓库 where 仓库号 NOT IN ;
（select 仓库号 from 职工）

Eg28 检索那些仓库中至少已经有一个职工的仓库的信息.

Select * from 仓库 where exists
(select * from 职工 where 仓库号=仓库.仓库号)

等价于

select * from 仓库 where 仓库号 in
(select 仓库号 from 职工)

Eg29.检索有职工的工资大于或等于WH1仓库中任何一名职工工资的仓库号

select distinct 仓库号 from 职工 where 工资>=;
any (select 工资 from 职工 where仓库号=“WH1”)
它等价于
select distinct 仓库号 from 职工 where 工资>=;
(select MIN(工资)from职工where仓库号=“WH1”
eg30.检索有职工的工资大于或等于WH1仓库中所有职工工资的仓库号
select distinct 仓库号 from 职工 where 工资>=;
(select 工资from 职工 where仓库号=“WH1”
等价于
select disinct 仓库号 from 职工 where 工资>=;
all (select max (工资)from职工where仓库号=“WH1”
10.集合的并运算
SQL支持集合的并（union)运算，即可以将两个select语句

的查询结果通过并运算合并成一个查询结果。进行并运算的两个查询结果必须具有相同的字段个数，并且对应字段的值出自同一个值域，即具有相同的数据类型和取值范围。

Eg31 如下语句的结果是城市为北京和上海的仓库信息。

```
Select * from 仓库 where 城市=“北京”
union select * from 仓库 where 城市=“上海”
```

11.SQL select 的几个特殊选项

(1)显示部分结果

格式:top nexpr [percent]

说明:nexpr是数字表达式,当不便percent时,nexpr是1至32767间的整数,说明显示前几个记录,当便用percent时,nexpr是0.01至99.99间的实数,说明显示结果前百分之几的记录.

注: top 短语要与order by短语同使用才有效

Eg32 显示工资最高的三位职工的信息.

Select * Top 3 from 职工 order by 工资 desc

Eg33 显示工资最低的那30%职工的信息.

Select * top 30 percent from 职工 order by 工资

(2)将查询结果放到数组中

格式: into array arrayname

说明: arrayname是任意数组变量名

eg34 将查询到的职工信息存放在数组tmp中

select * from 职工 into array tmp

tmp(1.1)	tmp(1.2)	tmp(1.3)
Wh2	E1	1220
Wh1	E3	1210
Wh2	E4	1250
Wh3	E6	1230
Wh1	E7	1250

(3)将查询结果存放在临时文件

格式: INTO CURSOR cursorname

说明: cursorname是临时文件名, 该临时文件是一个只读的.dbf文件, 当查询文件关闭时该文件将自动删除

Eg35 将查询到的职工信息存放在临时.dbf文件tmp中

select * from 职工 into cursor tmp

(4)将查询结果存放到永久表中

格式: into table / dbf tablename

Eg36 将工资最高的三位职工的信息存放到表highsal中.

Select * top 3 from 职工 into dbf highsal;

order by 工资

(5)将查询结果存放到文本文件.

格式: to file [additive]

说明: 扩展名为.txt的文本文件名,若使用addtive将结果追加在原文件的尾部否则将覆盖原文件.

Eg37 将上例查询结果存储在文本文件tmp.txt中

Select * top 3 from 职工 to [ADDitive];

order by 工资 desc

(6)将查询结果直接输出到打印机

格式: to printer [prompt]

12、超联接查询

（1）在新的SQL标准中支持两个新的关系联接运算符，一个是左联接（*=）和右联接（=*）：首先保证一个表中满足条件的结果表中，然后将满足联接条件的元组与另一个表的元组进行联接，不满足联接条件的则应将来自另一个表的属性值置为空值。

（2）在VF中不支持超联接运算“*=”和“=*”，VF中有专门的联接运算，下面给出SQL中超联接的部分语句格式:

select ……from table inner/left/right/ full jion table on join condition where ……

- INNER JOIN等价于JION,为普通的联接,在VF中称为内部联接.
- LEFT JOIN为左联接
- RIGHT JOIN为右联接
- FULL JOIN称为全联接
- ON condition指定联接条件

Eg38 内部联接,即只有满足条件的记录才出现在查询结果中.

select 仓库.仓库号,城市,面积,职工号,工资 from ;

仓库(inner) join 职工 on 仓库.仓库号=职工.仓库号

等价于

select 仓库.仓库号,城市,面积,职工号,工资 from 仓库,;

职工 where 仓库.仓库号=职工.仓库号