# 表

## 复制表

## 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;

## 临时表

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。

# 分表

# 分片

分片解决的是拓展性问题，属于水平拆分，引入分片，就引入了数据路由和分区键的概念。

分表解决的是数据量多大的问题，分库解决的是数据库性能瓶颈问题。

# 分库

单纯的分表可以解决数据量过大导致索引变慢的问题，但是无法解决多并发请求访问同一个库导致数据库响应变慢的问题，所以通常水平拆分都至少要采用分库的方式（垂直拆分采用分表的方式），用于一并解决大数据量和高并发的问题，这也是部分开源的分片疏忽句酷中间件支支持分库的原因。