常用SQL 语句

### 表别名的使用

Select \* from tablename1 a,tablename b where a.column1=b.column1

### 查看表信息（字段等）

**Sybase:**

Sp\_help tablename

查看SQL语句执行计划

**MySQL：**

explain select surname,first\_name form a,b where a.id=b.id

**Sybase:**

Set showplan on

查看系统表

Select id from sysobjects where name=”tablename”

Select name,length,scale,usertype from syscolumns where id= idnum 选取字段名称、长度、精度、数据类型

Select name from systypes where usertype= 选取字段类型

Select \* from usertype where usertype=

## 增删改查系列

### 选取前N行

**Sybase&&sql server:**

Select top 5 \* from tablename

**oracle:**

select \* from tab where rownum <= 10;

**mysql:**

select \* from tab limit 10

**db2:**

select \* from tab fetch first 10 rows only

### 批量插入

|  |
| --- |
| INSERT mytable (first\_column,second\_column,third\_column)  VALUES ('some data','some more data','yet more data') ,  VALUES ('some data','some more data','yet more data') ,  VALUES ('some data','some more data','yet more data') |

### 插入执行无反应

执行插入操作之后，许久没有反应，通常有可能是多人同时操作，造成表被锁。

使用如下语句查看哪些进程造成了表被锁

|  |
| --- |
| select sess.sid,  sess.serial#,  lo.oracle\_username,  lo.os\_user\_name,  ao.object\_name,  lo.locked\_mode  from v$locked\_object lo,  dba\_objects ao,  v$session sess  where ao.object\_id = lo.object\_id and lo.session\_id = sess.sid |

通过以上sql就可以知道哪个进程、序列，oracle用户名、操作系统用户名、表名、锁表模式几个字段

下面一步就是将改锁表的进程和序列杀掉了，执行下面的语句即可。

alter system kill session '1020,38953' --(1020，就是执行第一步语句得到的sid字段值，38953就是serial#字段值)

### 清空数据表

|  |
| --- |
| TRUNCATE TABLE `mytable` |

注意：删除表中的所有记录，应使用TRUNCATE TABLE语句。注意这里为什么要用TRUNCATE TABLE语句代替DELETE语句:当你使用TRUNCATE TABLE语句时，记录的删除是**不作记录**的。也就是说，这意味着TRUNCATE TABLE要比DELETE快得多。

Select 创建表

INSERT语句与DELETE语句和UPDATE语句有一点不同，它一次只操作一个记录。然而，有一个方法可以使INSERT 语句一次添加多个记录。要作到这一点，你需要把INSERT语句与SELECT语句结合起来，象这样:

|  |
| --- |
| INSERT mytable(first\_column,second\_column)  SELECT another\_first,another\_second FROM anothertable WHERE another\_first='Copy Me!'; |

这个语句从anothertable拷贝记录到mytable.只有表anothertable中字段another\_first的值为'Copy Me!'的记录才被拷贝。

　　当为一个表中的记录建立备份时，这种形式的INSERT语句是非常有用的。在删除一个表中的记录之前，你可以先用这种方法把它们拷贝到另一个表中。

如果你需要拷贝整个表，你可以使用SELECT INTO语句。例如，下面的语句创建了一个名为newtable的新表，该表包含表mytable的所有数据:

|  |
| --- |
| SELECT \* INTO newtable FROM mytable; |

你也可以指定只有特定的字段被用来创建这个新表。要做到这一点，只需在字段列表中指定你想要拷贝的字段。另外，你可以使用WHERE子句来限制拷贝到新表中的记录。下面的例子只拷贝字段second\_columnd的值等于'Copy Me!'的记录的first\_column字段。

|  |
| --- |
| SELECT first\_column INTO newtable  FROM mytable  WHERE second\_column='Copy Me!'; |

使用SQL修改已经建立的表是很困难的。例如，如果你向一个表中添加了一个字段，没有容易的办法来去除它。另外，如果你不小心把一个字段的数据类型给错了，你将没有办法改变它。但是，使用本节中讲述的SQL语句，你可以绕过这两个问题。

　　例如，假设你想从一个表中删除一个字段。使用SELECT INTO语句，你可以创建该表的一个拷贝，但不包含要删除的字段。这使你既删除了该字段，又保留了不想删除的数据。

如果你想改变一个字段的数据类型，你可以创建一个包含正确数据类型字段的新表。创建好该表后，你就可以结合使用UPDATE语句和SELECT语句，把原来表中的所有数据拷贝到新表中。通过这种方法，你既可以修改表的结构，又能保存原有的数据。

Select 1

就是如果表里如果有记录，就显示1,简单理解就是不查询具体列，只要有值就显示1。表有多少条记录,结果就是多少条"1"的行；一般用来判断表中是否有记录

一般用来当做判断子查询是否成功（即是否有满足条件的时候使用）比如

select \* from ta where exists (select 1 from ta.id = tb.id)

这个判断就是(select 1 from ta.id = tb.id)这个查询如果有返回值的话表示当前查询满足条件，一般来说就简单话的用select 1 当然也可以用select \* ,或者select 任何东东，因为这里仅仅是表明子查询有结果就行了，至于什么结果无所谓。