

一、基础

1、说明：创建数据库

```
CREATE DATABASE database-name
```

2、说明：删除数据库

```
drop database dbname
```

3、说明：备份 sql server

```
--- 创建 备份数据的 device
```

```
USE master
```

```
EXEC sp_addumpdevice 'disk', 'testBack', 'c:\mssql7backup\MyNwind_1.dat'
```

```
--- 开始 备份
```

```
BACKUP DATABASE pubs TO testBack
```

4、说明：创建新表

```
create table tabname(col1 type1 [not null] [primary key], col2 type2 [not null],...)
```

根据已有的表创建新表：

A: create table tab_new like tab_old (使用旧表创建新表)

B: create table tab_new as select col1,col2... from tab_old definition only

5、说明：删除新表

```
drop table tabname
```

6、说明：增加一个列

```
Alter table tabname add column col type
```

注：列增加后将不能删除。DB2 中列加上后数据类型也不能改变，唯一能改变的是增加 varchar 类型的长度。

7、说明：添加主键： Alter table tabname add primary key(col)

说明：删除主键： Alter table tabname drop primary key(col)

8、说明：创建索引： create [unique] index idxname on tabname(col...)

删除索引： drop index idxname

注：索引是不可更改的，想更改必须删除重新建。

9、说明：创建视图： create view viewname as select statement

删除视图： drop view viewname

10、说明：几个简单的基本的 sql 语句

选择： select * from table1 where 范围

插入： insert into table1(field1,field2) values(value1,value2)

删除： delete from table1 where 范围

更新： update table1 set field1=value1 where 范围

查找： select * from table1 where field1 like ' %value1%' ---like 的语法很精妙，查资料！

排序： select * from table1 order by field1,field2 [desc]

总数： select count as totalcount from table1

求和: `select sum(field1) as sumvalue from table1`

平均: `select avg(field1) as avgvalue from table1`

最大: `select max(field1) as maxvalue from table1`

最小: `select min(field1) as minvalue from table1`

11、说明: 几个高级查询运算词

A: UNION 运算符

UNION 运算符通过组合其他两个结果表 (例如 TABLE1 和 TABLE2) 并消去表中任何重复行而派生出一个结果表。当 ALL 随 UNION 一起使用时 (即 UNION ALL), 不消除重复行。两种情况下, 派生表的每一行不是来自 TABLE1 就是来自 TABLE2。

B: EXCEPT 运算符

EXCEPT 运算符通过包括所有在 TABLE1 中但不在 TABLE2 中的行并消除所有重复行而派生出一个结果表。当 ALL 随 EXCEPT 一起使用时 (EXCEPT ALL), 不消除重复行。

C: INTERSECT 运算符

INTERSECT 运算符通过只包括 TABLE1 和 TABLE2 中都有的行并消除所有重复行而派生出一个结果表。当 ALL 随 INTERSECT 一起使用时 (INTERSECT ALL), 不消除重复行。

注: 使用运算词的几个查询结果行必须是一致的。

12、说明: 使用外连接

A、left (outer) join:

左外连接 (左连接): 结果集几包括连接表的匹配行, 也包括左连接表的所有行。

SQL: `select a.a, a.b, a.c, b.c, b.d, b.f from a LEFT OUT JOIN b ON a.a = b.c`

B: right (outer) join:

右外连接 (右连接): 结果集既包括连接表的匹配连接行, 也包括右连接表的所有行。

C: full/cross (outer) join:

全外连接: 不仅包括符号连接表的匹配行, 还包括两个连接表中的所有记录。

12、分组: Group by:

一张表, 一旦分组 完成后, 查询后只能得到组相关的信息。

组相关的信息: (统计信息) count, sum, max, min, avg 分组的标准)

在 SQLServer 中分组时: 不能以 text, ntext, image 类型的字段作为分组依据

在 selecte 统计函数中的字段, 不能和普通的字段放在一起;

13、对数据库进行操作:

分离数据库: `sp_detach_db`; 附加数据库: `sp_attach_db` 后接表明, 附加需要完整的路径名

14. 如何修改数据库的名称:

`sp_renamedb 'old_name', 'new_name'`

二、提升

1、说明：复制表(只复制结构,源表名: a 新表名: b) (Access 可用)

法一: `select * into b from a where 1<<1` (仅用于 SQLServer)

法二: `select top 0 * into b from a`

2、说明：拷贝表(拷贝数据,源表名: a 目标表名: b) (Access 可用)

`insert into b(a, b, c) select d,e,f from b;`

3、说明：跨数据库之间表的拷贝(具体数据使用绝对路径) (Access 可用)

`insert into b(a, b, c) select d,e,f from b in '具体数据库' where 条件`

例子: `..from b in ' &Server.MapPath(".") &"\data.mdb" &"' where..`

4、说明：子查询(表名 1: a 表名 2: b)

`select a,b,c from a where a IN (select d from b)` 或者: `select a,b,c from a where a IN (1,2,3)`

5、说明：显示文章、提交人和最后回复时间

`select a.title,a.username,b.adddate from table a, (select max(adddate) adddate from table where table.title=a.title) b`

6、说明：外连接查询(表名 1: a 表名 2: b)

`select a.a, a.b, a.c, b.c, b.d, b.f from a LEFT OUT JOIN b ON a.a = b.c`

7、说明：在线视图查询(表名 1: a)

`select * from (SELECT a,b,c FROM a) T where t.a > 1;`

8、说明：between 的用法, between 限制查询数据范围时包括了边界值, not between 不包括

`select * from table1 where time between time1 and time2`

`select a,b,c, from table1 where a not between 数值 1 and 数值 2`

9、说明：in 的使用方法

`select * from table1 where a [not] in ('值 1', '值 2', '值 4', '值 6')`

10、说明：两张关联表，删除主表中已经在副表中没有的信息

`delete from table1 where not exists (select * from table2 where table1.field1=table2.field1)`

11、说明：四表联查问题：

`select * from a left inner join b on a.a=b.b right inner join c on a.a=c.c inner join d on a.a=d.d where`

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/736210232240010123>