1. 一张表，里面有 ID 自增主键，当 insert 了 17 条记录之后，删除了第 15,16,17 条记录，再把 Mysql 重启，再 insert 一条记录，这条记录的 ID 是 18 还是 15 ？

如果存储引擎类型是MyIsam，那么结果是18，因为MyIsam会把自增主键id的最大值存在文件中，服务重启也不会丢失。如果是InnoDB，那么结果是15，因为InnoDB是把自增主键的最大值记录到内存中，所以重启服务会丢失。

1. Heap 表是什么？

它是基于memory存储引擎创建的表，数据存在于内存中，用于临时高速存储。不支持自增主键和事务，索引不能为NULl,支持b-tree索引和hash索引。

1. FLOAT 和 DOUBLE、decimal的区别

Float：浮点型，含字节数为4，数值只有7个有效位，如果不指定精度，默认会按照实际精度存储；

Double：双精度类型，含字节数为8，数值有15个有效位，如果不指定精度，默认会按照实际精度存储；

Decimal：数字类型，不存在精度损失，28个有效位,如果不指定精度，则默认整数位为10，小数位为0。如果超过了指定的长度和精度，会提示警告。

超过精度的都会四舍五入。

1. char与varchar的区别：

char：长度固定，即char(M)类型的数据列里，每个值都占用M个字节，如果某个长度小于M，MySQL就会在它的右边用空格字符补足。

varchar:长度是可变的，即在varchar(M)类型的数据列里，每个值只占用刚好够用的字节再加上一个用来记录其长度的字节（即总长度为L+1字节）。

声明为char的列长度是固定的,char的长度可选范围在0-255之间.也就是char最大能存储255个字符.如果该列是utf8编码,则该列所占用的字节数=字符数\*3.如果是gbk编码则该列所占用的字节数=字符数\*2。

声明为varchar的列长度是可变的,在mysql5.0.3之前varchar的长度范围为0-255,mysql5.0.3之后varchar的长度范围为0-65535个字节.采用varchar类型存储数据需要1-2个字节(长度超过255时需要2个字节)来存储字符串的实际长度.如果该列的编码为gbk,每个字符最多占用2个字节,最大长度不能超过32766个字符.如果该列的编码为utf8,每个字符最多占3个字节,最大字符长度为21845.

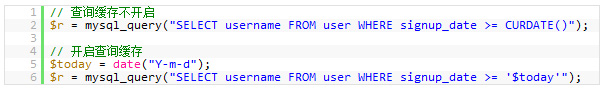
1. mysql字符串类型的Enum、blob、set

blob是二进制类型，可以容纳可变数量的数据，最大65k；

Enum与set都是枚举类型，enum只能存一个值，set可以存多个

1. 性能优化：

①如果开启了查询缓存，sql中如果有内置函数作为条件，那么改为用变量替换掉查询语句中的内置函数。如：



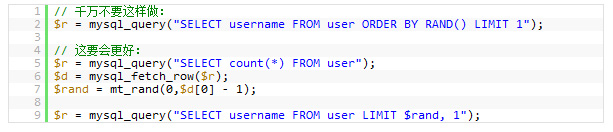
②使用explain解析查询语句，然后分析新能瓶颈

③当只要查询一行数据时，再查询语句后面加上limit 1;

④为搜索字段加索引；

⑤，如果时join查询，为join字段加索引；

⑥千万不要用order by rand().rand()是内置函数，执行的时候非常耗性能。就算加上limit 1也无济于事。



⑦避免使用select (\*) 和select count(\*)

⑧能用Enum就尽量不要用varchar，因为数据库实际上保存的tinyint类型。

⑨字段尽量不要设置成默认为Null，因为null需要额外的空间，并且比较的时候更复杂。

⑩从 PROCEDURE ANALYSE() 取得建议。

1. 唯一索引比普通索引快吗？为什么？

不一定，还可能慢。

1. 查询时，在未使用limit 1的情况下，唯一索引查询到一条数据后立马返回，普通索引会继续匹配下一条数据，发现没了匹配数据之后菜返回，虽然普通索引多花了时间但是这个时间差可能可以忽略不计，如果使用了limit 1，则都是立马返回，查询时间差不多。
2. 新增数据时，唯一索引还需要额外校验字段的唯一性,需要花费更长的时间.
3. MySQL怎么恢复半个月前的数据？

通过整库备份+binlog日志进行恢复，前提是要有定期整库备份，且保存了binlog日志。

1. MyISAM和InnoDB\*\*的区别有哪些？

InnoDB支持事务, MyISAM不支持.

InnoDB支持行级锁, MyISAM支持表级锁.

InnoDB支持多版本并发控制(MVVC), MyISAM不支持.

InnoDB支持外键, MyISAM不支持.

MyISAM支持全文索引, InnoDB部分版本不支持(但可以使用Sphinx插件)

1. mysql架构或者说是由那几个部分组成？

1. Server

连接器: 管理连接, 权限验证.

分析器: 词法分析, 语法分析.

优化器: 执行计划生成, 索引的选择.

执行器: 操作存储引擎, 返回执行结果.

1. 存储引擎: 存储数据, 提供读写接口.

11、MySQL查询缓存有什么弊端, 应该什么情况下使用, 8.0版本对查询缓存有什么变更.

查询缓存可能会失效非常频繁, 对于一个表, 只要有更新, 该表的全部查询缓存都会被清空. 因此对于频繁更新的表来说, 查询缓存不一定能起到正面效果。对于读远多于写的表可以考虑使用查询缓存.8.0版本的查询缓存功能被删了。

1. 一千万条数据的表如何分页？

数据量过大的情况下，limit offse会由于扫描数据太多导致越往后查询越慢，可以适当配合最后当前页最后一条主键id进行判断，即where条件后加上主键id >或<,比如：SELECT \* FROM T WHERE id > #{ID} LIMIT #{LIMIT} . 当然, 这种情况下ID必须是有序的, 这也是有序ID的好处之一。

1. 订单表数据越来越大导致查询慢，如何处理？

分库分表，由于历史订单的使用率不高，因此可以将订单表按时间进行拆分，根据数据量的大小考虑按月分表或按年分表. 订单ID最好包含时间(如根据雪花算法生成), 此时既能根据订单ID直接获取到订单记录, 也能按照时间进行查询。