# 概述

关系数据库通常包含多个表。数据库实际是表的集合，数据库的数据或信息都是存储在表中的。表是对数据进行存储和操作的一种逻辑结构。对用户而言，一个表表示一个数据库对象。

数据库中的表与人们日常生活中使用的表格类似，也是由行（Row）和列（Column）组成。列由同类的信息组成，每列又称为一个字段，每列的标题称为字段名。行包括了若干列信息项，一行数据称为一条记录，表达有一定意义的信息组合。一个表由一条或多条记录组成，没有记录的表称为空表。每个表中通常都有一个主关键字，用于唯一地确定一条记录。经常见到的成绩表就是一种表，它由行和列组成，并且可以通过名字来识别数据，列包含了列的名字、数据类型以及列的其他属性；行包含了列的记录或者数据。

# 分类

在SQL中，并不是所有的表都是相同的。有些表是永久的，有些表则是临时的。有些表是模式对象，而有些表则包含在模块中，所有的模块表也是临时的。

## 永久表

永久表保存存储在数据库中的SQL数据。它是一种最常见的表，如果没有特别说明，通常所说的表就是指永久表。只要表的定义存在，永久表就始终存在。它的创建语句为CREAT TABLE。

## 全局临时表

这种表只有在SQL会话的上下文引用该表的定义时实际的表才会存在，对话结束后表就不再存在，不能从一个SQL会话访问在另一个会话中创建的表。全局临时表的创建语句为CREAT GLOBALTEMPORARY TABLE。

## 局部临时表

和全局临时表一样，局部临时表只有在SQL会话的过程中才能被引用，并且不能从另一个SQL会话对其进行访问。

而与全局临时表不同之处在于：我们在SQL会话内的任何地方都可以访问全局临时表；而局部临时表只有在相关的SQL模块内才能被访问。局部临时表的创建语句为CREAT LOCALTEMPORARY TABLE。

当操作非常大的表时，你可能偶尔需要运行很多查询获得一个大量数据小的子集，不是对整个表运行这些查询，而是让MySQL每次找出所需的少数记录，将记录选择到一个临时表可能更快些，然后再这些表上运行查询。

创建临时表很容易，给正常的CREATE TABLE语句加上TEMPORARY关键字即可。

临时表将在连接MySQL期间存在，当你断开的时候，MySQL将自动删除表并释放所有空间，当然你可以在仍然连接的时候删除表并释放空间。

如果在你创建名为tmp\_table临时表时名为tmp\_table的表在数据库中已经存在，临时表将有必要屏蔽（隐藏）非临时表tmp\_table。

如果声明临时表是一个HEAP表，MySQL允许你指定在内存中创建它：

CREATE TEMPORARY TABLE tmp\_table(

)TYPE=HEAP;

因为HEAP表存储在内存中，对它运行的查询比磁盘上的临时表快些。然而，HEAP表与一般的表有些不同，且有自身的限制。

# 派生表

当主查询中包含派生表，或者当select语句中包含union子句，或者select语句语句中包含一个字段的order by子句（对另一个字段的group by子句）时，MySQL为了完成查询，则需要自动创建临时表存储临时结果集，这种临时表是由MySQL自行创建，自行维护。

对于自动创建的临时表，由于内存临时表的性能更加优越，MySQL总是首先使用内存临时表，而当内存临时表变得太大时，达到某个预知的时候，内存临时表就转存为外存临时表。也就是说，外存临时表是内存临时表在存储空间上的一种延伸。内存临时表转存为外存临时表的阈值由系统变量max\_heap\_table\_size和tmp-table\_size的较小值决定。

派生表是从select语句返回的虚拟表，派生表类似于临时表，但是在select语句中使用派生表比临时表简单的多，因此它不需要创建临时表的步骤，所以当select语句的from子句中使用独立子查询时，我们将其称为派生表。

派生表一般在from子句中使用，如：

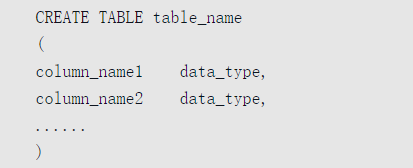
select \* from (select \* from table) table1;

# 操作

## 创建

常用的创建数据库表的方法有两种：一种是通过数据库管理系统（DBMS）提供的交互式创建工具创建，另一种是通过SQL直接创建。

在SQL中，创建数据库表的基本关键字为Create Table，在其后要指明创建的数据库表的名称，接着要分别定义表中各列的名称、数据类型等。语法如下：



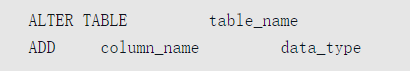
表的名字对大小写不敏感。表名要紧接在Create Table关键词的后面，且第一个字符必须是A～Z之一，其余的字符可以是字母，也可以是“\_”、“#”、“$”和“@”等符号。表中各列的定义在括号中完成，且各列之间以逗号隔开。不同的表，其列名可以相同，但是在同一个表中，不允许出现相同的列名。在定义了列名后，我们一定要指明该列的数据类型。

## 修改

在实际设计和创建数据库表的时候，我们很难做到一步到位，往往需要在使用的过程中，不断地修改完善。在SQL中，我们可以采用ALTER TABLE命令来修改已经创建的表结构。使用ALTER TABLE命令可以向表中增加新列、删除已有的列、也可以修改已经创建的列。

注意：对表定义的修改，不同的数据库系统有不同的限制。例如，Oracle数据库就限制对列的修改只能是加大列的宽度而不能是缩小，而且不能删除列。

### 增加新列



table\_name指的是要修改的表的名字，ADD关键字后面接要创建列的列名、数据类型等，当然也可以对列设置非空约束和缺省值。

当用ALTER TABLE语句向表中添加新列时，DBMS向表的列定义的尾部添加列，即在查询中将位于表的最右边。除非指定默认值，DBMS为已有行上的新列设NULL值。

由于DBMS为已有行上的新列设NULL值，当使用ALTER TABLE语句向表中添加新列时，我们不能简单地添加NOT NULL约束，还必须提供缺省值。因为如果没有提供缺省值，DBMS假设已有行上的新列为NULL值，这就和NOT NULL约束相抵触。当然，如果表中不存在数据，则不存在这个问题。

### 删除列

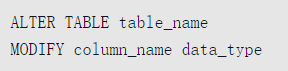
同样，在使用数据库表的过程中，如果其某列信息已经无效或不再需要，为了节省数据库空间，提高查询性能，我们可以采用DROP COLUMN关键字删除表中的某列，语法如下：



table\_name是要修改的表的名字，DROP COLUMN关键字后面接要删除列的名字。当然，一次可以删除多个列，只需要在DROP COLUMN关键字后面依次列出要删除的列的名字，中间用逗号分开即可。

### 修改列

如果发现数据库表中某列的结构不能满足实际的需求，在不破坏数据的情况下，SQL允许利用MODIFY关键字修改表中某列的结构。常用的修改操作主要包括字符长度限制的修定和非空约束的限制或取消，语法如下：



table\_name是要修改的表的名字，MODIFY关键字后面接要修改列的列名和修改后的数据条件。

**注意：**

在SQL Server数据库系统中，并不支持MODIFY关键字。要修改数据库中的列，我们可以通过ALTER COLUMN关键字实现，即将MODIFY替换为ALTER COLUMN即可。

用户可以通过MODIFY关键字或者ALTER COLUMN关键字增加或减少表中某列的最多字符数，但是，当要减少表中某列的最多字符数要特别慎重。当数据库表中该列存在已有记录的字符数多于减少后的最多字符限制时，表的修改就会失败。

当然，我们也可以通过MODIFY或者ALTER COLUMN关键字增加或取消表中某列的非空约束。

## 删除

表的删除非常容易，使用DROP TABLE关键词即可实现。语法如下。



只要在DROP TABLE关键词后面接上要删除表的名字即可。这里表的删除不仅删除了表内存储的数值，而是整个表结构都被删除了，也就是该表不存在了。

## 重命名

在创建表的时候，表的名字就被确定了，但在实际应用中，有时候需要修改表的名字而不改变其他信息，这时候就可以采用SQL的重命名表命令。

不同的DBMS对表的重命名提供的命令不尽相同。在DB2、MySQL,、Oracle数据库系统中可采用RENAME关键词，而在SQL Server和Sybase数据库系统中可采用SP\_RENAME关键词重命名表。

除了可以对表进行重命名以外，我们还可以对表中的列进行重命名。在SQL Server数据库系统中，重命名表中的列同样使用SP\_RENAME关键词。