

转 MySQL常用命令

2013年10月10日 00:13:49 阅读数：1675

1.导出整个数据库

mysqldump -u 用户名 -p --default-character-set=latin1 数据库名 > 导出的文件名(数据库默认编码是latin1)

mysqldump -u wcnc -p smgp_apps_wcnc > wcnc.sql

2.导出一个表

mysqldump -u 用户名 -p 数据库名 表名> 导出的文件名

mysqldump -u wcnc -p smgp_apps_wcnc users> wcnc_users.sql

3.导出一个数据库结构

mysqldump -u wcnc -p -d --add-drop-table smgp_apps_wcnc > d:wcnc_db.sql

-d 没有数据 --add-drop-table 在每个create语句之前增加一个drop table

4.导入数据库

A:常用source 命令

进入mysql数据库控制台，

如mysql -u root -p

mysql>use 数据库

然后使用source命令，后面参数为脚本文件(如这里用到的.sql)

mysql>source wcnc_db.sql

B:使用mysqldump命令

mysqldump -u username -p dbname < filename.sql

C:使用mysql命令

mysql -u username -p -D dbname < filename.sql

一、启动与退出

1、进入MySQL：启动MySQL Command Line Client（MySQL的DOS界面），直接输入安装时的密码即可。此时的提示符是：mysql>

2、退出MySQL：quit或exit

二、库操作

1、创建数据库

命令：create database <数据库名>

例如：建立一个名为xhkdb的数据库

mysql> create database xhkdb;

2、显示所有的数据库

命令：show databases （注意：最后有个s）

mysql> show databases;

3、删除数据库

命令：drop database <数据库名>

例如：删除名为 xhkdb的数据库

mysql> drop database xhkdb;

4、连接数据库

命令：use <数据库名>

例如：如果xhkdb数据库存在，尝试存取它：

mysql> use xhkdb;

屏幕提示：Database changed

5、查看当前使用的数据库

mysql> select database();

6、当前数据库包含的表信息：

mysql> show tables; （注意：最后有个s）

三、表操作，操作之前应连接某个数据库

1、建表

命令：create table <表名> (<字段名1> <类型1> [...<字段名n> <类型n>]);

mysql> create table MyClass(

> id int(4) not null primary key auto_increment,

> name char(20) not null,

> sex int(4) not null default '0',

> degree double(16,2));

2、获取表结构

命令：desc 表名，或者show columns from 表名

mysql>DESCRIBE MyClass

mysql> desc MyClass;

mysql> show columns from MyClass;

3、删除表

命令：drop table <表名>

例如：删除表名为 MyClass 的表

mysql> drop table MyClass;

4、插入数据

命令：insert into <表名> [(<字段名1>[...<字段名n >])] values (值1)[(值n)]

例如，往表 MyClass中插入二条记录, 这两条记录表示：编号为1的名为Tom的成绩为96.45, 编号为2 的名为Joan 的成绩为82.99, 编号为3 的名为Wang 的成绩为96.5.

mysql> insert into MyClass values(1,'Tom',96.45),(2,'Joan',82.99), (2,'Wang', 96.59);

5、查询表中的数据

1)、查询所有行

命令：select <字段1, 字段2, ...> from < 表名 > where < 表达式 >

例如：查看表 MyClass 中所有数据

mysql> select * from MyClass;

2)、查询前几行数据

例如：查看表 MyClass 中前2行数据

mysql> select * from MyClass order by id limit 0,2;

或者：

```
mysql> select * from MyClass limit 0,2;
```

6、删除表中数据

命令：delete from 表名 where 表达式

例如：删除表 MyClass中编号为1 的记录

```
mysql> delete from MyClass where id=1;
```

7、修改表中数据：update 表名 set 字段=新值... where 条件

```
mysql> update MyClass set name='Mary' where id=1;
```

7、在表中增加字段：

命令：alter table 表名 add字段 类型 其他;

例如：在表MyClass中添加了一个字段passtest，类型为int(4)，默认值为0

```
mysql> alter table MyClass add passtest int(4) default '0'
```

8、更改表名：

命令：rename table 原表名 to 新表名;

例如：在表MyClass名字更改为YouClass

```
mysql> rename table MyClass to YouClass;
```

更新字段内容

```
update 表名 set 字段名 = 新内容
```

```
update 表名 set 字段名 = replace(字段名,'旧内容','新内容');
```

文章前面加入4个空格

```
update article set content=concat('    ',content);
```

字段类型

1. INT[(M)] 型：正常大小整数类型

2. DOUBLE[(M,D)] [ZEROFILL] 型：正常大小(双精密)浮点数字类型

3. DATE 日期类型：支持的范围是1000-01-01到9999-12-31。MySQL以YYYY-MM-DD格式来显示DATE值，但是允许你使用字符串或数字把值赋给DATE列

4. CHAR(M) 型：定长字符串类型，当存储时，总是是用空格填满右边到指定的长度

5. BLOB TEXT类型，最大长度为65535(2¹⁶-1)个字符。

6. VARCHAR型：变长字符串类型

5.导入数据库表

(1) 创建.sql文件

(2) 先产生一个库如auction.c:mysqlbin>mysqladmin -u root -p creat auction，会提示输入密码，然后成功创建。

(2) 导入auction.sql文件

```
c:mysqlbin>mysql -u root -p auction < auction.sql。
```

通过以上操作，就可以创建了一个数据库auction以及其中的一个表auction。

6. 修改数据库

(1) 在mysql的表中增加字段：

```
alter table dbname add column userid int(11) not null primary key auto_increment;
```

这样，就在表dbname中添加了一个字段userid，类型为int(11)。

7. mysql数据库的授权

```
mysql>grant select,insert,delete,create,drop
```

```
on *.* (或test.*/*user.*/*..)
```

to 用户名@localhost

identified by '密码'；

如：新建一个用户帐号以便可以访问数据库，需要进行如下操作：

```
mysql> grant usage
```

```
-> ON test.*
```

```
-> TO testuser@localhost；
```

Query OK, 0 rows affected (0.15 sec)

此后就创建了一个新用户叫：testuser，这个用户只能从localhost连接到数据库并可以连接到test 数据库。下一步，我们必须指定testuser这个用户可以执行哪些操作：

```
mysql> GRANT select, insert, delete,update
```

```
-> ON test.*
```

```
-> TO testuser@localhost；
```

Query OK, 0 rows affected (0.00 sec)

此操作使testuser能够在每一个test数据库中的表执行SELECT，INSERT和DELETE以及UPDATE查询操作。现在我们结束操作并退出MySQL客户程序：

```
mysql> exit
```

Bye9!

1:使用SHOW语句找出在服务器上当前存在什么数据库：

```
mysql> SHOW DATABASES;
```

2:2、创建一个数据库MYSQLDATA

```
mysql> Create DATABASE MYSQLDATA;
```

3:选择你所创建的数据库

```
mysql> USE MYSQLDATA; (按回车键出现Database changed 时说明操作成功！)
```

4:查看现在的数据库中是否存在什么表

```
mysql> SHOW TABLES;
```

5:创建一个数据库表

```
mysql> Create TABLE MYTABLE (name VARCHAR(20), sex CHAR(1));
```

6:显示表的结构：

```
mysql> DESCRIBE MYTABLE;
```

7:往表中加入记录

```
mysql> insert into MYTABLE values ("hyq","M");
```

8:用文本方式将数据装入数据库表中（例如D:/mysql.txt）

```
mysql> LOAD DATA LOCAL INFILE "D:/mysql.txt" INTO TABLE MYTABLE;
```

9:导入.sql文件命令（例如D:/mysql.sql）

```
mysql>use database;
```

```
mysql>source d:/mysql.sql;
```

10:删除表

```
mysql>drop TABLE MYTABLE;
```

11:清空表

```
mysql>delete from MYTABLE;
```

12:更新表中数据

```
mysql>update MYTABLE set sex="f" where name="hyq";
```

以下是无意中在网络看到的使用MySQL的管理心得，

摘自: http://www1.xjtusky.com/article/htmldata/2004_12/3/57/article_1060_1.html

在windows中MySQL以服务形式存在,在使用前应确保此服务已经启动,未启动可用net start mysql命令启动。而Linux中启动时可用"/etc/rc.d/init.d/mysqld start"命令,注意启动者应具有管理员权限。

刚安装好的MySQL包含一个含空密码的root帐户和一个匿名帐户,这是很大的安全隐患,对于一些重要的应用我们应将安全性尽可能提高,在这里应把匿名帐户删除、 root帐户设置密码,可用如下命令进行:

```
use mysql;
delete from User where User="";
update User set Password=PASSWORD('newpassword') where User='root';
如果要对用户所用的登录终端进行限制,可以更新User表中相应用户的Host字段,在进行了以上更改后应重新启动数据库服务,此时登录时可用如下类似命令:
mysql -uroot -p;
mysql -uroot -pnewpassword;
mysql mydb -uroot -p;
mysql mydb -uroot -pnewpassword;
```

上面命令参数是常用参数的一部分,详细情况可参考文档。此处的mydb是要登录的数据库的名称。

在进行开发和实际应用中,用户不应该只用root用户进行连接数据库,虽然使用root用户进行测试时很方便,但会给系统带来重大安全隐患,也不利于管理技术的提高。我们给一个应用中使用的用户赋予最恰当的数据库权限。如一个只进行数据插入的用户不应赋予其删除数据的权限。MySQL的用户管理是通过 User表来实现的,添加新用户常用的方法有两个,一是在User表插入相应的数据行,同时设置相应的权限;二是通过GRANT命令创建具有某种权限的用户。其中GRANT的常用用法如下:

```
grant all on mydb.* to NewUserName@HostName identified by "password";
grant usage on *.* to NewUserName@HostName identified by "password";
grant select,insert,update on mydb.* to NewUserName@HostName identified by "password";
grant update,delete on mydb.TestTable to NewUserName@HostName identified by "password";
```

若要给此用户赋予他在相应对象上的权限的管理能力,可在GRANT后面添加WITH GRANT OPTION选项。而对于用插入User表添加的用户,Password字段应用PASSWORD 函数进行更新加密,以防不轨之人窃看密码。对于那些已经不用的用户应给予清除,权限过界的用户应及时回收权限,回收权限可以通过更新User表相应字段,也可以使用REVOKE操作。

下面给出本人从其它资料(www.cn-java.com)获得的对常用权限的解释:

全局管理权限:

FILE: 在MySQL服务器上读写文件。

PROCESS: 显示或杀死属于其它用户的服务线程。

RELOAD: 重载访问控制表,刷新日志等。

SHUTDOWN: 关闭MySQL服务。

数据库/数据表/数据列权限:

Alter: 修改已存在的数据表(例如增加/删除列)和索引。

Create: 建立新的数据库或数据表。

Delete: 删除表的记录。

Drop: 删除数据表或数据库。

INDEX: 建立或删除索引。

Insert: 增加表的记录。

Select: 显示/搜索表的记录。

Update: 修改表中已存在的记录。

特别的权限:

ALL: 允许做任何事(和root一样)。

USAGE: 只允许登录--其它什么也不允许做。

MySQL常用命令

有很多朋友虽然安装好了mysql但却不知如何使用它。在这篇文章中我们就从连接MySQL、修改密码、增加用户等方面来学习一些MySQL的常用命令。

有很多朋友虽然安装好了mysql但却不知如何使用它。在这篇文章中我们就从连接MySQL、修改密码、增加用户等方面来学习一些MySQL的常用命令。

一、连接MySQL

格式: mysql -h主机地址 -u用户名 -p用户密码

1、例1:连接到本机上的MySQL

首先在打开DOS窗口,然后进入目录 mysqlbin,再键入命令mysql -uroot -p,回车后提示你输密码,如果刚安装好MySQL,超级用户root是没有密码的,故直接回车即可进入到MySQL中了,MySQL的提示符是:mysql>

2、例2:连接到远程主机上的MySQL

假设远程主机的IP为:110.110.110.110,用户名为root,密码为abcd123。则键入以下命令:

```
mysql -h110.110.110.110 -uroot -pabcd123
```

(注:u与root可以不用加空格,其它也一样)

3、退出MySQL命令: exit (回车)

二、修改密码

格式: mysqladmin -u用户名 -p旧密码 password 新密码

1、例1:给root加个密码ab12。首先在DOS下进入目录mysqlbin,然后键入以下命令

```
mysqladmin -uroot -password ab12
```

注:因为开始时root没有密码,所以-p旧密码一项就可以省略了。

2、例2:再将root的密码改为djg345

```
mysqladmin -uroot -pab12 password djg345
```

MySQL常用命令(下)

一、操作技巧

1、如果你打命令时,回车后发现忘记加分号,你无须重打一遍命令,只要打个分号回车就可以了。也就是说你可以把一个完整的命令分成几行来打,完后用分号作结束标志就OK。

2、你可以使用光标上下键调出以前的命令。但以前我用过的一个MySQL旧版本不支持。我现在用的是mysql-3.23.27-beta-win。

二、显示命令

1、显示数据库列表。

```
show databases;
```

刚开始时才两个数据库:mysql和test。mysql库很重要它里面有MySQL的系统信息,我们改密码和新增用户,实际上就是用这个库进行操作。

2、显示库中的数据表:

```
use mysql; // 打开库,学过FOXBASE的一定不会陌生吧
```

show tables;

3、显示数据表的结构：

describe 表名;

4、建库：

create database 库名;

5、建表：

use 库名；

create table 表名 (字段设定列表)；

6、删库和删表：

drop database 库名;

drop table 表名；

7、将表中记录清空：

delete from 表名;

8、显示表中的记录：

select * from 表名;

三、一个建库和建表以及插入数据的实例

drop database if exists school; //如果存在SCHOOL则删除

create database school; //建立库SCHOOL

use school; //打开库SCHOOL

create table teacher //建立表TEACHER

(

id int(3) auto_increment not null primary key,

name char(10) not null,

address varchar(50) default '深圳',

year date

); //建表结束

//以下为插入字段

insert into teacher values('','glchengang','深圳一中','1976-10-10');

insert into teacher values('','jack','深圳一中','1975-12-23');

注：在建表中（1）将ID设为长度为3的数字字段:int(3)并让它每个记录自动加一:auto_increment并不能为空:not null而且让他成为主字段primary key

（2）将NAME设为长度为10的字符字段

（3）将ADDRESS设为长度50的字符字段，而且缺省值为深圳。varchar和char有什么区别呢，只有等以后的文章再说了。

（4）将YEAR设为日期字段。

如果你在mysql提示符键入上面的命令也可以，但不方便调试。你可以将以上命令原样写入一个文本文件中假设为school.sql，然后复制到c:\下，并在DOS状态进入目录\mysql\bin，然后键入以下命令：

mysql -uroot -p密码 < c:\school.sql

如果成功，空出一行无任何显示；如有错误，会有提示。（以上命令已经调试，你只要将//的注释去掉即可使用）。

四、将文本数据转到数据库中

1、文本数据应符合的格式：字段数据之间用tab键隔开，null值用\来代替。

例：

3 rose 深圳二中 1976-10-10

4 mike 深圳一中 1975-12-23

2、数据传入命令 load data local infile "文件名" into table 表名;

注意：你最好将文件复制到\mysql\bin目录下，并且要先用use命令打表所在的库。

五、备份数据库：（命令在DOS的\mysql\bin目录下执行）

mysqldump --opt school>school.bbb

注释:将数据库school备份到school.bbb文件，school.bbb是一个文本文件，文件名任取，打开看看你会有新发现。

一.SELECT语句的完整语法为：

SELECT[ALL|DISTINCT|DISTINCTROW|TOP]

{*[table.*]|[table.]field1[AS alias1][,[table.]field2[AS alias2][,...]]}

FROM tableexpression[,...][IN externaldatabase]

[WHERE...]

[GROUP BY...]

[HAVING...]

[ORDER BY...]

[WITH OWNERACCESS OPTION]

说明：

用中括号([])括起来的部分表示是可选的，用大括号({})括起来的部分是表示必须从中选择其中的一个。

1 FROM子句

FROM 子句指定了SELECT语句中字段的来源。FROM子句后面是包含一个或多个的表达式(由逗号分开)，其中的表达式可为单一表名称、已保存的查询或由 INNER JOIN、LEFT JOIN 或 RIGHT JOIN 得到的复合结果。如果表或查询存储在外部数据库，在IN 子句之后指明其完整路径。

例：下列SQL语句返回所有有定单的客户：

SELECT OrderID, Customer.customerID

FROM Orders Customers

WHERE Orders.CustomerID=Customers.CustomeersID

2 ALL、DISTINCT、DISTINCTROW、TOP谓词

(1) ALL 返回满足SQL语句条件的所有记录。如果没有指明这个谓词，默认为ALL。

例：SELECT ALL FirstName, LastName

FROM Employees

(2) DISTINCT 如果有多个记录的选择字段的数据相同，只返回一个。

(3) DISTINCTROW 如果有重复的记录，只返回一个

(4) TOP显示查询头尾若干记录。也可返回记录的百分比，这是要用 TOP N PERCENT子句（其中N 表示百分比）

例：返回5%定货额最大的定单

```
SELECT TOP 5 PERCENT*
FROM [ Order Details]
ORDER BY UnitPrice*Quantity*(1-Discount) DESC
```

3 用 AS 子句为字段取别名

如果想为返回的列取一个新的标题，或者，经过对字段的计算或总结之后，产生了一个新的值，希望把它放到一个新的列里显示，则用AS保留。

例：返回FirstName字段取别名为NickName

```
SELECT FirstName AS NickName ,LastName ,City
FROM Employees
```

例：返回新的一列显示库存价值

```
SELECT ProductName ,UnitPrice ,UnitsInStock ,UnitPrice*UnitsInStock AS valueInStock
FROM Products
```

二 .WHERE 子句指定查询条件

1 比较运算符

比较运算符 含义

= 等于

> 大于

< 小于

>= 大于等于

<= 小于等于

<> 不等于

!> 不大于

!< 不小于

例：返回96年1月的定单

```
SELECT OrderID, CustomerID, OrderDate
FROM Orders
```

```
WHERE OrderDate>=#1/1/96# AND OrderDate<=#1/30/96#
```

注意：

Microsoft JET SQL 中，日期用'#'定界。日期也可以用Datevalue()函数来代替。在比较字符型的数据时，要加上单引号"，尾空格在比较中被忽略。

例：

```
WHERE OrderDate>=#96-1-1#
```

也可以表示为：

```
WHERE OrderDate>Datevalue('1/1/96')
```

使用 NOT 表达式求反。

例：查看96年1月1日以后的定单

```
WHERE Not OrderDate<=#1/1/96#
```

2 范围（BETWEEN 和 NOT BETWEEN）

BETWEEN ...AND...运算符指定了要搜索的一个闭区间。

例：返回96年1月到96年2月的定单。

```
WHERE OrderDate Between #1/1/96# And #2/1/96#
```

3 列表（IN ， NOT IN）

IN 运算符用来匹配列表中的任何一个值。IN子句可以代替用OR子句连接的一连串的条件。

例：要找出住在 London、 Paris或Berlin的所有客户

```
SELECT CustomerID, CompanyName, ContactName, City
FROM Customers
```

```
WHERE City In('London',' Paris',' Berlin')
```

4 模式匹配(LIKE)

LIKE运算符检验一个包含字符串数据的字段值是否匹配一指定模式。

LIKE运算符里使用的通配符

通配符 含义

? 任何一个单一的字

* 任意长度的字符

0-9之间的单一数字

[字符列表] 在字符列表里的任一值

[! 字符列表] 不在字符列表里的任一值

- 指定字符范围，两边的值分别为其上下限

例：返回邮政编码在（171）555-0000到（171）555-9999之间的客户

```
SELECT CustomerID ,CompanyName,City,Phone
FROM Customers
```

```
WHERE Phone Like '(171)555-####'
```

LIKE运算符的一些样式及含义

样式 含义 不符合

LIKE 'A*' A后跟任意长度的字符 Bc,c255

LIKE '5*]' 5*5 555

LIKE '5?5' 5与5之间有任何一个字符 55,5wer5

LIKE '5###5' 5235, 5005 5kd5,5346

LIKE '[a-z]' a-z间的任意一个字符 5,%

LIKE '[!0-9]' 非0-9间的任意一个字符 0,1

LIKE '[]' 1,*

三 .用ORDER BY子句排序结果

ORDER子句按一个或多个（最多16个）字段排序查询结果，可以是升序（ASC）也可以是降序（DESC），缺省是升序。ORDER子句通常放在SQL语句的最后。

ORDER子句中定义了多个字段，则按照字段的先后顺序排序。

例：

```
SELECT ProductName,UnitPrice, UnitInStock
FROM Products
```

```
ORDER BY UnitInStock DESC , UnitPrice DESC, ProductName
```

ORDER BY 子句中可以用字段在选择列表中的位置号代替字段名，可以混合字段名和位置号。

例：下面的语句产生与上列相同的效果。

```
SELECT ProductName,UnitPrice, UnitInStock
FROM Products
```

```
ORDER BY 1 DESC , 2 DESC,3
```

四 .运用连接关系实现多表查询

例：找出同一个城市中供应商和客户的名字

```
SELECT Customers.CompanyName, Suppliers.ComPany.Name
FROM Customers, Suppliers
```

```
WHERE Customers.City=Suppliers.City
```

例：找出产品库存量大于同一种产品的定单的数量的产品和定单

```
SELECT ProductName,OrderID, UnitInStock, Quantity
FROM Products, [Order Deals]
WHERE Product.productID=[Order Details].ProductID
AND UnitsInStock>Quantity
```

另一种方法是用 Microsof JET SQL 独有的 JNNER JOIN 语法：

```
FROM table1 INNER JOIN table2
```

```
ON table1.field1 comparision table2.field2
```

其中comparision 就是前面WHERE子句用到的比较运算符。

```
SELECT FirstName,lastName,OrderID,CustomerID,OrderDate
FROM Employees
```

```
INNER JOIN Orders ON Employees.EmployeeID=Orders.EmployeeID
```

注意：

INNER JOIN不能连接Memo OLE Object Single Double 数据类型字段。

在一个JOIN语句中连接多个ON子句

语法：

```
SELECT fields
```

```
FROM table1 INNER JOIN table2
```

```
ON table1.field1 compopr table2.field1 AND
```

```
ON table1.field2 compopr table2.field2 OR
```

```
ON table1.field3 compopr table2.field3
```

也可以

```
SELECT fields
```

```
FROM table1 INNER JOIN
```

```
(table2 INNER JOIN [( ]table3
```

```
[INNER JOER] [( ]tablex[INNER JOIN]
```

```
ON table1.field1 compopr table2.field1
```

```
ON table1.field2 compopr table2.field2
```

```
ON table1.field3 compopr table2.field3
```

外部连接返回更多记录，在结果中保留不匹配的记录，不管存不存在满足条件的记录都要返回另一侧的所有记录。

```
FROM table [LEFT|RIGHT]JOIN table2
```

```
ON table1.field1comparision table.field2
```

用左连接来建立外部连接，在表达式的左边的表会显示其所有的数据

例：不管有没有定货量，返回所有商品

```
SELECT ProductName ,OrderID
```

```
FROM Products
```

```
LEFT JOIN Orders ON Products.PrductsID=Orders.ProductID
```

右连接与左连接的差别在于：不管左侧表里有没有匹配的记录，它都从左侧表中返回所有记录。

例：如果想了解客户的信息，并统计各个地区的客户分布，这时可以用一个右连接，即使某个地区没有客户，也要返回客户信息。

空值不会相互匹配，可以通过外连接才能测试被连接的某个表的字段是否有空值。

```
SELECT *
```

```
FROM talbe1
```

```
LEFT JOIN table2 ON table1.a=table2.c
```

1 连接查询中使用Iif函数实现以0值显示空值

Iif表达式： Iif(IsNull(Amount,0,Amout)

例：无论定货大于或小于¥ 50，都要返回一个标志。

```
Iif([Amount]>50,?Big order?,?Small order?)
```

五. 分组和总结查询结果

在SQL的语法里，GROUP BY和HAVING子句用来对数据进行汇总。GROUP BY子句指明了按照哪几个字段来分组，而将记录分组后，用HAVING子句过滤这些记录。

GROUP BY 子句的语法

```
SELECT fiddlist
```

```
FROM table
```

```
WHERE criteria
```

```
[GROUP BY groupfieldlist [HAVING groupcriteria]]
```

注：Microsoft Jet数据库 Jet 不能对备注或OLE对象字段分组。

GROUP BY字段中的Null值以备分组但是不能被省略。

在任何SQL合计函数中不计算Null值。

GROUP BY子句后最多可以带有十个字段，排序优先级按从左到右的顺序排列。

例：在'WA'地区的雇员表中按头衔分组后，找出具有同等头衔的雇员数目大于1人的所有头衔。

```
SELECT Title ,Count(Title) as Total
```

```
FROM Employees
```

```
WHERE Region = 'WA'
```

```
GROUP BY Title
```

```
HAVING Count(Title)>1
```

JET SQL 中的聚积函数

聚集函数 意义

SUM () 求和

AVG () 平均值

COUNT () 表达式中记录的数目

COUNT (*) 计算记录的数目

MAX 最大值

MIN 最小值

VAR 方差

STDEV 标准误差

FIRST 第一个值

LAST 最后一个值

六. 用Parameters声明创建参数查询

Parameters声明的语法:

```
PARAMETERS name datatype[,name datatype[, ...]]
```

其中name 是参数的标志符,可以通过标志符引用参数.

Datatype说明参数的数据类型.

使用时要把PARAMETERS 声明置于任何其他语句之前.

例:

```
PARAMETERS[Low price] Currency,[Beginning date]datetime
```

```
SELECT OrderID ,OrderAmount
```

```
FROM Orders
WHERE OrderAmount>[low price]
AND OrderDate>=[Beginning date]
```

七. 功能查询

所谓功能查询,实际上是一种操作查询,它可以对数据库进行快速高效的操作.它以选择查询为目的,挑选出符合条件的数据,再对数据进行批处理.功能查询包括更新查询,删除查询,添加查询,和生成表查询.

1 更新查询

UPDATE子句可以同时更改一个或多个表中的数据.它也可以同时更改多个字段的值.

更新查询语法:

```
UPDATE 表名
SET 新值
WHERE 准则
例:英国客户的定货量增加5%,货运量增加3%
```

```
UPDATE OEDERS
SET OrderAmount = OrderAmount *1.1
Freight = Freight*1.03
WHERE ShipCountry = 'UK'
```

2 删除查询

DELETE子句可以使用户删除大量的过时的或冗于的数据.

注:删除查询的对象是整个记录.

DELETE子句的语法:

```
DELETE [表名.*]
FROM 来源表
WHERE 准则
例:要删除所有94年前的定单
```

```
DELETE *
FROM Orders
WHERE OrderData<#94-1-1#
```

3 追加查询

INSERT子句可以将一个或一组记录追加到一个或多个表的尾部.

INTO 子句指定接受新记录的表

valueS 关键字指定新记录所包含的数据值.

INSERT 子句的语法:

```
INSETR INTO 目的表或查询(字段1,字段2,...)
```

```
valueS(数值1,数值2,...)
```

例:增加一个客户

```
INSERT INTO Employees(FirstName,LastName,title)
valueS('Harry','Washington','Trainee')
```

4 生成表查询

可以一次性地把所有满足条件的记录拷贝到一张新表中.通常制作记录的备份或副本或作为报表的基础.

SELECT INTO子句用来创建生成表查询语法:

```
SELECT 字段1,字段2,...
```

```
INTO 新表[IN 外部数据库]
```

```
FROM 来源数据库
```

```
WHERE 准则
```

例:为定单制作一个存档备份

```
SELECT *
INTO OrdersArchive
FROM Orders
```

八. 联合查询

UNION运算可以把多个查询的结果合并到一个结果集里显示.

UNION运算的一般语法:

```
[表]查询1 UNION [ALL]查询2 UNION ...
```

例:返回巴西所有供给商和客户的名字和城市

```
SELECT CompanyName,City
FROM Suppliers
WHERE Country = 'Brazil'
UNION
```

```
SELECT CompanyName,City
FROM Customers
WHERE Country = 'Brazil'
```

注:

缺省的情况下,UNION子句不返回重复的记录.如果想显示所有记录,可以加ALL选项

UNION运算要求查询具有相同数目的字段.但是,字段数据类型不必相同.

每一个查询参数中可以使用GROUP BY 子句 或 HAVING 子句进行分组.要想以指定的顺序来显示返回的数据,可以在最后一个查询的尾部使用ORDER BY子句.

九. 交叉查询

交叉查询可以对数据进行总和,平均,计数或其他总和和算法的计算,这些数据通过两种信息进行分组:一个显示在表的左部,另一个显示在表的顶部.

Microsoft Jet SQL 用TRANSFROM语句创建交叉表查询语法:

```
TRANSFORM aggfunction
SELECT 语句
GROUP BY 子句
PIVOT pivotfield[IN(value1 [,value2[...]]) ]
Aggfounction指SQL累积函数,
SELECT语句选择作为标题的的字段,
GROUP BY 分组
```

说明:

Pivotfield 在查询结果集中创建列标题时用的字段或表达式,用可选的IN子句限制它的取值.

value代表创建列标题的固定值.

例:显示在1996年里每一季度每一位员工所接的定单的数目:

```
TRANSFORM Count(OrderID)
SELECT FirstName&''&LastName AS FullName
FROM Employees INNER JOIN Orders
ON Employees.EmployeeID = Orders.EmployeeID
WHERE DatePart("yyyy",OrderDate)= '1996'
GROUP BY FirstName&''&LastName
ORDER BY FirstName&''&LastName
```

POVOT DatePart("q",OrderDate)&季度'

十.子查询

子查询可以理解为 套查询.子查询是一个SELECT语句.

1 表达式的值与子查询返回的单一值做比较

语法:

表达式 comparison [ANY|ALL|SOME](子查询)

说明:

ANY 和SOME谓词是同义词,与比较运算符(=,<,>,<=,>=)一起使用.返回一个布尔值True或 False.ANY的意思是,表达式与子查询返回的一系列的值逐一比较,只要其中的一次比较产生True结果,ANY测试的返回 True值(既WHERE子句的结果),对应于该表达式的当前记录将进入主查询的结果中.ALL测试则要求表达式与子查询返回的一系列的值都比较都产生 True结果,才回返回True值.

例:主查询返回单价比任何一个折扣大于等于25%的产品的单价要高的所有产品

```
SELECT * FROM Products
```

```
WHERE UnitPrice>ANY
```

```
(SELECT UnitPrice FROM[Order Details] WHERE Discount>0.25)
```

2 检查表达式的值是否匹配子查询返回的一组值的某个值

语法:

[NOT]IN(子查询)

例:返回库存价值大于等于1000的产品.

```
SELECT ProductName FROM Products
```

```
WHERE ProductID IN
```

```
(SELECT ProdoctID FROM [Order DEtails]
```

```
WHERE UnitPrice*Quantity>= 1000)
```

3检测子查询是否返回任何记录

语法:

[NOT]EXISTS (子查询)

例:用EXISTS检索英国的客户

```
SELECT ComPanyName,ContactName
```

```
FROM Orders
```

```
WHERE EXISTS
```

```
(SELECT *
```

```
FROM Customers
```

```
WHERE Country = 'UK' AND
```

```
Customers.CustomerID= Orders.CustomerID)
```

文章标签:

mysql

命令

控制台

管理

个人分类: [J2EE](#)

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com