Mysql

查询语句书写顺序：select – from- where- group by- having- order by-limit

查询语句执行顺序：from - where -group by - having - select - order by-limit

一、sql组成：  
[DDL](https://www.baidu.com/s?wd=DDL&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvmvnsnW03nHTLPyuBuWD0IAYqnWm3PW64rj0d0AP8IA3qPjfsn1bkrjKxmLKz0ZNzUjdCIZwsrBtEXh9GuA7EQhF9pywdQhPEUiqkIyN1IA-EUBtYPHbdrH6LPj6" \t "https://zhidao.baidu.com/question/_blank)：数据库模式定义语言，关键字：create  
[DML](https://www.baidu.com/s?wd=DML&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvmvnsnW03nHTLPyuBuWD0IAYqnWm3PW64rj0d0AP8IA3qPjfsn1bkrjKxmLKz0ZNzUjdCIZwsrBtEXh9GuA7EQhF9pywdQhPEUiqkIyN1IA-EUBtYPHbdrH6LPj6" \t "https://zhidao.baidu.com/question/_blank)：数据操纵语言，关键字：Insert、delete、update  
[DCL](https://www.baidu.com/s?wd=DCL&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvmvnsnW03nHTLPyuBuWD0IAYqnWm3PW64rj0d0AP8IA3qPjfsn1bkrjKxmLKz0ZNzUjdCIZwsrBtEXh9GuA7EQhF9pywdQhPEUiqkIyN1IA-EUBtYPHbdrH6LPj6" \t "https://zhidao.baidu.com/question/_blank)：数据库控制语言 ，关键字：grant、remove  
DQL：数据库查询语言，关键字：select

1. **[union和union all的区别](http://www.cnblogs.com/xiangshu/articles/2054447.html)**

**Union因为要进行重复值扫描，所以效率低。如果合并没有刻意要删除重复行，那么就使用Union All**

相同个数的列，并且每个列的类型是一样的。但列名则不一定需要相同

union 和 union all都可以将多个结果集合并  
Union：对两个结果集进行并集操作，不包括重复行，同时进行默认规则的排序；   
Union All：对两个结果集进行并集操作，包括重复行，不进行排序；   
Intersect：对两个结果集进行交集操作，不包括重复行，同时进行默认规则的排序；   
Minus：对两个结果集进行差操作，不包括重复行，同时进行默认规则的排序。   
可以在最后一个结果集中指定Order by子句改变排序方式。

三、where与having

where针对表中的列发挥作用，查询数据  
having是对查询结果中得到的列发挥作用，筛选数据(也就是说从查询出的结果集再次进行筛选

四、内连接,外连接

内连接

连接的数据表相对应的匹配字段完全相等的连接。连接关键字是 inner join

外连接

分为左外连接与右外连接、全连接。

左连接的结果集包括指定的左表全部数据与匹配的右表数据,右表中没匹配的全为空值.关键字 left join

右连接的结果集包含指定的右表全部数据与匹配的左边数据,左边中没匹配的全为空值.关键字 right join

全连接返回左右数据表的所有行.关键字 full join

自然连接，不需要人为指定连接字段，自然会自动找同名字段进行连接，会删除连接后的重复列。 关键字 natural

交叉连接  生成笛卡尔积－它不使用任何匹配或者选取条件，而是直接将一个数据源中的每个行与另一个数据源的每个

效率问题:  
1.inner join比left join快  
注:inner join  内连接等价于下面的sql: SELECT A.name, B.address FROM A, B WHERE A.id = B.A\_id  
所以一般要用一般的连接就可以了.  
2.连接字段建索引

1. SQL优化

<http://www.cnblogs.com/zhyunfe/p/6209074.html>

1. ACID与事务隔离级别

事务（*transaction*）是数据库管理系统的执行单位，可以是一个数据库操作（如Select操作）或者是一组操作序列。事务ACID属性，即原子性（*Atomicity*）、一致性(*Consistency*)、隔离性（*Isolation*）、持久性（*Durability*）。

原子性：保证事务中的所有操作全部执行或全部不执行。例如执行转账事务，要么转账成功，要么失败。成功，则金额从转出帐户转入到目的帐户，并且两个帐户金额将发生相应的变化；失败，则两个账户的金额都不变。不会出现转出帐户扣了钱，而目的帐户没有收到钱的情况。

一致性：保证数据库始终保持数据的一致性——事务操作之前是一致的，事务操作之后也是一致的，不管事务成功与否。如上面的例子，转账之前和之后数据库都保持数据上的一致性。

隔离性：多个事务并发执行的话，结果应该与多个事务串行执行效果是一样的。显然最简单的隔离就是将所有事务都串行执行：先来先执行，一个事务执行完了才允许执行下一个。但这样数据库的效率低下，如：两个不同的事务只是读取同一批数据，这样完全可以并发进行。为了控制并发执行的效果就有了不同的隔离级别。下面将详细介绍。

持久性：持久性表示事务操作完成之后，对数据库的影响是持久的，即使数据库因故障而受到破坏，数据库也应该能够恢复。通常的实现方式是采用日志。

事务隔离级别（*transaction isolation levels*）：隔离级别就是对对事务并发控制的等级。[ANSI](http://en.wikipedia.org/wiki/American_National_Standards_Institute" \o "American National Standards Institute)/ [ISO](http://en.wikipedia.org/wiki/International_Organization_for_Standardization" \o "International Organization for Standardization) [SQL](http://en.wikipedia.org/wiki/SQL" \o "SQL)将其分为串行化（SERIALIZABLE）、可重复读（REPEATABLE READ）、读已提交（READ COMMITED）、读未提交（READ UNCOMMITED）四个等级。为了实现隔离级别通常数据库采用锁（Lock）。一般在编程的时候只需要设置隔离等级，至于具体采用什么锁则由数据库来设置。首先介绍四种等级，然后举例解释后面三个等级（可重复读、读已提交、读未提交）中会出现的并发问题。

串行化（SERIALIZABLE）：所有事务都一个接一个地串行执行，这样可以避免幻读（phantom reads）。对于基于锁来实现并发控制的数据库来说，串行化要求在执行范围查询（如选取年龄在10到30之间的用户）的时候，需要获取范围锁（range lock）。如果不是基于锁实现并发控制的数据库，则检查到有违反串行操作的事务时，需要滚回该事务。

可重复读（REPEATABLE READ）：所有被Select获取的数据都不能被修改，这样就可以避免一个事务前后读取数据不一致的情况。但是却没有办法控制幻读，因为这个时候其他事务不能更改所选的数据，但是可以增加数据，因为前一个事务没有范围锁。Mysql默认的隔离等级

读已提交（READ COMMITED）：被读取的数据可以被其他事务修改。这样就可能导致不可重复读。也就是说，事务的读取数据的时候获取读锁，但是读完之后立即释放（不需要等到事务结束），而写锁则是事务提交之后才释放。释放读锁之后，就可能被其他事物修改数据。该等级也是SQL Server默认的隔离等级。

读未提交（READ UNCOMMITED）：这是最低的隔离等级，允许其他事务看到没有提交的数据。这种等级会导致脏读（Dirty Read）。

**1.** **脏读** ：脏读就是指当一个事务正在访问数据，并且对数据进行了修改，而这种修改还没有提交到数据库中，这时，另外一个事务也访问这个数据，然后使用了这个数据。

**2.** **不可重复读** ：是指在一个事务内，多次读同一数据。在这个事务还没有结束时，另外一个事务也访问该同一数据。那么，在第一个事务中的两 次读数据之间，由于第二个事务的修改，那么第一个事务两次读到的的数据可能是不一样的。这样就发生了在一个事务内两次读到的数据是不一样的，因此称为是不 可重复读。例如，一个编辑人员两次读取同一文档，但在两次读取之间，作者重写了该文档。当编辑人员第二次读取文档时，文档已更改。原始读取不可重复。如果 只有在作者全部完成编写后编辑人员才可以读取文档，则可以避免该问题。

**3.** **幻读** : 是指当事务不是独立执行时发生的一种现象，例如第一个事务对一个表中的数据进行了修改，这种修改涉及到表中的全部数据行。 同时，第二个事务也修改这个表中的数据，这种修改是向表中插入一行新数据。那么，以后就会发生操作第一个事务的用户发现表中还有没有修改的数据行，就好象 发生了幻觉一样。例如，一个编辑人员更改作者提交的文档，但当生产部门将其更改内容合并到该文档的主复本时，发现作者已将未编辑的新材料添加到该文档中。 如果在编辑人员和生产部门完成对原始文档的处理之前，任何人都不能将新材料添加到文档中，则可以避免该问题。



七、读写分离

八、加密

九、索引

MySQL数据库索引的4大类型以及相关的索引创建

1. 普通索引：最基本的MySQL数据库索引，它没有任何限制

创建方式

创建索引：CREATE INDEX indexName ON mytable(username(length));

修改表结构：ALTER mytable ADD INDEX [indexName] ON (username(length))

创建表时制定：

CREATE TABLE mytable( ID INT NOT NULL, username VARCHAR(16) NOT NULL, INDEX [indexName] (username(length)) );

如果是CHAR，VARCHAR类型，length可以小于字段实际长度；如果是BLOB和TEXT类型，必须指定 length

删除索引的语法：

DROP INDEX [indexName] ON mytable;

1. 唯一索引：MySQL数据库索引列的值必须唯一，但允许有空值。如果是组合索引，则列值的组合必须唯一

创建索引：CREATE UNIQUE INDEX indexName ON mytable(username(length))

修改表结构：ALTER mytable ADD UNIQUE [indexName] ON (username(length))

创建表的时候直接指定：

CREATE TABLE mytable(ID INT NOT NULL, username VARCHAR(16) NOT NULL, UNIQUE [indexName] (username(length)) );

1. 主键索引：一种特殊的唯一索引，不允许有空值。一个表只能有一个主键。

一般是在建表的时候同时创建主键索引：

CREATE TABLE mytable( ID INT NOT NULL, username VARCHAR(16) NOT NULL,PRIMARY KEY(ID) );

也可以用 ALTER 命令

1. 组合索引

ALTER TABLE mytable ADD INDEX name\_city\_age (name(10),city,age);

最左前缀：顾名思义，就是最左优先，上例中我们创建了lname\_fname\_age多列索引,相当于创建了(lname)单列索引，(lname,fname)组合索引以及(lname,fname,age)组合索引。

索引长度和区分度：索引长度太短，那么区分度就很低，吧索引长度加长，区分度就高，但是索引也是要占内存的，所以我们需要找到一个平衡点；

http://www.cnblogs.com/codeAB/p/6391303.html