

1、查询所有记录：Select *通常用于快速查看表中的记录。当对表的结构无法确切记忆时，或要快速查看表中的记录时，使用 Select *是很方便的。

use 企业销售管理系统

Select * from 职工

2、投影查询：很多时候并不需要将所有列的数据都显示出来。投影查询就是允许用户显示所需要的列。假设显示职工表中职工号、姓名、工资信息，具体操作方法如下。

Select 职工号,姓名,工资 from 职工

在 Select 查询中，还可以按照实际的需要或自己的喜好安排显示列的顺序，如果要把姓名放到第一列，则代码如下：

Select 姓名, 职工号,工资 from 职工

3、条件查询：显示 wh1 或 wh2 仓库、性别不为女、工资在 1300~2100 之间的职工信息。具体操作如下：

Select * from 职工号 where (仓库号=' wh1' or 仓库号=' wh2')

And not 性别=' 女' and 工资>=1300 and 工资<=2100

在这里一定要注意，(仓库号=' wh1' or 仓库号=' wh2')一定要加括号，否则就不是原题的意思。

4、谓词 In 查询：在查找特定条件的数据时，如果条件较多，就需要用到多个 Or 运算符，以查找满足其中任一条的记录。但使用多个 Or 运算符，将使 Where 子句变得过于冗长，这时使用 In 就非常清楚。另外，在后面讲解的嵌套查询中，也必须使用 In 查询。下面以显示 zg1、zg2、zg11、zg15 的订购单信息为例具体讲解一下。

使用 Or 运算符

Select * from 订购单 where 职工号=' zg1' or 职工号=' zg2')

Or 职工号=' zg11' or 职工号=' zg15'

使用谓词 In

Use 企业销售管理系统

Select * from 订购单 where 职工号 in(' zg1' , ' zg2' , ' zg11' , ' zg15')

在 Select 语句中，还支持 In 与 Not 结合使用。假设显示职工号不为 zg1, zg2, zg11, zg15 的订购单信息，具体代码如下：

使用 Or 运算符

Select * from 订购单 where 职工号!=' zg1' or 职工号!=' zg2')

Or 职工号!=' zg11' or 职工号!=' zg15'

使用谓词 In

Use 企业销售管理系统

Select * from 订购单 where 职工号 not in(' zg1' , ' zg2' , ' zg11' , ' zg15')

5、模糊 Like 查询：有的时候，我们不清楚所要查询的信息，如显示职工信息，但不能确定该职工的姓名，只知道他姓名中含有“王”字，那该如何查找呢？SQL 中提供了 Like 关键字，使用 Like 进行了查询时，还要指定通配符。

通配符及其含义

通配符 含 义

% 包含 0 个或多个字符

_ (下画线) 包含一个字符

[] 指定范围 (如 [a-z])

[^] 不属于指定范围 ([^a-z])

在 SQL 语句中, 还支持 Like 与 Not 结合使用。假设显示姓名中不含有“王”字的职工信息, 具体代码如下:

```
Select * from 职工 where 姓名 not like '王%'
```

在该例中, 如果你知道姓名中含有“王”字, 并且姓名只有两个字, 那么可以进一步缩小查找范围, 具体代码如下:

```
Select * from 职工 where 姓名 not like '王_'
```

这样就可以查找到姓名中只有两个字, 并且姓为“王”的职工信息。

6、空值 NULL 查询: 在 SQL 语句中使用 IS NULL 来进行是否为空的判断。下面以显示性别为空的职工信息为例具体讲解一下。

```
Select * from 职工 where 性别 is null
```

假设显示性别不为空的职工信息, 具体代码如下:

```
Select * from 职工 where 性别 is not null
```

7、限制范围 Between...And 查询: 在数据医疗引擎查询中, 限制范围也是经常使用的一个条件。当然可以使用大于等于号、小于等于号和 And 运算符三者来完成范围的限制, 但使用 Between...And 结构, 会使 SQL 更清楚。下面以显示工资在 1300~1900 之间的职工信息为例具体讲解一下。

Between...And 语句

```
Select * from 职工 where 工资 Between 1300 And 1900
```

大于等于号、小于等于号与 And 运算符

```
Select * from 职工 where 工资>=1300 And 工资<=1900
```

在 SQL 语句中, 还支持 Between...And 与 Not 结合使用。假设显示工资不在 1300~1900 之间的职工信息, 具体代码如下:

Between...And 语句

```
Select * from 职工 where 工资 not Between 1300 And 1900
```

大于等于号、小于等于号与 And 运算符

```
Select * from 职工 where 工资<=1300 And 工资>=1900
```

8、消除重复字段数据 Distinct 查询: 在 SQL 投影查询中, 可能会有许多重复的数据, 使用关键字 Distinct 就可以从结果集中除去重复的数据。下面以显示职工的工资信息为例具体讲解一下:

显示有重复的数据

```
Select 工资 from 职工
```

显示无重复的数据

```
Select Distinct 工资 from 职工
```

9、单表嵌套查询: 就是外层 Select 语句与内层 Select 语句都来源于同一张表。下面以显示仓库面积大于 wh1 仓库面积的仓库信息为例来讲解一下。

首先利用 Select 语句求出 wh1 仓库的仓库面积, 然后再显示面积大于该值的仓库信息。

Select * from 仓库 where 面积>(select 面积 from 仓库 where 仓库号=' wh1')
嵌套查询中也可以带有逻辑运算符、谓词 In、Like、Between...And 结构。假设显示 wh1 或 wh2 仓库的工资不大于职工 zg16 工资的职工信息，并要求职工姓名不含有“平”字。

首先利用 Select 语句求出职工 zg16 的工资，然后再对姓名进行模糊查询。

Select * from 职工 where 工资<=(select 工资 from 职工 where 职工号=' zg16') and 姓名 not like '%平%' and 仓库号 in('wh1' , 'wh2')

10、多表嵌套查询：多表嵌套查询就是外层 Select 语句与内层 Select 语句来源于不同的表，但表之间要存在关系。这也是 SQL 语句中最重要的查询。下面以显示北京地区的职工信息为例来讲解一下。

首先利用 Select 语句求出北京地区的仓库号，然后利用仓库号再显示职工信息。

use 企业销售管理系统

Select * from 职工 where 仓库号 in(select 仓库号 from 仓库 where 城市=' 北京')

不仅可以实现两个表的嵌套，还可以实现多个表的嵌套，假设显示北京地区的职工的订购单信息。

首先利用 Select 语句得到北京地区的仓库号，然后利用得到的仓库号得到该仓库中的职工号，再利用得到的职工号得到职工的订购单信息。

Select * from 订购单 where 职工号 in(Select 职工号 from 职工 where 仓库号 in (select 仓库号 from 仓库 where 城市=' 北京'))

多表嵌套查询中也可以带有逻辑运算符、谓词 In、Like、Between、And 结构。假设显示北京地区的职工的订购单信息，并且职工的工资大于 1200，订购日期为 2003 年 9 月 3 日～2006 年 1 月 1 日。

Select * from 订购单 where 职工号 in(Select 职工号 from 职工 where 工资>1200 and 仓库号 in(select 仓库号 from 仓库 where 城市=' 北京'))and 订购日期 between '2003-09-03' and '2006-01-01'

11、单级排序查询：排序的关键字是 Order by，默认状态下是升序，关键字是 Asc。降序排列的关键字是 Desc。排序字段可以是数值型，也可以是字符型、日期时间型。下面以按工资从高到低显示职工信息为例来讲解一下。

Select * from 职工 Order by 工资 desc

在排序查询中还可以带有逻辑运算符、谓词 In、Like、Between、And 结构。假设显示 wh1 或 wh2 仓库的，工资不大于职工 zg16 工资的职工信息，并要求职工姓名不含有“平”字，按工资从低到高显示。

首先利用 Select 语句求出职工 zg16 的工资，然后再对姓名进行模糊查询，最后再排序。

Select * from 职工 where 工资<=(Select 工资 from 职工 where 职工号=' zg16') and 姓名 not like '%平%' and 仓库号 in ('wh1' , 'wh2') Order by 工资

12、多级排序查询：按照一列进行排序后，如果该列有重复的记录值，则重复记录值这部分就没有进行有效的排序，这就需要再附加一个字段，作为第二次排序的标准，对没有排开的记录进行再排列。下面以按工资从高到低显示职工信息为例来讲解一下。单级排序，按工资降序。

Select * from 职工 Order by 工资 desc

多级排序，先按工资降序，再按职工 ID 升序。

Select * from 职工 Order by 工资 desc,职工 ID asc

13、与别名一起使用的统计函数：在实际编程中，有时候需要知道所有记录某项值的总和、平均值、最大值等，这时就要用到统计函数查询。常用的统计函数共有 6 个。

统计函数及其意义

统计函数 含 义

COUNT(*) 统计选择的记录的个数

COUNT () 统计特定列中值的个数

SUM () 计算总和（必须是数值型字段）

AVG () 计算平均值（必须是数值型字段）

MAX () 确定最大值

MIN () 确定最小值

在使用统计函数时，还要注意 COUNT ()、SUM ()、AVG () 可以使用 DISTINCT 关键字，以在计算机中不包含重复的行。而对于 MAX ()、MIN ()、COUNT (*)，由于不会改变其结果，因此没有必要使用 DISTINCT。

利用 SELECT 语句显示统计值是没有列名的，下面以显示职工的最大值、最小值、工资总和、平均工资、职工人数为例来讲解一下。

```
Select max(工资),min(工资), SUM(工资),avg(工资),count(*) from 职工
```

这时会发现统计值没有列名，现在来给统计字段添加列名。

```
Select max(工资)as 最大工资,min(工资) as 最小工资, SUM(工资)as 工资总和,avg(工资)as 平均工资,count(*) as 职工人数 from 职工
```

在统计函数中还可以加条件，假设显示工资大于 1500 的职工的最大值、最小值、工资总和、平均工资、职工人数信息。

```
Select max(工资)as 最大工资,min(工资) as 最小工资, SUM(工资)as 工资总和,avg(工资)as 平均工资,count(*) as 职工人数 from 职工 where 工资>1500
```

14、带有统计函数的嵌套查询：还可以利用统计函数得到的数据作为 Select 查询语句的条件进行查询。下面以显示工资大于所有职工平均值的职工信息为例来讲解一下。

```
Select * from 职工 where 工资>(select avg(工资) from 职工)
```

在统计函数查询中可以带有比较运算符、逻辑运算符、In、Between...And 等。假设显示工资大于 wh1 或 wh2 仓库中职工平均工资，并且职工姓名不含有“亮”字的职工信息。

```
Select *from 职工 where 工资>(select avg(工资) from 职工 where 仓库号=' wh1' or 仓库号=' wh2' ) and 姓名 not like '%亮%'
```

15、统计函数字段显示在表字段中及运算字段：如果在职工表中添加一个职工平均工资值字段，这就对实际编程有很大的意义，即可以利用平均字段进行运算，大大提高编程速度。下面以在职工表中添加平均工资为例为讲解一下。

```
Select 职工.*, (select avg(工资) from 职工) as 平均工资 from 职工
```

在职工表中添加了一个平均工资字段，即在 SELECT 查询字段中添加了一个 SELECT 子查询。

在 SELECT 查询中，还可以计算出每名职工的工资与所有职工平均工资之差，具体操作如下：

```
Select 职工.*, (select avg(工资) from 职工) as 平均工资,工资-(select avg(工资) from 职工) as 平均工资的差 from 职工
```

16、分组查询：使用 GROUPBY 分组查询，要注意分组的标准一定要有意义，如显示不同仓库

的平均工资，显示不同职工的最大订单金额等。如果显示不同职工的平均工资那就没有意义了。下面显示不同仓库的平均工资、工资和、最大工资、最小工资、工资人数信息为例来讲解一下。

Select 仓库号,max(工资) as 最大工资, min(工资) as 最小工资, sum(工资) as 工资总和, avg(工资) as 平均工资, count(*) as 职工人数 from 职工 group by 仓库号
在分组查询中还可以带有逻辑运算符、谓词 In、Like、Between...And 结构。假设显示工资大于 zgl 职工工资的不同仓库的平均工资、工资和、最大工资、最小工资、工资人数信息。
首先利用 SELECT 语句求出职工 zgl 的工资，然后再分组显示统计信息。

Select 仓库号,max(工资) as 最大工资, min(工资) as 最小工资, sum(工资) as 工资总和, avg(工资) as 平均工资, count(*) as 职工人数 from 职工 where 工资>(Select 工资 from 职工 where 职工号='zgl') group by 仓库号

17、带有 Having 的分组查询：分组以前的条件，要用 Where 关键字，而分组之后的条件则使用 Having 关键字。下面以显示不同仓库的平均工资、工资和、最大工资、最小工资、工资人数信息，但要求该仓库平均工资大于 1760 为例来讲解一下。

Select 仓库号,max(工资) as 最大工资, min(工资) as 最小工资, sum(工资) as 工资总和, avg(工资) as 平均工资, count(*) as 职工人数 from 职工 group by 仓库号 having avg(工资)>1760

在分组查询的 Having 条件中还可以带有逻辑运算符、谓词 In、Like、Between...And 结构。假设显示不同仓库的平均工资、工资和、最大工资、最小工资、工资人数信息，但要求该仓库的平均工资大于 1760，最大工资与最小工资之差在 380~800 之间。

Select 仓库号,max(工资) as 最大工资, min(工资) as 最小工资, sum(工资) as 工资总和, avg(工资) as 平均工资, count(*) as 职工人数 from 职工 group by 仓库号 having avg(工资)>1760 and max(工资)-min(工资) between 380 and 800

18、带有分组的嵌套查询：还可以利用分组得到的数据作为 SELECT 查询语句的条件进行查询。下面以显示职工所在仓库的最大工资与最小工资之差在 380~800 之间的职工所在仓库信息为例为讲解一下。

先求出职工所在仓库的最大工资与最小工资之差在 380~800 之间的职工所在的仓库号，再利用这个仓库号显示仓库信息。

Select * from 仓库,where 仓库号 in(Select 仓库号 from 职工 group by 仓库号 having max(工资)-min(工资) between 380 and 800

19、内外层嵌套实现分组功能：利用 group by 分组显示统计字段信息，有很多字段不能显示。如显示不同仓库的平均工资信息，在这个查询中只能显示仓库号，不能显示职工号及职工的工资。因为不同仓库的平均工资，有几个仓库就显示几条平均工资，而每个仓库中的职工则有很多。

下面以显示不同职工经手订购单金额最大的订单信息为例来讲解一下。

在这里应该边求出每名职工订购单金额的最大值，边显示该职工所对应的这条订购单信息。具体代码如下：

Select a.* from 订购单 a where 金额=(select max(金额) from 订购单 where 职工号=a.职工号)

注意：上面语句中的订购单 a 是给这个订购单起的一个别名，大家可以把它看成就其实是订购单的一张复制的表，在这里运用它只是为了使内外层的嵌套更清晰、更容易理解。

20、分组查询字段显示在表字段中及运算字段：如果在职工表中添加职工所在仓库的平均工资值字段，这就对实际编程有很大的意义，即可以利用平均字段进行运算，大大提高编程速度。下面以在职工表中添加职工所在的仓库的平均工资字段为例来讲解一下。

```
Select a.*, (Select avg(工资) from 职工 where 仓库号=a. 仓库号) as 所在仓库的平均工资 from 职工 a
```

在 Select 查询中，还可以计算出每名职工的工资与职工所在仓库的平均工资之差，具体操作如下：

```
Select a.*, (Select avg(工资) from 职工 where 仓库号=a. 仓库号) as 所在仓库的平均工资, 工资- (Select avg(工资) from 职工 where 仓库号=a. 仓库号) as 所在仓库的平均工资之差 from 职工 a
```

21、Exists 谓词查询：其实 In 就是谓词查询，还有谓词 Exists 与 Exists，这两个谓词实现的功能是相同的，只是写法不同。In 多用于嵌套子查询语句中，而 Exists 多用于判断 Select 语句是否返回查询结果。常用量词有 Any、All 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，结果就为真。而 All 则要求子查询中所有行都使结果为真时，结果才为真。下面以显示那些仓库中还没有职工的仓库信息为例，讲解一下 In 谓词与 Exists 谓词在代码上的不同。

用 Exists 谓词实现：

```
Select * from 仓库 where not Exists(Select * from 职工 where 仓库号=仓库. 仓库号)
```

用 In 谓词实现：

```
Select * from 仓库 where 仓库号 not in(Select 仓库号 from 职工)
```

22、Any 量词查询：Any 量词查询，在进行比较运算时，只要子查询中有一行能使结果为真，则结果即为真。下面就以显示工资大于等于 wh2 仓库中任一名职工工资的职工信息为例，讲解一下 Any 量词的应用。

用 Any 量词实现：

```
Select * from 职工 where 工资>=any(Select 工资 from 职工 where 仓库号=' wh2' )
```

使用统计函数实现：

```
Select * from 职工 where 工资>=(Select min(工资) from 职工 where 仓库号=' wh2' )
```

23、All 量词查询：All 量词查询，要求子查询中所有行都使结果为真时，结果才为真。下面就以显示工资大于等于 wh1 仓库中所有职工工资的职工信息为例，讲解一下 All 量词的应用。

用 Any 量词实现：

```
Select * from 职工 where 工资>=All(Select 工资 from 职工 where 仓库号=' wh1' )
```

使用统计函数实现：

```
Select * from 职工 where 工资>=(Select MAX(工资) from 职工 where 仓库号=' wh1' )
```

前面讲解的量词查询都可以用统计函数代替，但有些查询是不能用统计函数代替的。下面以显示工资大于所有不同仓库的平均工资的职工信息为例为讲解一下。

```
Select * from 职工 where 工资>all(Select avg(工资) from 职工 group by 仓库号)
```

该题用统计函数就无法解决。

24、显示部分记录的 Top 查询：在编程中，有时只需显示满足条件的前几条记录，这时就可

以使用 Top 关键字。可以直接使用 Top 数字,显示指定条数记录;也可以使用 Top 数字 Percent,显示所有满足条件记录的前百分之几条记录。

下面以显示工资最高的前三条职工信息为例来讲解一下。

```
Select top3 * from 职工 order by 工资 desc
```

25、保存查询:利用 Into 语句可以把查询的结果保存成一张新表。下面就以备份“职工”表为例讲解一下保存查询。

Use 企业销售管理系统

```
Select * into 职工备份 from 职工
```

这样就生成一张新表,表名为“职工备份”。

26、集合的并运算:SQL 支持集合的并运算 (Union),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。为了进行并运算,要求这样的两个查询结果具有相同的字段个数,并且对应的字段的值要出自同一个值域,即具有相同的数据类型和取值范围。

```
Select * from 仓库 where 城市='北京'
```

```
Union
```

```
Select * from 仓库 where 城市='上海'
```

27、集合的交运算:SQL 支持集合的交运算 (Intersect),即可以将两个 Select 语句的查询结果通过交运算合并成一个查询结果。为了进行交运算,要求这样的两个查询结果具有相同的字段个数,并且对应的字段的值要出自同一个值域,即具有相同的数据类型和取值范围。

```
Select 仓库号 from 仓库
```

```
Intersect
```

```
Select 仓库号 from 职工
```

注意它们对集合的差运算不支持。

28、多表连接查询:多表连接查询就是将多个表中的数据结合到一起的查询,即连接操作可以在一个 Select 语句中完成从多个表中查找和处理数据。使用连接的列必须是可连接的,即它们具有相同的数据类型、相同的意义。使用连接的列,列名可以相同,也可以不同。

29、笛卡儿积:有的时候,如果连接条件没有设置好,或者没有设置连接条件,则结果将包含太多的行,这就是由笛卡儿积造成的。从原理上来讲,连接首先将形成表的笛卡儿积,即形成用于连接的表中所有的行的组合。下面以仓库表与职工表连接为例为讲解一下。

```
Select * from 职工, 仓库
```

这是从两张表中显示所有字段,所以字段有重复,并且产生了 90 条记录,原因是仓库表中有 5 条记录,职工表有 18 条记录,所以连接的结果是 $18 \times 5 = 90$ 条记录。用于连接的表越多,则笛卡儿积的结果将越大。

30、多表连接:在进行多表连接时,一定要注意连接条件,下面以仓库表与职工表连接为例来讲解一下多表连接。

```
Select * from 职工,仓库 where 职工.仓库号=仓库.仓库号
```

这时显示的记录就没有那么多了,只显示 18 条记录,并且发现两张表的仓库号是相对应的。假设显示工资大于 1500,面积大于 600 的城市与姓名信息。

Select 城市, 姓名 * from 职工, 仓库 where 职工. 仓库号=仓库. 仓库号 and 工资
工资>1500 and 面积>600

31、超连接查询：超连接查询也是连接查询，所以必须有两张或两张以上的表。超连接查询共有 4 种：内连接查询、左连接查询、右连接查询、全连接查询。使用超连接查询不仅可以把满足条件的记录显示出来，还可以把一部分不满足条件的记录以 NULL 显示出来。

1) 内连接查询：只有满足连接条件的记录才出现在查询结果集中。下面以显示面积大于 600 的仓库号、职工号、城市、面积、工资信息为例为讲解一下内连接：

超连接：

Select 仓库. 仓库号, 职工号, 城市, 面积, 工资 from 仓库 inner join 职工 on 职工. 仓库号=仓库. 仓库号 and 面积>600

普通连接：

Select 仓库. 仓库号, 职工号, 城市, 面积, 工资 from 职工, 仓库 where 职工. 仓库号=仓库. 仓库号 and 面积>600

2) 左连接查询：除满足连接条件的记录显示外，第一张表中不满足条件的记录也出现在查询结果集中。下面以显示面积大于 600 的仓库号、职工号、城市、面积、工资信息为例为讲解一下左连接。

Select 仓库. 仓库号, 职工号, 城市, 面积, 工资 from 仓库 left join 职工 on 职工. 仓库号=仓库. 仓库号 and 面积>600

3) 右连接查询：除满足连接条件的记录显示外，第二张表中不满足条件的记录也出现在查询结果集中。下面以显示面积大于 600 的仓库号、职工号、城市、面积、工资信息为例为讲解一下右连接。

Select 仓库. 仓库号, 职工号, 城市, 面积, 工资 from 仓库 right join 职工 on 职工. 仓库号=仓库. 仓库号 and 面积>600

4) 全连接查询：除满足连接条件的记录显示外，两张表中不满足条件的记录也出现在查询结果集中。下面以显示面积大于 600 的仓库号、职工号、城市、面积、工资信息为例为讲解一下全连接。

Select 仓库. 仓库号, 职工号, 城市, 面积, 工资 from 仓库 full join 职工 on 职工. 仓库号=仓库. 仓库号 and 面积>600

5) 多表超连接查询：前面讲解了两个表的超连接，那么如果是 3 张表或更多张表该如何实现超链接呢？下面以显示面积大于 600 的仓库号、职工号、城市、面积、工资、金额信息为例为讲解一下多表超连接。要显示的字段信息来源于 3 张表，仓库表、职工表、订购单表，这里实现仓库表与职工表的左连接，职工表与订购单表的右连接。

Select 仓库. 仓库号, 职工号, 城市, 面积, 工资, 金额 from 仓库 left join 职工 on 职工. 仓库号=仓库. 仓库号 right join 订购单 on 职工. 职工号=订购单. 职工号 and 面积>600

32、Insert 插入语句：向数据库表中插入数据是最常用的功能：

1) 指定所有列：

Insert into 仓库(仓库号, 城市, 面积, 创建时间) values ('wh6' , ' 郑州' , ' 800' , ' 2005-12-01')

按下键盘上的 F5 键，会显示(1 行受影响)，这表示已成功地插入一条记录。再在数据库引擎查询文档中输入“Select * from 仓库”，然后按下键盘上的 F5 键，就可以执行该 SQL 语句，这样就可以在查询结果上看到你插入的记录。在这里要注意，仓库 ID 是自动编号，所以不能插入。其实上面的代码还可以省略表后面的字段，这样也可以成功地插入记录。

Insert into 仓库 values ('wh6' , ' 郑州' , ' 800' , ' 2005-12-01')

2) 指定部分列:

下面以向仓库表中添加一条记录为例来讲解一下指定部分列。

Insert into 仓库(仓库号,面积) values ('wh7' ,800)

上述代码也可以这样来写:

Insert into 仓库 values ('wh7' ,null,800,null)

或

Insert into 仓库(仓库号,城市,面积,创建时间) values ('wh7' ,null,800,nul)

3) 带有子查询的插入语句: 使用带有子查询的插入语句, 首先要注意两个表的结构相同。下面还是用一个实例来讲解一下。

首先创建一张与仓库表结构相同的空表, 然后在数据库引擎查询文档中输入如下代码:

Select * into 仓库 blank from 仓库 where 面积<0

按下键盘上的 F5 键, 会显示 (0 行受影响), 这表示已成功地插入一条记录。再在数据库引擎查询文档中输入 "Select * from 仓库 blank", 然后按下键盘上的 F5 键, 就可以执行该 SQL 语句, 这样就产生了一张与仓库表结构相同的空表。

下面就可以利用带有子查询的插入语句向仓库 blank 中插入记录了。

Insert into 仓库 blank(仓库号,城市,面积,创建时间) select 仓库号,城市,面积,创建时间 from 仓库 where 面积>600

按下键盘上的 F5 键, 会显示 (6 行受影响), 这表示已成功地插入 6 行记录。再在数据库引擎查询文档中输入 "Select * from 仓库 blank", 然后按下键盘上的 F5 键, 就可以执行该 SQL 语句, 这样就可以利用 Insert 语句一次向表中插入多条记录。

33、Update 更新语句: 向数据库表中添加记录后, 可能需要对数据库进行修改。

1) 修改所有记录:

下面以为所有仓库的面积都增加 100 为例为讲解一个修改所有记录。

Update 仓库 set 面积=面积+100

2) 指定条件的更新语句:

下面以把城市在上海或北京的仓库的面积增加 100 为例, 来讲解一下指定条件的更新语句。

Use 企业销售管理系统

Update 仓库 set 面积=面积+100 where 城市=' 上海' or 城市=' 北京'

34、Delete 删除语句: 与添加记录一样, 删除记录也是数据库中一个重要的功能。

Delete 语句仅仅删除数据库表中的记录, 不会删除表。

Delete 语句不能删除单个列的值, 而是删除整个记录。

关键字 WHERE 可以有, 也可以没有。如果不加条件, 则删除整张表的所有记录; 如果有条件, 则删除表中满足条件的记录。

1) 指定条件的删除语句:

下面以删除仓库 blank 表中仓库面积大于 900 的记录为例, 来讲解一下指定条件的删除语句。

Delete from 仓库 blank 面积>900

2) 删除表中所有记录:

下面以删除仓库 blank 表中所有记录为例, 来讲解一下删除表中所有记录。

Delete from 仓库 blank

或

truncate from 仓库 blank

truncate 语句与 Delete 语句之间是有区别的，Delete 语句实际上是对表中每一个记录执行删除操作，而数据库执行随之可能产生的所有任务，如触发器；truncate 语句仅仅把所有记录一次删除，将会忽略定义的触发器，因此执行速度比 Delete 语句要快。由于 truncate 语句不处理删除引起的其他事件，因此使用它可能产生数据库完整性问题，所以对于有依赖关系的表，最好使用 Delete 语句删除。

