



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

第三版：MySQL 45 道

为什么要使用视图？什么是视图？

为什么要使用视图？

为了提高复杂 SQL 语句的复用性和表操作的安全性，MySQL 数据库管理系统提供了视图特性。

什么是视图？

视图是一个虚拟的表，是一个表中的数据经过某种筛选后的显示方式，视图由一个预定义的查询 select 语句组成。

视图有哪些特点？哪些使用场景？

视图特点：

- 1、视图的列可以来自不同的表，是表的抽象和在逻辑意义上建立的新关系。
- 2、视图是由基本表(实表)产生的表(虚表)。
- 3、视图的建立和删除不影响基本表。
- 4、对视图内容的更新(添加，删除和修改)直接影响基本表。
- 5、当视图来自多个基本表时，不允许添加和删除数据。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

视图用途： 简化 sql 查询，提高开发效率，兼容老的表结构。

视图的常见使用场景：

- 1、 重用 SQL 语句；
- 2、 简化复杂的 SQL 操作。
- 3、 使用表的组成部分而不是整个表；
- 4、 保护数据
- 5、 更改数据格式和表示。视图可返回与底层表的表示和格式不同的数据。

视图的优点，缺点，讲一下？

- 1、 查询简单化。视图能简化用户的操作
- 2、 数据安全性。视图使用户能以多种角度看待同一数据，能够对机密数据提供安全保护
- 3、 逻辑数据独立性。视图对重构数据库提供了一定程度的逻辑独立性

count(1)、count(*) 与 count(列名) 的区别？

- 1、 count(*)包括了所有的列，相当于行数，在统计结果的时候，不会忽略列值为 NULL

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

2、count(1)包括了忽略所有列，用1代表代码行，在统计结果的时候，不会忽略列值为NULL

3、count(列名)只包括列名那一列，在统计结果的时候，会忽略列值为空（这里的空不是只空字符串或者0，而是表示null）的计数，即某个字段值为NULL时，不统计。

什么是游标？

游标提供了一种对从表中检索出的数据进行操作的灵活手段，就本质而言，游标实际上是一种能从包括多条数据记录的结果集中每次提取一条记录的机制。

什么是存储过程？有哪些优缺点？

存储过程，就是一些编译好了的SQL语句，这些SQL语句代码像一个方法一样实现一些功能（对单表或多表的增删改查），然后给这些代码块取一个名字，在用这个功能的时候调用即可。

优点：

- 1、存储过程是一个预编译的代码块，执行效率比较高
- 2、存储过程在服务器端运行，减少客户端的压力
- 3、允许模块化程序设计，只需要创建一次过程，以后在程序中就可以调用该过程任意次，类似方法的复用
- 4、一个存储过程替代大量T_SQL语句，可以降低网络通信量，提高通信速率

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

5、 可以一定程度上确保数据安全

缺点：

- 1、 调试麻烦
- 2、 可移植性不灵活
- 3、 重新编译问题

什么是触发器？触发器的使用场景有哪些？

触发器，指一段代码，当触发某个事件时，自动执行这些代码。

使用场景：

- 1、 可以通过数据库中的相关表实现级联更改。
- 2、 实时监控某张表中的某个字段的更改而需要做出相应的处理。
- 3、 例如可以生成某些业务的编号。
- 4、 注意不要滥用，否则会造成数据库及应用程序的维护困难。

MySQL 中都有哪些触发器？

MySQL 数据库中有六种触发器：

- 1、 Before Insert

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

- 2、 After Insert
- 3、 Before Update
- 4、 After Update
- 5、 Before Delete
- 6、 After Delete

超键、候选键、主键、外键分别是什么？

- 1、 超键：在关系模式中，能唯一标识元组的属性集称为超键。
- 2、 候选键：是最小超键，即没有冗余元素的超键。
- 3、 主键：数据库表中对储存数据对象予以唯一和完整标识的数据列或属性的组合。一个数据列只能有一个主键，且主键的取值不能缺失，即不能为空值（Null）。
- 4、 外键：在一个表中存在的另一个表的主键称此表的外键。。

SQL 约束有哪几种呢？

- 1、 NOT NULL: 约束字段的内容一定不能为 NULL。
- 2、 UNIQUE: 约束字段唯一性，一个表允许有多个 Unique 约束。
- 3、 PRIMARY KEY: 约束字段唯一，不可重复，一个表只允许存在一个。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

4、 FOREIGN KEY: 用于预防破坏表之间连接的动作，也能防止非法数据插入外键。

5、 CHECK: 用于控制字段的值范围。

谈谈六种关联查询，使用场景。

1、 交叉连接

2、 内连接

3、 外连接

4、 联合查询

5、 全连接

6、 交叉连接

varchar(50)中 50 的涵义

1、 字段最多存放 50 个字符

2、 如 varchar(50) 和 varchar(200) 存储 "jay" 字符串所占空间是一样的，后者在排序时会消耗更多内存

MySQL 中 int(20)和 char(20)以及 varchar(20)的区别

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

- 1、 int(20) 表示字段是 int 类型，显示长度是 20
- 2、 char(20)表示字段是固定长度字符串，长度为 20
- 3、 varchar(20) 表示字段是可变长度字符串，长度为 20

drop、delete 与 truncate 的区别

	delete	truncate	drop
类型	DML	DDL	DDL
回滚	可回滚	不可回滚	不可回滚
删除内容	表结构还在，删除表的全部或者一部分数据行	表结构还在，删除表中的所有数据	从数据库中删除表，所有的数据行，索引和权限也会被删除
删除速度	删除速度慢，逐行删除	删除速度快	删除速度最快

UNION 与 UNION ALL 的区别?

- 1、 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；
- 2、 Union All：对两个结果集进行并集操作，包括重复行，不进行排序；
- 3、 UNION 的效率高于 UNION ALL

SQL 的生命周期?

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

- 1、 服务器与数据库建立连接
- 2、 数据库进程拿到请求 sql
- 3、 解析并生成执行计划，执行
- 4、 读取数据到内存，并进行逻辑处理
- 5、 通过步骤一的连接，发送结果到客户端
- 6、 关掉连接，释放资源

一条 Sql 的执行顺序？

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



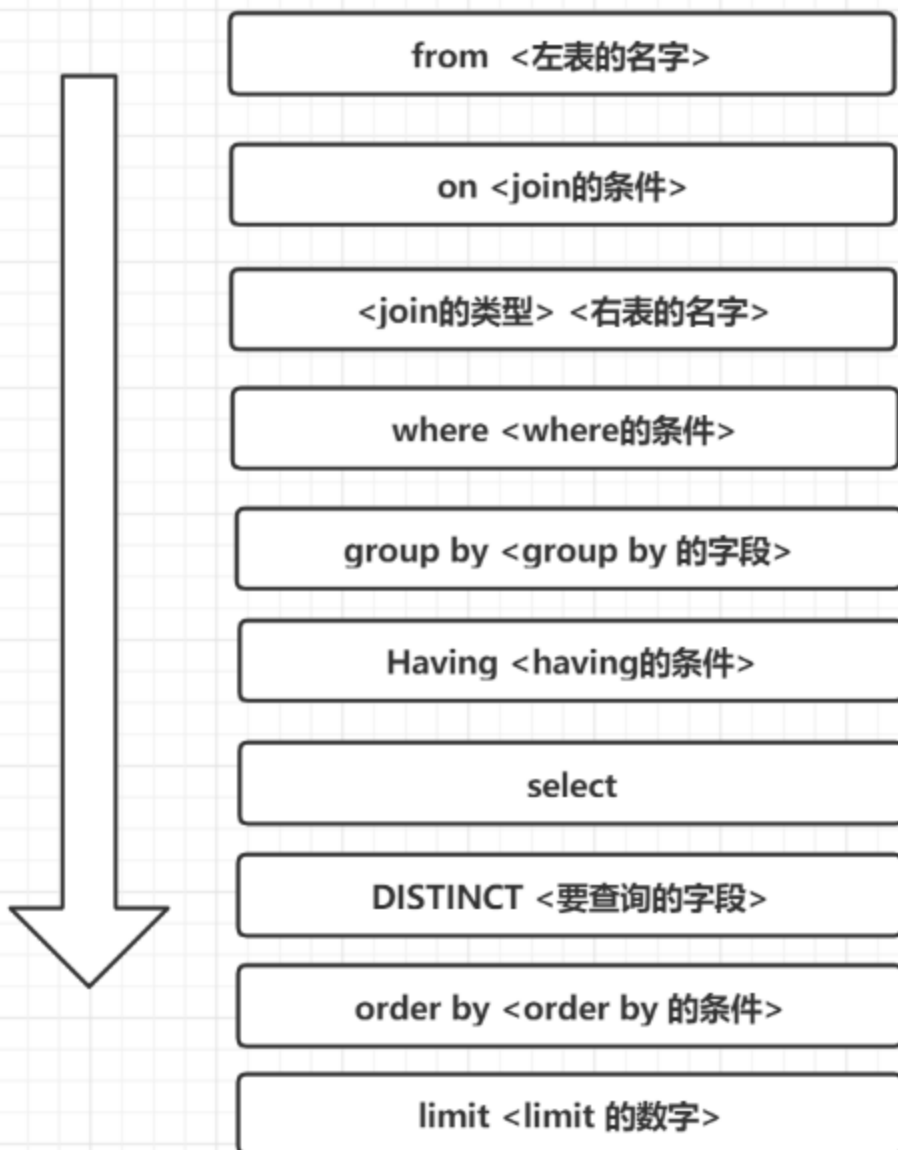
微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题



列值为 NULL 时，查询是否会用到索引？

列值为 NULL 也是可以走索引的

计划对列进行索引，应尽量避免把它设置为可空，因为这会让 MySQL 难以优化。引用了可空列的查询，同时增加了引擎的复杂度。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

关心过业务系统里面的 sql 耗时吗？统计过慢查询吗？对慢查询都怎么优化过？

- 1、我们平时写 Sql 时，都要养成用 explain 分析的习惯。
- 2、慢查询的统计，运维会定期统计给我们

优化慢查询：

- 1、分析语句，是否加载了不必要的字段/数据。
- 2、分析 SQL 执行语句，是否命中索引等。
- 3、如果 SQL 很复杂，优化 SQL 结构
- 4、如果表数据量太大，考虑分表

主键使用自增 ID 还是 UUID，为什么？

如果是单机的话，选择自增 ID；如果是分布式系统，优先考虑 UUID 吧，但还是最好自己公司有一套分布式唯一 ID 生产方案吧。

- 1、自增 ID：数据存储空间小，查询效率高。但是如果数据量过大，会超出自增长的值范围，多库合并，也有可能有问题。
- 2、uuid：适合大量数据的插入和更新操作，但是它无序的，插入数据效率慢，占用空间大。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

MySQL 自增主键用完了怎么办？

自增主键一般用 int 类型，一般达不到最大值，可以考虑提前分库分表的。

字段为什么要求定义为 not null？

null 值会占用更多的字节，并且 null 有很多坑的。

如果要存储用户的密码散列，应该使用什么字段进行存储？

密码散列，盐，用户身份证号等固定长度的字符串，应该使用 char 而不是 varchar 来存储，这样可以节省空间且提高检索效率。

MySQL 驱动程序是什么？

这个 jar 包：MySQL-connector-java-5.1.18.jar

MySQL 驱动程序主要帮助编程语言与 MySQL 服务端进行通信，如连接、传输数据、关闭等。

如何优化长难的查询语句？有实战过吗？

- 1、 将一个大的查询分为多个小的相同的查询
- 2、 减少冗余记录的查询。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

3、 一个复杂查询可以考虑拆成多个简单查询

4、 分解关联查询，让缓存的效率更高。

优化特定类型的查询语句

平时积累吧：

- 1、 比如使用 `select 具体字段` 代替 `select *`
- 2、 使用 `count(*)` 而不是 `count(列名)`
- 3、 在不影响业务的情况，使用缓存
- 4、 `explain` 分析你的 SQL

MySQL 数据库 cpu 飙升的话，要怎么处理呢？

排查过程：

- 1、 使用 `top` 命令观察，确定是 MySQLd 导致还是其他原因。
- 2、 如果是 MySQLd 导致的，`show processlist`，查看 `session` 情况，确定是不是有消耗资源的 sql 在运行。
- 3、 找出消耗高的 sql，看看执行计划是否准确，索引是否缺失，数据量是否太大。

处理：

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

1、 kill 掉这些线程(同时观察 cpu 使用率是否下降),

2、 进行相应的调整(比如说加索引, 改 sql、改内存参数)

3、 重新跑这些 SQL。

其他情况:

也有可能是每个 sql 消耗资源并不多, 但是突然之间, 有大量的 session 连进来导致 cpu 飙升, 这种情况就需要跟应用一起来分析为何连接数会激增, 再做出相应的调整, 比如说限制连接数等

读写分离常见方案?

1、 应用程序根据业务逻辑来判断, 增删改等写操作命令发给主库, 查询命令发给备库。

2、 利用中间件来做代理, 负责对数据库的请求识别出读还是写, 并分发到不同的数据库中。(如: amoeba, MySQL-proxy)

MySQL 的复制原理以及流程

主从复制原理, 简言之, 就三步曲, 如下:

1、 主数据库有个 bin-log 二进制文件, 纪录了所有增删改 Sql 语句。(binlog 线程)

2、 从数据库把主数据库的 bin-log 文件的 sql 语句复制过来。(io 线程)

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

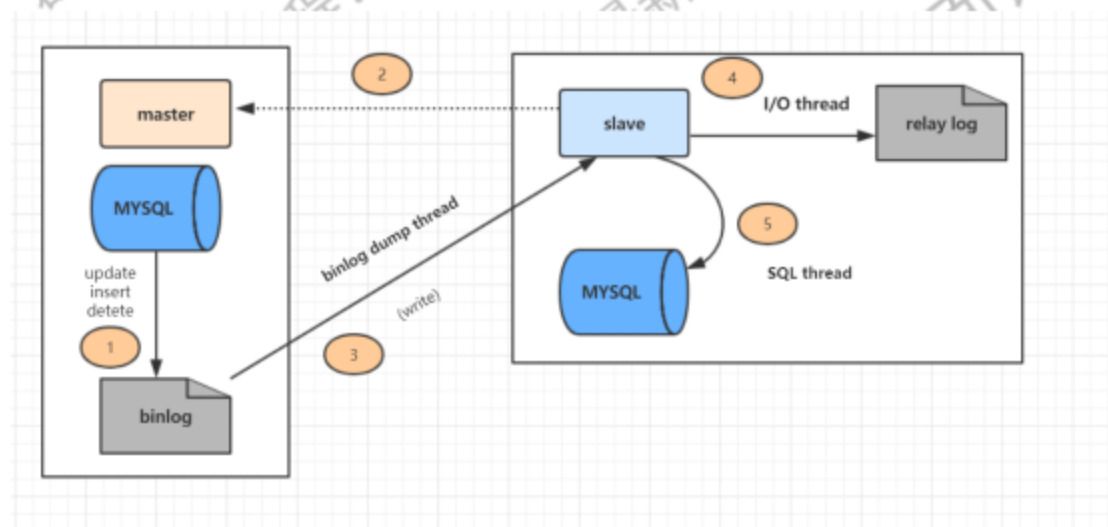
扫码关注



回复：面试题 获取最新版面试题

3、从数据库的 relay-log 重做日志文件中再执行一次这些 sql 语句。（Sql 执行线程）

如下图所示：



上图主从复制分了五个步骤进行：

- 1、主库的更新事件(update、insert、delete)被写到 binlog
- 2、从库发起连接，连接到主库。
- 3、此时主库创建一个 binlog dump thread，把 binlog 的内容发送到从库。
- 4、从库启动之后，创建一个 I/O 线程，读取主库传过来的 binlog 内容并写入到 relay log
- 5、还会创建一个 SQL 线程，从 relay log 里面读取内容，从 Exec_Master_Log_Pos 位置开始执行读取到的更新事件，将更新内容写入到 slave 的 db

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

MySQL 中 DATETIME 和 TIMESTAMP 的区别

存储精度都为秒

区别：

- 1、 DATETIME 的日期范围是 1001——9999 年；TIMESTAMP 的时间范围是 1970——2038 年
- 2、 DATETIME 存储时间与时区无关；TIMESTAMP 存储时间与时区有关，显示的值也依赖于时区
- 3、 DATETIME 的存储空间为 8 字节；TIMESTAMP 的存储空间为 4 字节
- 4、 DATETIME 的默认值为 null；TIMESTAMP 的字段默认不为空(not null)，默认值为当前时间(CURRENT_TIMESTAMP)

InnoDB 的事务实现原理？

- 1、 原子性：是使用 undo log 来实现的，如果事务执行过程中出错或者用户执行了 rollback，系统通过 undo log 日志返回事务开始的状态。
- 2、 持久性：使用 redo log 来实现，只要 redo log 日志持久化了，当系统崩溃，即可通过 redo log 把数据恢复。
- 3、 隔离性：通过锁以及 MVCC,使事务相互隔离开。
- 4、 一致性：通过回滚、恢复，以及并发情况下的隔离性，从而实现一致性。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

谈谈 MySQL 的 Explain

Explain 执行计划包含字段信息如下：分别是 id、select_type、table、partitions、type、possible_keys、key、key_len、ref、rows、filtered、Extra 等 12 个字段。

我们重点关注的是 type，它的属性排序如下：

```
system > const > eq_ref > ref > ref_or_null >
index_merge > unique_subquery > index_subquery >
range > index > ALL
```

InnoDB 的事务与日志的实现方式

有多少种日志

innodb 两种日志 redo 和 undo。

日志的存放形式

- 1、redo: 在页修改的时候, 先写到 redo log buffer 里面, 然后写到 redo log 的文件系统缓存里面(fwrite), 然后再同步到磁盘文件 (fsync)。
- 2、Undo: 在 MySQL5.5 之前, undo 只能存放在 ibdata 文件里面, 5.6 之后, 可以通过设置 innodb_undo_tablespaces 参数把 undo log 存放在 ibdata 之外。

事务是如何通过日志来实现的

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

1、因为事务在修改页时，要先记 undo，在记 undo 之前要记 undo 的 redo，然后修改数据页，再记数据页修改的 redo。Redo (里面包括 undo 的修改) 一定要比数据页先持久化到磁盘。

2、当事务需要回滚时，因为有 undo，可以把数据页回滚到前镜像的状态，崩溃恢复时，如果 redo log 中事务没有对应的 commit 记录，那么需要用 undo 把该事务的修改回滚到事务开始之前。

3、如果有 commit 记录，就用 redo 前滚到该事务完成时并提交掉。

MySQL 中 TEXT 数据类型的最大长度

1、TINYTEXT: 256 bytes

2、TEXT: 65,535 bytes(64kb)

3、MEDIUMTEXT: 16,777,215 bytes(16MB)

4、LONGTEXT: 4,294,967,295 bytes(4GB)

1、### 500 台 db，在最快时间之内重启。

2、可以使用批量 ssh 工具 pssh 来对需要重启的机器执行重启命令。

3、也可以使用 salt (前提是客户端有安装 salt) 或者 ansible (ansible 只需要 ssh 免登通了就行) 等多线程工具同时操作多台服务

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

你是如何监控你们的数据库的？你们的慢日志都是怎么查询的？

监控的工具有很多，例如 zabbix，lepus，我这里用的是 lepus

你是否做过主从一致性校验，如果有，怎么做的，如果没有，你打算怎么做？

主从一致性校验有多种工具 例如 checksum、MySQLdiff、pt-table-checksum 等

你们数据库是否支持 emoji 表情存储，如果不支持，如何操作？

更换字符集 utf8-->utf8mb4

MySQL 如何获取当前日期？

```
SELECT CURRENT_DATE();
```

一个 6 亿的表 a，一个 3 亿的表 b，通过外间 tid 关联，你如何最快的查询出满足条件的第 50000 到第 50200 中的这 200 条数据记录。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

1、如果 A 表 TID 是自增长,并且是连续的,B 表的 ID 为索引

```
select * from a,b where a.tid = b.id and a.tid>500000 limit 200;
```

2、如果 A 表的 TID 不是连续的,那么就需要使用覆盖索引.TID 要么是主键,要么是辅助索引,B 表 ID 也需要有索引。

```
select * from b, (select tid from a limit 50000,200) a where b.id = a .tid;
```

MySQL 一条 SQL 加锁分析

一条 SQL 加锁, 可以分 9 种情况进行:

- 1、组合一: id 列是主键, RC 隔离级别
- 2、组合二: id 列是二级唯一索引, RC 隔离级别
- 3、组合三: id 列是二级非唯一索引, RC 隔离级别
- 4、组合四: id 列上没有索引, RC 隔离级别
- 5、组合五: id 列是主键, RR 隔离级别
- 6、组合六: id 列是二级唯一索引, RR 隔离级别
- 7、组合七: id 列是二级非唯一索引, RR 隔离级别
- 8、组合八: id 列上没有索引, RR 隔离级别

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

9、 组合九：Serializable 隔离级别

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题