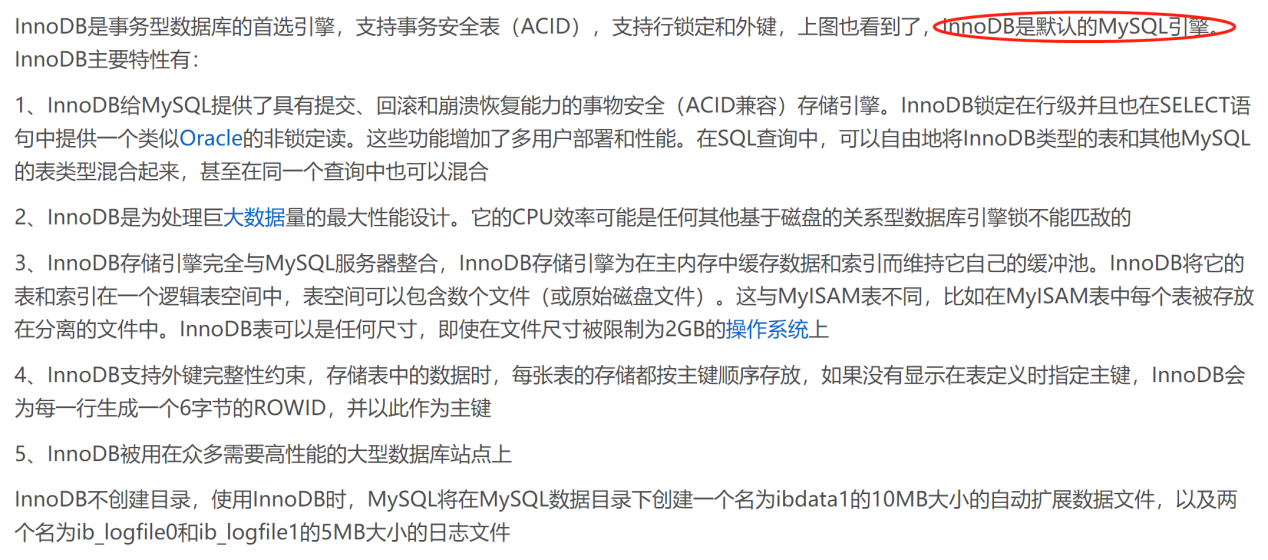
# MySql的存储引擎

## InnoDB存储引擎



## MyISAM存储引擎

## MEMORY存储引擎

## Archive存储引擎

# Sql语句执行顺序

From

Join On

Where

Group by

With

Having

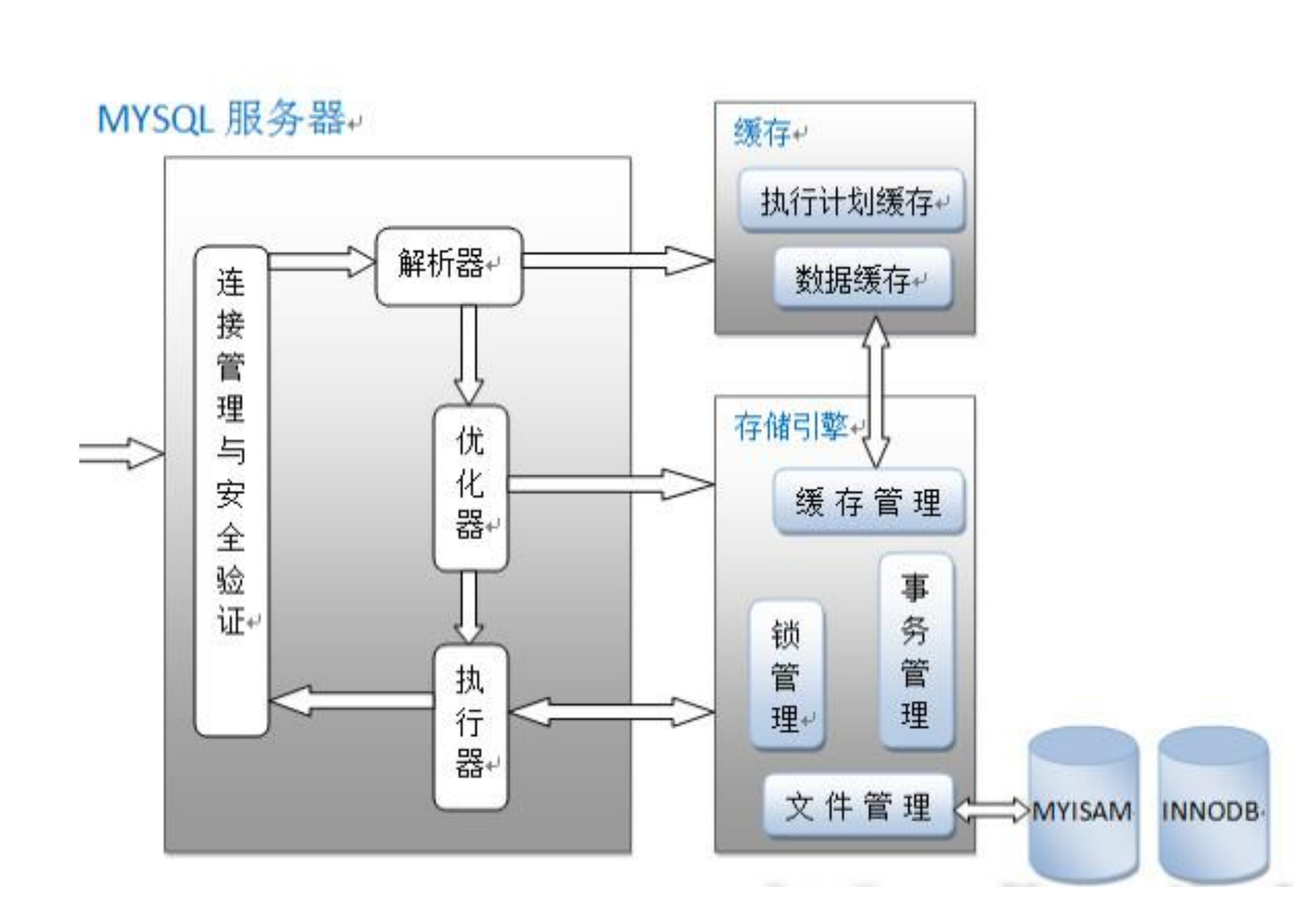
Select

Distinct

Order by

以上每一个步骤都会生成一个虚拟表，该虚拟表用作下一个步骤的输入，也就是下一个步骤在上一个步骤的基础上进行筛选。中间过程生成的虚拟表对用户是不可见的，只有最后一步生成的虚拟表才会给调用表。

# 架构



## 连接管理与安全验证

每个客户都会建立一个与服务器连接的线程，服务器会有一个连接池来管理这些链接，如果客户端需要建立连接，还需要验证用户名、密码、主机信息等。

## 解析器

分析查询语句，最终生成解析树。首先解析器会对查询语句做语法分析，分析语法是否合法。解析器还会查询缓存，如果缓存中有对应的语句，就返回查询结果不再进行接下来的优化操作，前提是缓存中的数据没被修改。

## 优化器

主要对查询语句进行优化，包括选择合适的索引，数据的读取方式，包括获取查询的开销信息，统计信息等。优化器指向存储引擎。

## 执行器

执行查询语句，返回查询结果，生成执行计划包括一些与存储引擎的一些处理操作。

# 事务的四大特征

原子性、一致性、隔离性、持久性

# 四种隔离级别

读未提交

读已提交 sqlserver oracle的默认隔离级别

可重复读 mysql默认的隔离级别

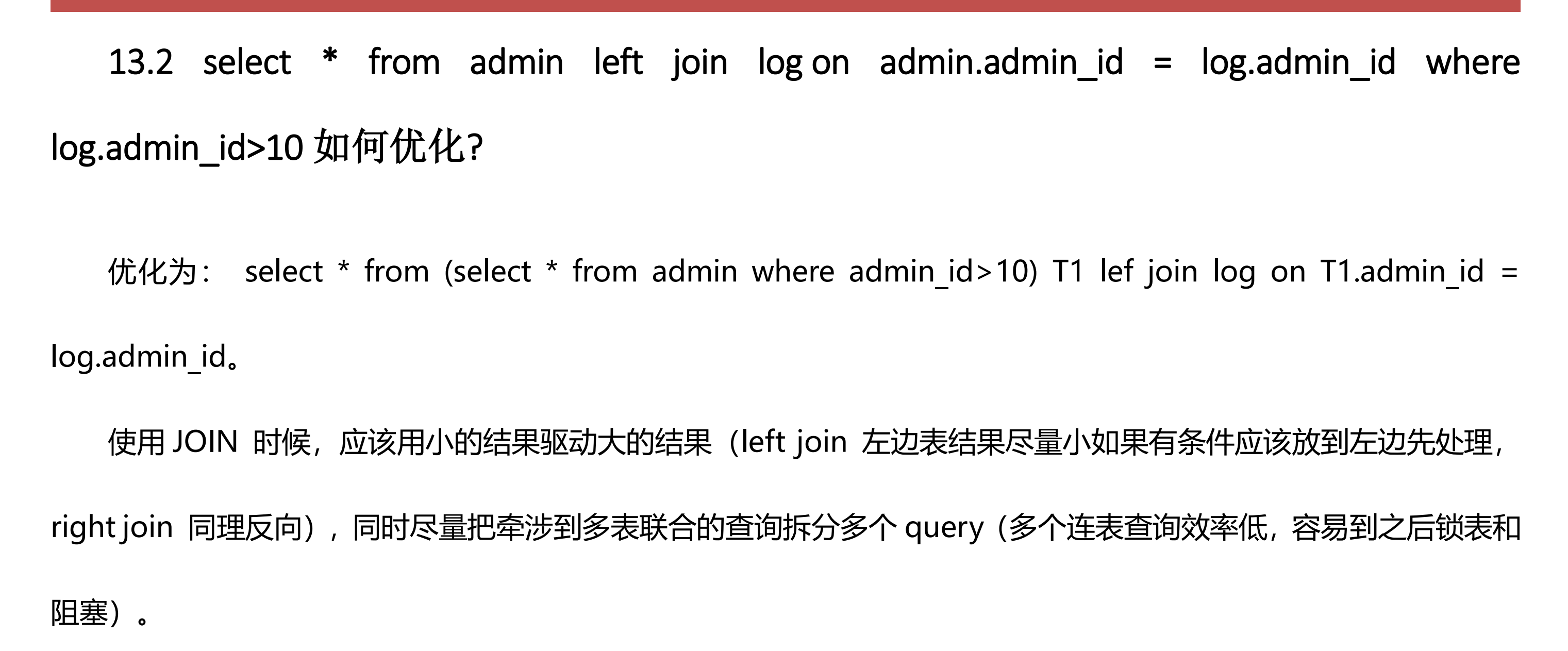
串行读

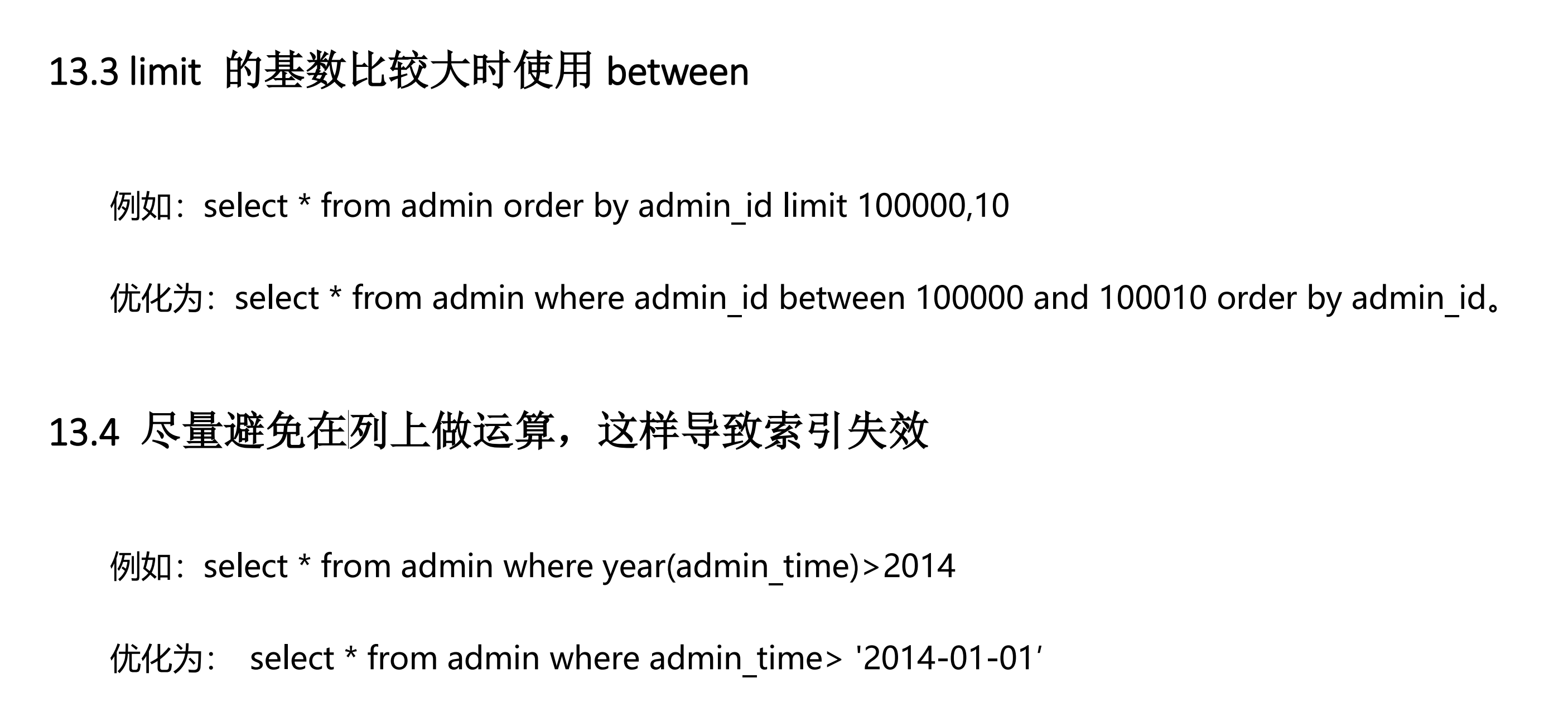
# 存储过程

# 触发器



# 优化





# 如何提高MySql的安全性

# Oracle分页

Rownum

# 存储过程和存储函数

存储过程作为一个独立的部分来执行，函数作为查询的一部分来调用

函数必须有返回值，过程没有

函数可以单独执行，过程必须通过execute

函数可以嵌入到sql中，过程不行

可以将比较复杂的查询写成函数，然后到存储过程中调用这些函数。

# 存储过程和sql的对比

## 提高性能

Sql语句在创建过程中进行分析和编译，存储过程是预编译的，在首次运行一个存储过程时，查询优化器对其进行分析、优化，并给出最终被存在系统表中的存储计划，这样在执行过程中便可以节省此开销

## 降低网络开销

存储过程调用时，只需提供存储过程名和必要的参数，从而可降低网络的流量

## 便于进行代码移植

数据库专业人员可以随时对存储过程进行修改，但对应用程序源代码却没有任何影响，从而极大的提高了程序的可移植性

## 更强的安全性

1. 系统管理员可以对执行的某一个存储过程进行权限限制，避免非授权用户访问数据
2. 在通过网络调用过程时，只有对执行过程的调用是可见的。无法看到表和数据库名称
3. 使用过程参数有助于避免sql注入攻击

劣势：

存储过程需要专门的数据库开发人员进行维护，但实际情况是往往有程序员兼职

设计逻辑变更，修噶存储过程没sql灵活

# 提高查询效率

## 数据库设计方面

1. 对查询进行优化，尽量避免全盘扫描，考虑在where和order by列上建索引
2. 尽量避免在where字句上进行null判断，否则放弃索引进行全盘扫描。可以在设计时候，该字段为null时用0代替或者其他代替。
3. 尽量使用数字型字段，字符型会逐个比较字符串中的每一个字符，数字型只需要比较一次

## Sql语句方面

## Java方面

