1. 连接
2. 概述

本质上说，连接就是将各个表中的记录都取出来进行依次匹配，并将匹配后的组合发送给客户端，这就是连接查询。如果连接查询的结果集中包含一个表中的每一条记录与另一个表中的每一条记录相互匹配的组合，那么这样的结果集就称为笛卡尔积。MySQL连接查询的语法就是在FROM语句后面跟多个表名即可。

SELECT \* FROM t1，t2;

1. 连接过程

在连接时需要过滤掉特定的记录组合，过滤条件分为两种：涉及单表的条件和涉及两表的条件。

1. 涉及单表的条件

这就是单表查询中的搜索条件，如t1.m1 > 1只针对t1表。

1. 涉及两表的条件

在两个表之间进行比较，如t1.m1=t2.m2。首先呢，确定第一个需要查询的表，这个表称为驱动表，选择代价最小的访问方法执行单表查询语句。从驱动表中获取到每一条记录，都需要到被驱动表中查询匹配的记录，同样是单表查询，不过是多次的查询，所谓匹配的记录指的是符合过滤条件的记录。

1. 内连接和外连接
2. 内连接

若驱动表中的记录在被驱动表中找不到匹配的记录，则该记录不会加入到最后的结果集中。

1. 外连接

针对驱动表中的某条记录，即使在被驱动表中没有找到与之匹配的记录，也仍然需要把该驱动表记录加入到结果集中。根据选取的驱动表的不同，外连接分为左外连接（选取左侧的表为驱动表）和右外连接（选取右侧的表为驱动表）。

1. WHERE子句和ON子句

WHERE子句适用于外连接和内连接，只要不符合的记录都不会加入到最后的结果集中。而对于ON子句，专用于外连接中，如果找不到能够匹配ON子句过滤条件的记录，那么该驱动表记录仍然会被加入到结果集中，各个字段用NULL值填充。ON子句在内连接中与WHERE子句是等价的。

1. 语法问题
2. 左连接

SELECT \* FROM t1 LEFT [OUTER] JOIN t2 ON 连接条件 [WHERE 普通过滤条件];

OUTER可省略，ON子句是必须的，WHERE子句可省略。

1. 右连接

将LEFT 换成了 RIGHT即可。

1. 内连接

SELECT \* FROM t1 [INNER | CROSS] JOIN t2 [ON 连接条件] [WHERE 普通过滤条件];

ON 子句和 WHERE子句都是可以省略的，INNER 和 CROSS 字段也是可以省略的。

1. 注意事项

内连接中，驱动表和被驱动表是可以互换的，不会影响最后的查询结果。而对于外连接来说，是不能互换的，因为驱动表的记录即使在被驱动表中找不到符合ON子句连接条件的记录，也会被加入到结果集中。

1. 连接的原理

最基本的连接就是通过嵌套循环一层一层地进行单表查询，选取代价最低的单表访问方法，从前一个驱动表中的查询结果集中通过ON或者WHERE子句来匹配被驱动表中的记录，最简单也是最笨拙的连接查询算法。

我们可以通过索引来加快查询速度，因为每一次查询其实还是单表查询，可以使用索引来优化速度。对于被驱动表，需要被访问多次，如果被驱动表的数据很多且无法使用索引查询，需要从磁盘上I/O多次，代价很大。MySQL中使用名为Join Buffer（连接缓冲区），即在连接查询前申请一块固定大小的内存，先将若干条驱动表结果集中的记录装在Buffer中，然后开始扫描被驱动表，每一条被驱动表的记录一次性地与Buffer中的多条驱动表记录进行匹配，过程都是在内存中完成，显著减少I/O代价。最好Buffer能够容纳驱动表结果集中所欲记录，那么只需要访问一次被驱动表即可完成连接操作。这种连接算法称为基于块的嵌套循环连接算法。

Join Buffer的大小可以通过系统变量 join\_buffer\_size配置，默认是256KB，注意Buffer中不会存放驱动表记录的所有列，只有查询列表中的列和过滤条件中的列，因此最好不要使用\*作为查询列表，为了在Buffer中放置更多的记录。