###### 内容回顾

|  |
| --- |
| 1、mysql架构介绍 |

###### 内容介绍

|  |
| --- |
| 1、mysql逻辑架构  2、mysql存储引擎  3、sql预热  4、索引分类  5、性能分析 |

###### Mysql逻辑架构

|  |
| --- |
| 1、总体概览    2、利用show profile 查看sql的执行周期  （1）修改配置文件/etc/my.cnf  新增一行：query\_cache\_type=1 重启mysql  （2）开启执行周期  先开启 show variables like '%profiling%';  set profiling=1;  （3）运行sql  select \* from xxx ;      （4）命中缓存条件  Sql一模一样 |

###### Mysql存储引擎

|  |
| --- |
| 1、MyISAM和InnoDB区别 |

###### Sql预热

|  |
| --- |
| 1、表关联  graphic  #求所有人物对应的掌门名称（查询效率待确认）  #1 No3  SELECT c.`name`,ab.name ceoname FROM t\_emp c LEFT JOIN  (SELECT a.`name`,b.`id` FROM t\_emp a  INNER JOIN t\_dept b ON b.`CEO`=a.`id`)ab  ON c.`deptId`=ab.id  ;  #2 No4  SELECT ab.name,c.`name` ceoname FROM  (SELECT a.`name`,b.`CEO` FROM t\_emp a  LEFT JOIN t\_dept b ON a.`deptId`=b.`id`)ab  LEFT JOIN t\_emp c ON ab.ceo= c.id;  #3 No2  SELECT a.`name`,c.`name` ceoname FROM t\_emp a  LEFT JOIN t\_dept b ON a.`deptId`=b.`id`  LEFT JOIN t\_emp c ON b.`CEO`=c.`id`;  #4 No1  SELECT a.`name`,(SELECT c.name FROM t\_emp c WHERE c.id=b.`CEO`)ceoname  FROM t\_emp a  LEFT JOIN t\_dept b ON a.`deptId`=b.`id`;  作业：  1、8个sql，要求用格式：以 姓名.sql 命名，由各组组长收取，打包发出  2、思考题：向数据表中插入100万数据，怎样做插入数据能更快些 |

###### 索引简介

|  |
| --- |
| 1、是什么  索引（Index）是帮助MySQL高效获取数据的数据结构  2、数据结构对比    3、索引优缺点 |

###### 索引结构

|  |
| --- |
| 4、索引结构  （1）btree：非叶子节点：数据、向下指针、指向数据指针  （2）b+tree：非叶子节点：数据、向下指针    Mysql索引结构选择b+tree  （3）时间复杂度  （4）聚簇索引与非聚簇索引 |

###### 索引分类

|  |
| --- |
| 1、查看索引  SHOW INDEX FROM t\_emp;  2、创建索引  （1）单值索引  CREATE INDEX idx\_name ON t\_emp(NAME);  （2）唯一索引  CREATE UNIQUE INDEX idx\_empno ON t\_emp(empno);  （3）主键索引  （4）复合索引  CREATE INDEX idx\_age\_deptId\_name ON t\_emp(age,deptId,NAME); |

###### 性能分析

|  |
| --- |
| 1、是什么  使用EXPLAIN关键字可以模拟优化器执行SQL查询语句，从而知道MySQL是  如何处理你的SQL语句的。分析你的查询语句或是表结构的性能瓶颈  2、能干什么    3、字段  （1）id    （2）type    （3）key\_len  Where后筛选条件用上索引的长度  字段数值越大，查询越快  （4）rows  整个sql物理扫描行数  字段数值越小，查询越快  （5）Extra |