



黑马程序员

《MySQL 数据库原理、设计与应用（第2版）》

教学设计

课程名称：_____

授课年级：_____

授课学期：_____

教师姓名：_____

年 月



课题名称	第 11 章 数据库优化	计划课时	8 课时
教学引入	在数据库的学习中，不仅要学会对数据的基本操作，还要根据实际需求，对数据库进行优化，从而使 MySQL 服务器的性能得到充分发挥。数据库优化主要包括在创建数据表时选择合适的存储引擎、给字段添加索引、合理利用锁机制、对数据库进行分区和分表、整理数据碎片以及 SQL 优化。本章对数据库优化的相关内容进行详细讲解。		
教学目标	<ul style="list-style-type: none"> ● 使学生熟悉存储引擎的基本概念，能够说明什么是存储引擎 ● 使学生掌握存储引擎的基本使用，能够给数据表使用合适的存储引擎 ● 使学生熟悉索引的基本概念，能够归纳索引的分类 ● 使学生掌握索引的基本使用，能够创建索引、查看索引和删除索引 ● 使学生熟悉索引的使用原则，能够归纳使用索引时的注意事项 ● 使学生熟悉锁机制的概念，能够解释表级锁和行级锁的区别 ● 使学生掌握锁机制的使用，能够给数据表添加合适的锁类型 ● 使学生了解分表技术，能够说出水平分表和垂直分表的实现方式 ● 使学生了解分区技术，能够说出对数据分区的实现原理 ● 使学生掌握分区的方法，能够创建分区、增加分区和删除分区 ● 使学生掌握数据碎片的整理，能够通过命令整理数据碎片 ● 使学生掌握 SQL 优化的方法，能够使用这些方法提高 SQL 的性能 		
教学重点	<ul style="list-style-type: none"> ● InnoDB 存储引擎 ● MyISAM 存储引擎 ● 创建索引 ● 查看索引 ● 删除索引 ● 表级锁 ● 行级锁 ● 创建分区 ● 增加分区 ● 删除分区 ● 整理数据碎片 ● 慢查询日志 ● 通过 performance_schema 进行查询分析 		
教学难点	<ul style="list-style-type: none"> ● 表级锁 ● 行级锁 		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		



第一课时

(存储引擎概述、MySQL 支持的存储引擎、InnoDB 存储引擎、MyISAM 存储引擎)

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

提到“引擎”一词可能会联想到发动机，它是一个机器的核心部分。例如，直升机和火箭，它们都有各自的引擎。当建造不同的机器时要选择合适的引擎，比如直升机不能使用火箭的引擎，火箭也不能使用直升机的引擎。

在 MySQL 中，存储引擎是数据库的核心，在创建数据表时也需要根据不同的使用场景选择合适的存储引擎。本节课讲解存储引擎的概念、MySQL 支持的存储引擎和常用的存储引擎。

三、新课讲解

知识点 1-存储引擎概述

教师通过 PPT 讲解存储引擎的概念。

知识点 2- MySQL 支持的存储引擎

教师通过 PPT 讲解 MySQL 支持的存储引擎。

讲解使用 SHOW 语句查看 MySQL 支持的存储引擎。

讲解 MySQL 支持的存储引擎：

- (1) InnoDB 存储引擎。
- (2) MyISAM 存储引擎。
- (3) MEMORY 存储引擎。
- (4) MRG_MYISAM 存储引擎。
- (5) CSV 存储引擎。
- (6) FEDERATED 存储引擎。
- (7) PERFORMANCE_SCHEMA 存储引擎。
- (8) BLACKHOLE 存储引擎。
- (9) ARCHIVE 存储引擎。

知识点 3- InnoDB 存储引擎

教师通过 PPT 结合实际操作的方式讲解 InnoDB 存储引擎。

1. 存储格式

通过实际操作演示让数据表共用同一个表空间文件，关闭 InnoDB 存储引擎的独立表空间。

2. 表空间设置

- (1) 自动扩展表空间。
- (2) 将数据存储到另一个文件。

通过实际操作演示当表空间文件 ibdata1 达到 12MB 时，将数据添加到另一个指定的表空间文件 ibdata2 中，当 ibdata2 达到 50MB 时再自动扩展。

3. 多版本并发控制。

知识点 4- MyISAM 存储引擎

教师通过 PPT 讲解 MyISAM 存储引擎。

讲解 MyISAM 存储引擎的概念和数据表的 3 个文件。

四、归纳总结



教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过高校教辅平台（<http://tch.ityxb.com>）布置本节课作业以及下节课的预习作业。

第二课时

（索引概述、索引结构、创建索引、查看索引、删除索引、索引的使用原则）

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

在实际生活中，为了能够在书籍中快速找到想要的内容，通常会通过目录查找内容。在 MySQL 中，为了在大量数据中快速找到指定的数据，可以使用索引。本节课对索引进行详细讲解。

三、新课讲解

知识点 1-索引概述

教师通过 PPT 讲解索引概述。

根据创建索引的语法的不同，可以将索引分为 5 种：

- 普通索引。
- 唯一索引。
- 主键索引。
- 全文索引。
- 空间索引。

根据创建索引的字段个数不同分类：

- 单列索引。
- 复合索引。

知识点 2-索引结构

教师通过 PPT 结合实际操作的方式讲解索引结构。

- （1）B+树索引。
- （2）哈希索引。

知识点 3-创建索引

教师通过 PPT 结合实际操作的方式讲解创建索引。

- （1）创建数据表的同时创建索引

讲解创建数据表的同时创建索引的语法。

通过实际操作分别演示创建数据表的同时创建单列索引和复合索引。

- （2）给已存在的数据表创建索引

讲解使用 CREATE INDEX 语句给已存在的数据表创建索引的语法。

通过实际操作演示给 shop 数据库中的 sh_goods 数据表创建单列的唯一索引。

- （3）修改数据表的同时创建索引

讲解使用 ALTER TABLE 语句在修改数据表的同时创建索引的语法。

通过实际操作分别演示修改数据表的同时创建单列索引和复合索引。

知识点 4-查看索引

教师通过 PPT 结合实际操作的方式讲解查看索引。



讲解查看索引的语法。

通过实际操作演示查看 sh_goods 表的所有索引信息。

知识点 5-删除索引

教师通过 PPT 结合实际操作的方式讲解删除索引。

(1) 使用 ALTER TABLE 语句删除索引

讲解使用 ALTER TABLE 语句删除索引的语法。

通过实际操作演示使用 ALTER TABLE 语句删除 sh_goods 数据表中名称为 ft_index 的全文索引，删除索引后，使用 SHOW CREATE TABLE 语句查看 sh_goods 数据表的创建语句，验证 ft_index 索引是否删除成功。

(2) 使用 DROP INDEX 语句删除索引

讲解使用 DROP INDEX 语句删除索引的语法。

通过实际操作演示使用 DROP INDEX 语句删除 sh_goods 数据表中名称为 name_index 的普通索引，删除索引后，使用 SHOW CREATE TABLE 语句查看 sh_goods 数据表的创建语句，验证 name_index 索引是否删除成功。

知识点 6-索引的使用原则

教师通过 PPT 结合实际操作的方式讲解索引的使用原则。

在使用索引时要遵循的原则：

- (1) 频繁使用的字段适合创建索引。
- (2) 数字类型的字段适合创建索引。
- (3) 存储空间较小的字段适合创建索引。
- (4) 重复值较高的字段不适合创建索引。
- (5) 频繁更新的字段不适合创建索引。

要想使索引生效，在查询时还有一些注意事项：

- (1) 查询时保证字段的独立。
- (2) 模糊匹配查询中通配符的使用。
- (3) 分组查询时排序的设置。

四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

第三、四课时

(锁机制概述、表级锁、行级锁)

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

数据库是一个多用户使用的共享资源。当多个用户并发存取数据时，在数据库中可能会产生多个事务同时存取同一数据的情况。若对并发操作不加控制就可能会读取和存储不正确的数据，破坏数据库的一致性，此时可以通过给数据表加锁来保证数据库的一致性。本节课对锁机制进行讲解。

三、新课讲解

知识点 1-锁机制概述



教师通过 PPT 讲解锁机制概述。

讲解 MySQL 中常见的锁和 MySQL 常用的存储引擎支持的锁类型。

知识点 2-表级锁

教师通过 PPT 结合实际操作的方式讲解表级锁。

以 MyISAM 存储引擎的数据表为例，讲解如何添加“隐式”表级锁和“显式”表级锁。

通过实际操作演示在 mydb 数据库中创建 table_lock 数据表，该数据表使用 MyISAM 存储引擎，给 table_lock 数据表添加表级锁后，查看数据的读写情况。

知识点 3-行级锁

教师通过 PPT 结合实际操作的方式讲解行级锁。

以 InnoDB 存储引擎的数据表为例，讲解如何添加“隐式”行级锁和“显式”行级锁。

通过实际操作演示添加行级排他锁时客户端 A 和客户端 B 执行 SQL 语句的状态。

四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过高校教辅平台（<http://tch.ityxb.com>）布置本节课作业以及下节课的预习作业。

第五课时

（分表技术、分区概述、创建分区、增加分区、删除分区）

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

随着时间的推移，数据库中创建的数据会越来越多，而单张数据表存储的数据是有限的，当数据表中的数据达到一定的量级（如百万级）时，即使添加了索引，查询数据时依然会很慢。此时可以考虑使用分表技术，根据不同的需求对数据表进行拆分，从而达到分散数据表压力的目的，提升数据库的访问性能。除了使用分表技术外，还可以使用 MySQL 本身支持的分区技术提高数据库的整体性能。本节课对分表技术和分区技术进行详细讲解。

三、新课讲解

知识点 1-分表技术

教师通过 PPT 讲解分表技术。

（1）水平分表。

（2）垂直分表。

知识点 2-分区概述

教师通过 PPT 讲解分区概述。

知识点 3-创建分区

教师通过 PPT 结合实际操作的方式讲解创建分区。

讲解创建数据表的同时创建分区的语法。

讲解每种算法对应的分区字段的语法。

讲解使用 RANGE 分区算法的语法。



通过实际操作演示创建 LIST 分区和 HASH 分区。

知识点 4-增加分区

教师通过 PPT 结合实际操作的方式讲解增加分区。

讲解给已创建的数据表增加分区的语法。

通过实际操作演示给 mydb 数据库中的 p_list 和 p_hast 数据表添加分区。

知识点 5-删除分区

教师通过 PPT 结合实际操作的方式讲解删除分区。

讲解删除分区的语法。

讲解仅清空各分区表中的数据，不删除对应的分区文件的语法。

通过实际操作演示删除 mydb.p_list 表中的分区。

四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

五、布置作业

教师通过高校教辅平台（<http://tch.ityxb.com>）布置本节课作业以及下节课的预习作业。

第六课时

（整理数据碎片、慢查询日志、通过 performance_schema 进行查询分析）

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接引入的方式导入新课

在 MySQL 中，当使用 DELETE 语句删除数据时，仅删除了数据表中保存的数据，数据所占用的存储空间仍然会被保留。如果项目中长期进行删除数据的操作，索引文件和数据文件都将产生“空洞”，形成很多不连续的碎片，这样会造成数据表占用的空间很大，但数据表中的记录数却很少的问题。

在项目的开发阶段，由于数据量比较小，开发人员更注重功能的实现，项目上线后，随着用户数量的不断增加，数据库中的数据量也不断增加，有些 SQL 语句查询数据的速度越来越慢，影响整个项目的性能。因此，对这些有问题的 SQL 语句进行优化成为急需解决的问题。

本节课对整理数据碎片和 MySQL 优化的相关内容进行讲解。

三、新课讲解

知识点 1-整理数据碎片

教师通过 PPT 讲解整理数据碎片。

通过案例演示数据碎片的整理。

知识点 2-慢查询日志

教师通过 PPT 方式讲解慢查询日志。

通过实际操作演示查看慢查询日志的开启状态、慢查询日志文件所在目录、查看慢查询日志超时时间、查看慢查询日志内容。

知识点 3-通过 performance_schema 进行查询分析

教师通过 PPT 结合实际操作的方式讲解通过 performance_schema 进行查询分析，演示如何监控和收集用户的历史事件。

四、归纳总结

教师回顾本节课所讲的知识，并通过测试题和实际操作的方式引导学生解



答问题并给予指导。

五、布置作业

教师通过高校教辅平台（<http://tch.ityxb.com>）布置本节课作业以及下节课的预习作业。

第七、八课时（上机练习）

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习考察学生对知识点的掌握情况。

上机一：（考察知识点为动手实践：数据库优化实战）

形式：单独完成

具体要求：

请结合前面所学知识完成动手实践，要求如下。

（1）在 mydb 数据库中创建 my_user 用户表，数据表的字段有 id、name 和 pid，添加 200 万条测试数据。

（2）设置查询时间超过 0.5 秒的为慢查询，执行一系列的查询语句，找到慢查询语句。

（3）通过 performance_schema 进行查询分析，获取语句执行的精确时间。

（4）优化慢查询语句，提高查询效率。

上机二：（考察知识点为表级锁）

形式：单独完成

具体要求：

（1）创建 table_lock 数据表并插入两条测试数据

（2）重新打开两个客户端 A 和 B，在客户端 A 中为 mydb.table_lock 添加读锁。

（3）在客户端 A 中执行查询和更新操作，查看其他未锁定的数据表。

（4）在客户端 B 中执行查询和更新操作。

（5）在客户端 A 中释放锁，查看客户端 B 的执行结果。

（6）在客户端 A 中为 mydb.table_lock 添加写锁，执行更新和查询操作。

（7）在客户端 B 中执行查询和更新操作。

（8）在客户端 A 中释放锁，查看客户端 B 的执行结果。

上机三：（考察知识点为行级锁）

形式：单独完成

具体要求：

1. 添加隐式行级锁

（1）创建 row_lock 数据表并插入 5 条测试数据。

（2）重新打开两个客户端 A 和 B，在客户端 A 中开启事务并修改 cid 等于 3 的 name 值，会隐式地为 mydb.row_lock 设置行级排他锁。

（3）在客户端 B 中开启事务，删除 cid 等于 2 和 3 的记录。

（4）在客户端 A 和客户端 B 中回滚事务。

2. 添加显式行级锁

（1）在客户端 A 中开启事务，为 cid 等于 3 的记录添加行级排他锁。

（2）在客户端 B 中开启事务，为 cid 等于 2 的记录添加隐式行级排他锁。

（3）在客户端 B 中为 row_lock 数据表添加表级读锁。

（4）在客户端 A 和客户端 B 中回滚事务。

上机四：（考察知识点为通过 performance_schema 进行查询分析）



	<p>形式：单独完成</p> <p>具体要求：</p> <p>（1）创建 test_user 用户，授予 test_user 用户查询 shop 数据库的权限。</p> <p>（2）向 setup_actors 数据表中添加一条新的记录，更新 setup_instruments 数据表和 setup_consumers 数据表的相关配置项。</p> <p>（3）重新打开一个新的命令行窗口，使用 test_user 用户登录 MySQL，执行要分析的 SQL 语句。</p> <p>（4）SQL 语句执行完成后，在原有的命令行窗口中查看 SQL 语句的耗时情况，查看这个 SQL 语句在执行过程中的状态和消耗时间。</p>
教学后记	