

数据库第八次上机

关于作业提交

TASK1

TASK2

请在PDF/WORD等任何方便助教阅读查看的文档中按照各个作业要求提交相关内容，记得标清题号。

若为PDF/WORD单文档文件直接提交即可，其他提交压缩包，命名为“学号_姓名_第*次实验”。

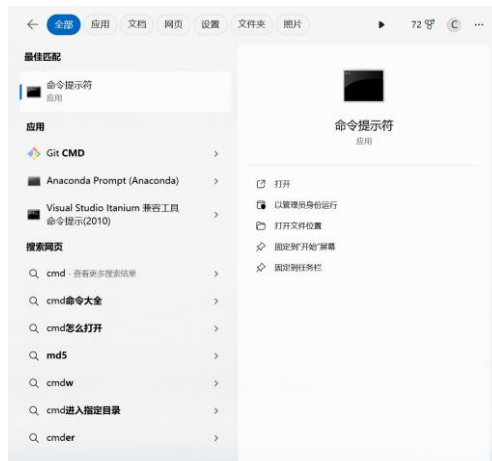
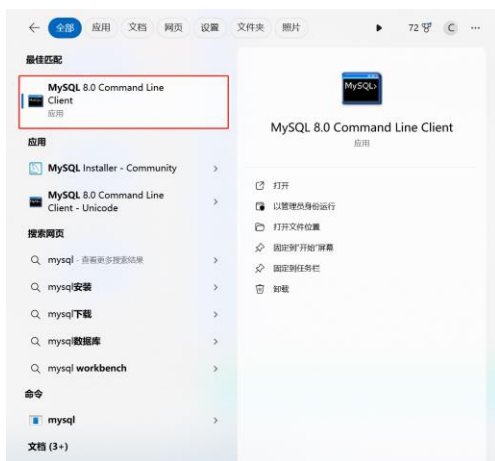
提交网址：软件学院云平台**第八次上机** scs.buaa.edu.cn
(按要求提交)

作业截止时间为**本周末24:00之前**，提交方式为提交到云平台。

本次上机概览

本次的上机内容为：

- 数据库备份和恢复
- 不同管理工具备份恢复方式和实现不同，为了实验的统一性，本次实验使用MySQL命令行界面和Windows/Linux命令行界面进行实验；



本次上机任务

Task1 逻辑备份（本次实验都使用MySQL命令行界面和Windows/Linux命令行界面进行实验）：

<本任务使用**mysqldump**实现备份恢复>

- 1.创建一个新的数据库db9，并创建一个表user（uid，name，money），添加两个用户，A用户余额为2000，B用户余额为3000
- 2.使用**mysqldump**工具备份数据库
- 3.删除该表
- 4.恢复数据库

提交内容：使用到的SQL语句/命令行语句，恢复前后通过show/select语句打印表内容来验证恢复成功与否，将上述内容统一通过pdf/word等任意方便查看的文档形式提交。

数据库备份

物理备份： 对数据库操作系统的物理文件（数据文件、日志文件等）进行的备份（针对出现问题时需要快速恢复大型数据库）

划分： 冷备份、热备份、温备份

(1) 冷备份： 数据库关闭的状态下进行的备份

(优点： 备份速度快，恢复时简单，通过直接打包数据库文件)

(2) 热备份： 数据库处于运行状态时进行的备份

(备份依赖于数据库的日志文件)

(3) 温备份： 数据库锁定表格的状态下进行的备份

(不可往表格中写入但是可以读取的状态)

逻辑备份： 对数据库的逻辑组件进行的备份

(对create database、create table、insert等语句信息进行的备份，适用于可编辑数据值或者表结构较小的数据量)

数据库备份

备份策略的三种角度

(1) **完全备份**: 又叫完整备份, 对整个数据库进行备份, 保存的是备份完成时刻的数据库 (是差异备份和增量备份的前提)

特点: 备份恢复简单, 占用磁盘空间大, 备份时间较长

(2) **差异备份**: 备份上一次完全备份后被修改过的所有文件

特点: 备份数据量大, 恢复时需要恢复上次的完全备份 + 最近一次的差异备份

(3) **增量备份**: 以上一次完全备份或者上一次增量备份的时间为节点, 只备份时间节点数据发生变化的部分 (通过日志文件备份)

特点: 备份数据量小, 占用空间小, 备份速度快, 恢复时需要一次完整备份 + 所有增量备份依次恢复

数据库备份

数据备份的方法

- (1) 物理冷备份：（使用tar工具直接打包）
- (2) 专用备份工具：（mysqldump或者mysqlhotcopy）
- (3) 通过二进制日志进行增量备份
- (4) 通过第三方工具进行备份

本次实验涉及2，3

逻辑备份

以mysqldump工具为例做使用说明：

首先windows平台下该工具**通过cmd调用**，而不是mysql的命令行，请注意！

需要将目录**cd到mysql安装目录的bin文件夹**下，如C:\Program Files\MySQL\MySQL Server 8.0\bin

关于mysql安装路径在哪，请参考：

<https://www.bilibili.com/read/cv14959506>

逻辑备份

mysqldump工具命令:

导出命令:

mysqldump -u[用户名] -p[密码] 数据库名 [表名] > 导出的文件名.sql

如没设定保存文件路径, 默认在Mysql的bin目录下

还原命令:

mysql -u[用户名] -p[密码] [数据库名] < 导出的文件名.sql(路径)

注意空格

更多命令参数和示例可参考:

<https://blog.csdn.net/Wyongkang/article/details/123558313>

如果遇到了“<”运算符是为将来使用而保留的 错误, 请使用cmd为非powershell

本次上机任务

Task2 增量备份（本次实验都使用MySQL命令行界面和Windows/Linux命令行界面进行实验）：

<本任务使用mysqlbinlog实现备份恢复>

- 1.请先确保开启了日志服务并配置了环境变量（见下页），并创建一个表user2（uid, name, money），添加两个用户，A用户余额为 2000，B用户余额为3000（建议重新建表，避免日志未记录）
- 2.删除A用户
- 3.删除B用户
- 4.使用日志通过位置恢复B用户
- 5.使用日志通过时间恢复A用户

提交内容：使用到的SQL语句/命令行语句，恢复前后通过show/select语句打印表内容来验证恢复成功与否，将上述内容统一通过pdf/word等任意方便查看的文档形式提交。

增量备份

增量备份需要使用到日志文件，首先确保开启了日志服务，通过如下命令验证是否开启：

show variables like '%log_bin%';

```
mysql> show variables like '%log_bin%';
```

Variable_name	Value
log_bin	ON
log_bin_basename	C:\ProgramData\MySQL\MySQL Server 8.0\Data\DESKTOP-TE550GE-bin
log_bin_index	C:\ProgramData\MySQL\MySQL Server 8.0\Data\DESKTOP-TE550GE-bin.index
log_bin_trust_function_creators	OFF
log_bin_use_v1_row_events	OFF
sql_log_bin	ON

6 rows in set (0.00 sec)

其中log_bin_basename所在路径就是日志文件所在路径

如log_bin显示OFF,请修改配置文件my.ini, 具体步骤请参考：

https://blog.csdn.net/qq_16268979/article/details/114802174

mysqlbinlog

请****首先确保mysql已经加入环境变量，否则命令行无法使用mysqlbinlog等工具****
如何配置环境变量：<http://c.biancheng.net/view/7143.html>

Binlog日志，即binary log，是二进制日志文件，有两个作用，一个是**增量备份**，另一个是**主从复制**，即主节点维护一个binlog日志文件，从节点从binlog中同步数据，也可以通过binlog日志来恢复数据。

****mysqlbinlog工具相关命令在windows命令行执行，mysql命令行无法使用****

后续执行命令出现文件权限问题参考

https://blog.csdn.net/weixin_30588655/article/details/114323462

mysqlbinlog

基于位置的恢复：

```
mysqlbinlog --no-defaults --start-position=[**] --stop-position=[**]  
"日志文件路径" | mysql -u[用户名] -p[密码]
```

举例：

```
mysqlbinlog --no-defaults --start-position=236 --stop-position=466  
"C:\ProgramData\MySQL\MySQL Server 8.0\Data\DESKTOP-TE650GE-bin.000001" |  
mysql -uroot -p123456
```

参数说明：

--stop-position: 表示恢复时间节点之前的数据也就是恢复到指定id结束
--start-position: 表示恢复时间节点之后的数据也就是从指定id开始恢复（具体是啥见后续查看日志）

****日志文件路径：** 请注意, 默认日志存储路径在 C:\ProgramData\MySQL\MySQL Server 8.0\Data 或类似路径, 执行命令需要明确指定该路径, 否则会报错找不到日志文件

查看日志

通过查看日志获取之前的运行命令记录，从而在误操作后能够找到需要补救执行的语句的位置或时间信息，从而通过mysqlbinlog工具恢复。

****以下命令均在MySQL命令行执行****



MySQL 8.0 Command Line Client

查看日志（可获取位置信息）

show binlog events in '日志文件名';

```
mysql> show binlog events in 'DESKTOP-TE550GE-bin.000001';
```

Log_name	Pos	Event_type	Server_id	End_log_pos	Info
DESKTOP-TE550GE-bin.000001	4	Format_desc	1	126	Server ver: 8.0.28, Binlog ver: 4
DESKTOP-TE550GE-bin.000001	126	Previous_gtids	1	157	
DESKTOP-TE550GE-bin.000001	157	Anonymous_Gtid	1	234	SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
DESKTOP-TE550GE-bin.000001	234	Query	1	358	use `smwy`; create table sm(id int , money int) /* xid=827 */
DESKTOP-TE550GE-bin.000001	358	Anonymous_Gtid	1	437	SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
DESKTOP-TE550GE-bin.000001	437	Query	1	519	BEGIN
DESKTOP-TE550GE-bin.000001	519	Query	1	622	use `smwy`; insert into sm values(1,2)
DESKTOP-TE550GE-bin.000001	622	Xid	1	653	COMMIT /* xid=829 */

日志查看

如想恢复delete from sm where id=1 误删除操作，则应通过mysqlbinlog执行对应插入操作，即 insert into sm values(1,2000) 对应的开始和结束位置如下图所示：

```
mysql> show binlog events in 'DESKTOP-TE550GE-bin.000001';
```

Log_name	Pos	Event_type	Server_id	End_log_pos	Info
DESKTOP-TE550GE-bin.000001	4	Format_desc	1	126	Server ver: 8.0.28, Binlog ver: 4
DESKTOP-TE550GE-bin.000001	126	Previous_gtid	1	157	
DESKTOP-TE550GE-bin.000001	157	Anonymous_gtid	1	234	SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
DESKTOP-TE550GE-bin.000001	234	Query	1	358	use `smwy`; create table sm(id int , money int) /* xid=827 */
DESKTOP-TE550GE-bin.000001	358	Anonymous_gtid	1	437	SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
DESKTOP-TE550GE-bin.000001	437	Query	1	519	BEGIN
DESKTOP-TE550GE-bin.000001	519	Query	1	622	use `smwy`; insert into sm values(1,2)
DESKTOP-TE550GE-bin.000001	622	Xid	1	653	COMMIT /* xid=829 */
DESKTOP-TE550GE-bin.000001	653	Anonymous_gtid	1	732	SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
DESKTOP-TE550GE-bin.000001	732	Query	1	814	BEGIN
DESKTOP-TE550GE-bin.000001	814	Query	1	905	use `smwy`; delete from sm
DESKTOP-TE550GE-bin.000001	905	Xid	1	936	COMMIT /* xid=831 */
DESKTOP-TE550GE-bin.000001	936	Anonymous_gtid	1	1022	SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
DESKTOP-TE550GE-bin.000001	1022	Query	1	1104	BEGIN
DESKTOP-TE550GE-bin.000001	1104	Query	1	1207	use `smwy`; insert into sm values(1,2)
DESKTOP-TE550GE-bin.000001	1207	Xid	1	1238	COMMIT /* xid=860 */
DESKTOP-TE550GE-bin.000001	1238	Anonymous_gtid	1	1317	SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
DESKTOP-TE550GE-bin.000001	1317	Query	1	1399	BEGIN
DESKTOP-TE550GE-bin.000001	1399	Query	1	1400	use `smwy`; delete from sm

mysqlbinlog

基于时间的恢复：

```
mysqlbinlog --no-defaults --start-datetime="2020-12-09 0:33:14" --stop-datetime="2022-12-09 0:33:14"  
"日志文件路径" | mysql -u[用户名] -p[密码]
```

参数说明：

--stop-datetime： 表示恢复时间节点之前的数据也就是恢复到指定时间节点结束

--start-datetime： 表示恢复时间节点之后的数据也就是从指定时间节点开始恢复
查看日志获取时间节点（具体是啥见后续查看日志）

****日志文件路径：** 请注意, 默认日志存储路径在 C:\ProgramData\MySQL\MySQL Server 8.0\Data 或类似路径，执行命令需要明确指定该路径，否则会报错找不到日志文件

如出现时间格式错误 可将="2020-12-09 0:33:14" 转化为="20201209003314"

查看日志

查看日志文件(获取时间信息):

cmd命令行执行命令:

```
mysqlbinlog --no-defaults "C:\ProgramData\MySQL\MySQL Server 8.0\Data\DESKTOP-TE550GE-bin.000001 (即要查看的日志文件路径) "
```

即可查阅日志文件, 具体的时间信息在:

```
C:\> Administrator: 命令提示符
# at 2117
#220514 15:05:24 server id 1  end_log_pos 2196 CRC32 0x9e5f15f7      Anonymous_GTID  last_committed=7      sequence_number=8      rbr_only=no
estamp=1652511924045195 immediate_commit_timestamp=1652511924045195      transaction_length=294
# original_commit_timestamp=1652511924045195 (2022-05-14 15:05:24.045195 中国标准时间)
# immediate_commit_timestamp=1652511924045195 (2022-05-14 15:05:24.045195 中国标准时间)
/*!80001 SET @@session.original_commit_timestamp=1652511924045195*//*!*/;
/*!80014 SET @@session.original_server_version=80028*//*!*/;
/*!80014 SET @@session.immediate_server_version=80028*//*!*/;
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 2196
#220514 15:05:24 server id 1  end_log_pos 2278 CRC32 0x099f57c8      Query  thread_id=50  exec_time=0  error_code=0
SET TIMESTAMP=1652511924*//*!*/;
BEGIN
/*!*/;
# at 2278
#220514 15:05:24 server id 1  end_log_pos 2380 CRC32 0xf082d3b3      Query  thread_id=50  exec_time=0  error_code=0
SET TIMESTAMP=1652511924*//*!*/;
delete from sm where id=1
/*!*/;
# at 2380
#220514 15:05:24 server id 1  end_log_pos 2411 CRC32 0x87486666      Xid = 872
```

more

可以注意到mysqlbinlog是恢复数据（或者说是重做），而不是回滚数据，很多应急场景单纯mysqlbinlog已经不再合适（也是增量备份本身的问题）。

恢复的本质是将原有的插入语句再执行一遍，而回滚则是回退到删除之前的状态。mysqlbinlog是mysql自带的命令，一般是在mysql安装目录下的bin目录里。因为我们是恢复数据，所以要找到已经删除语句的对应写入事件，将该事件再重新执行一遍即可。

要想实现真正的回滚，需要binlog2sql工具（仅供参考，可以使用不同的工具），该工具安装较为麻烦，有兴趣的同学可以尝试

https://blog.csdn.net/Interest1_wyt/article/details/124217711

关于作业提交

TASK1

TASK2

请在PDF/WORD等任何方便助教阅读查看的文档中按照各个作业要求提交相关内容，记得标清题号。

若为PDF/WORD单文档文件直接提交即可，其他提交压缩包，命名为“学号_姓名_第*次实验”。

提交网址：软件学院云平台第八次上机 scs.buaa.edu.cn
(按要求提交)

作业截止时间为本周末24:00之前，提交方式为提交到云平台。

相关参考

本次实验参考：

https://blog.csdn.net/Interest1_wyt/article/details/115028429

一般SQL语法：

<http://www.w3school.com.cn/sql/index.asp>

官方文档：

SQL Server: <https://docs.microsoft.com/zh-cn/sql/t-sql/language-elements/transactions-transact-sql?view=sql-server-2017>

MySQL: <https://dev.mysql.com/doc/refman/8.0/en/sql-syntax-transactions.html>

多用搜索引擎：

<https://cn.bing.com/>

<https://www.google.com/>

<https://www.baidu.com/>

以及数据库课程PPT