|  |  |
| --- | --- |
| 客户名称 | 贵州盘江煤电集团有限责任公司(MySQL) |
| 问题描述 | 使用外部数据源连接WIN版本的MySQL时，测试连接无法通过 |
| 处理过程 | 帮助修复过程发现，到数据库的端口是可以正常连接，但是通过数据库命令连接时，连接长时间卡主无结果，此时数据库查看有异常的连接解析到的用户名称是unauthenticated user状态，下方test用户为正常的连接请求状态。    查看数据库的安装日志发现安装过程有部分缺失系统包的异常被忽略，帮助重新安装VC环境，安装.Net framework 4.0 后再次验证恢复正常。  在安装数据库软件时，一定要注意在异常信息全部处理之后再进行下一步安装和使用，否则后期出现一些功能问题，会影响环境的稳定运行，并排查问题带来很大的困扰。 |
| 客户名称 | 贵州盘江煤电集团有限责任公司(MySQL) |
| 问题描述 | 项目反映EM的MySQL数据库启动异常 |
| 处理过程 | 协助启动数据库时发现日志中大量的表空间文件异常的错误信息    [ERROR] [MY-011971] [InnoDB] Tablespace 'innodb\_undo\_001' Page [page id: space=4294967279, page number=258] log sequence number 14969701562 is in the future! Current system log sequence number 14962727436  强制打开数据库，并马上使用mysqldump逻辑导出数据，导出过程有异常的表是损坏或者数据不一致的表，已无法完全恢复，只能恢复部分数据。  数据文件被物理修改或删除，表空间文件的版本与MySQL的程序版本不一致容易导致上面的问题，尤其是云桥和EM的客户在Windows环境部署MySQL，此问题经常发生。  切记任何情况下都不要轻易操作底层的数据库物理文件！一切的物理数据文件备份要建立在数据库服务关闭的基础上！ |
| 客户名称 | 深圳市晨光乳业有限公司(MySQL) |
| 问题描述 | 老云桥系统升级，wx\_wxpulic\_sendmsglog\_new表增加字段的操作无法正常修改 |
| 处理过程 | 客户的数据库版本是MySQL5.1，此版本中默认将所有的数据存放在ibdata文件中，该文件已经达到15GB，并且操作系统和数据库版本都非常老旧，该表上存有longtext和text字段，单表的大小初步估算可能有近10GB，尝试修改表结构增加字段的操作已无法完成。表结构详情如下：    后与客户协商，保留该表60天的消息数据，其他数据留作历史数据，进行修改表结构。  具体做法：    目前已发现较多起日志表缺失清理策略导致数据库故障或性能问题，MySQL修改大表结构(增删字段操作)需要重构整个表才可以完成，一些存放日志的大表一定要设置删除或者归档分表的策略，否则将会为以后的维护能带来较严重的阻碍！ |
| 客户名称 | 中海物业管理有限公司(MySQL) |
| 问题描述 | 提交流程到某个节点时数据库出现死锁，OA程序性能存在严重问题 |
| 处理过程 | 临时手工KILL产生死锁源头的会话，协助排查死锁产生的原因发现，系流程提交至此节点，项目二开嵌套了一个其他事务导致死锁频发。    死锁产生的原因：  ①会话A，update 1 nocommit ②会话B，update 2 nocommit  ③会话A，update 2 wait①  ④会话B，update 1 wait②  死锁产生，必须将其中的一个会话提交或回滚，另一个会话才得以继续执行。  如何减少死锁：  任何并发的事务场景都无法完全避免死锁，我们只能尽量减少发生的可能。  1、尽量减少一个会话持有的锁个数，分批多次提交事务。  2、尽量优化SQL，合理增加索引，减少SQL的执行时间，减少持有锁的周期。  3、避免将DML语句与SELECT语句放一个事务中，不要让SELECT额外增加锁的持有时间。  4、任何事务要么提交，要么回滚，在开启一个事务时一定要设置事务的结束和异常处理操作。  5、高并发的场景要设置一些'排队'机制，不要批量把所有的更新都一次性提交到数据库。 |
| 客户名称 | 重庆紫光化工股份有限公司(MySQL) |
| 问题描述 | 偶发的系统出现卡顿缓慢甚至有宕机的现象，多次触发系统的性能监控告警。 |
| 处理过程 | 帮助优化过程发现更新文档的阅读数的SQL并发很高，并存在性能问题，协助进行了优化：  改造前的SQL和执行计划      改造后的SQL和执行计划：      优化思路：  MySQL在处理嵌套和多层子查询时会优先JOIN的方式进行表关联处理，而此处的SQL经与开发同事讨论，b部分的子句查询结果一般会很小，如果将该结果先查询处理后，作为temp表与外层的表再进行关联，是我们想要的优化效果，为了让MySQL使用我们的逻辑进行查询，需要将b部分的子句额外增加一层嵌套，防止MySQL将SQL解析为JOIN的方式进行先关联再过滤，改造后的查询速度可以提高近5倍。 |
| 客户名称 | 德力西电气有限公司（SQLServer） |
| 问题描述 | 自定义报表-转圈或者显示404，客户E9系统使用的是E8的报表链接 |
| 处理过程 | 1.根据提供的慢SQL，分析下来主要是exists子查询中有三张大表的关联非常慢所致（两个千万级别的表，关联一张百万级别的，且基本无过滤数据的条件，几乎等同于全表关联筛选）；    2.综合测试后，需要借助于中间表的方案方能满足，如下：  2.1.先创建middle\_workflow\_request\_user\_lists中间表：  create table middle\_workflow\_request\_user\_lists(userid int,requestid int, primary key(userid,requestid))  2.2.通过添加两个计划任务，来定期同步关联后的数据到中间表（以下只是参考，实际执行频率根据客户需要调整）    2.3.最后报表调用逻辑改成如下的方式： |
| 客户名称 | 北京华大九天软件有限公司（SQLServer） |
| 问题描述 | formtable\_main\_15 我们3年前用的流程的一个表单，现在这个流程不用了，但是这个数据我们需要查询，昨天不小心UPDATE改错了，但是有备份这个完整的数据库也包括这张表格。希望从旧的数据库中恢复这张表格到现在的数据库。 |
| 处理过程 | 1.先从测试数据库服务器上提取formtable\_main\_15表包含数据的生成SQL脚本，并批量替换表名为tmp\_ft\_main\_15；  2.正式数据库服务器上执行刚刚的生成的SQL脚本，完成tmp\_ft\_main\_15表的创建和数据导入；  3. 正式数据库服务器上先对要操作的表执行数据备份操作，如下：  select \*into formtable\_main\_15\_bak\_20210507 from formtable\_main\_15  4.执行表数据还原操作：    5.强烈建议在手动对数据库中某张表做更新、删除操作前，做好备份数据的操作，各数据库备份方式，参考如下： |
| 客户名称 | 万马联合控股集团有限公司（SQLServer） |
| 问题描述 | 流程删除后，点恢复，在待办里出现很多流程号一样的流程 |
| 处理过程 | 1.流程删除后，点恢复后，在待办里面显示很多流程号一样的流程，OA前端表现如下：    2.查看后台ecology日志，报《事务在触发器中结束，批处理已中止》的错误，如下：    3.猜测和操作表上的触发器逻辑有关，最后查下来系workflow\_requestbase表上有个workflow\_requestbase\_getpinyin的触发器，在OA标准逻辑的基础上做了二次开发调整，在触发器的结尾处加上了commit关键字，如下：    而触发器中是不需要显式加commit、rollback关键字的，本身触发器的业务逻辑就和当前表的操作绑定在同一个事务中，在去掉该commit关键字后，功能测试恢复正常。 |
| 客户名称 | 赛轮集团股份有限公司（SQLServer） |
| 问题描述 | 客户定时同步数据出现比较严重的重复现象，需求协助删除重复数据，客户现在是正式环境，已经产生正式数据，无法全部删除再重新同步 |
| 处理过程 | 1.获取表字段清单  DECLARE @ColumnList VARCHAR(MAX)  SELECT @ColumnList = ISNULL(@ColumnList+ ',', '') + a.name FROM sys.columns a join sys.tables b on a.object\_id = b.object\_id where b.name ='uf\_grjf'  SELECT @ColumnList  2.对要操作的表先备份  select \* into uf\_grjf\_bak\_20210507 from uf\_grjf  3.按自增id以外的其它所有字段分组取最小的id值，并把数据放于tmp\_id\_lists0507中间表    4.对tmp\_id\_lists0507中间表的minid字段创建索引  create index idx\_minid on tmp\_id\_lists0507(minid)  5.执行重复数据删除操作，只保留分组后的最小id值  delete a from uf\_grjf a left join tmp\_id\_lists0507 b on a.id = b.minid where b.minid is null |
| 客户名称 | 博锐尚格科技股份有限公司（MySQL） |
| 问题描述 | 客户台账数据误删的恢复方法，不能直接导入 |
| 处理过程 | 1.通过客户环境现有的navicat客户端查询工具，在测试库中先行复制uf\_clzd表数据到uf\_clzd\_bak\_20210525\_o表中，如下：  create table uf\_clzd\_bak\_20210525\_o like uf\_clzd;  insert into uf\_clzd\_bak\_20210525\_o select\*From uf\_clzd  2.客户测试库和正式库在两台数据库服务器上，利用navicat本身的复制功能，通过选中测试库中刚刚新建的uf\_clzd\_bak\_20210525\_o表，右键复制，然后到正式库的表对象管理里面，右键选择粘贴，来实现uf\_clzd\_bak\_20210525\_o该表数据测试库到正式库的迁移；    3.先备份当前要在正式库环境操作的uf\_clzd表：  create table uf\_clzd\_bak\_20210525\_n like uf\_clzd;  insert into uf\_clzd\_bak\_20210525\_n select\*From uf\_clzd  4.正式库环境执行对uf\_clzd表的还原操作：  truncate table uf\_clzd;  insert into uf\_clzd select \*from uf\_clzd\_bak\_20210525\_o; |