

实验三：分布式数据库数据模式匹配和事务管理

学号：202103151422

姓名：温家伟

班级：大数据分析 2101

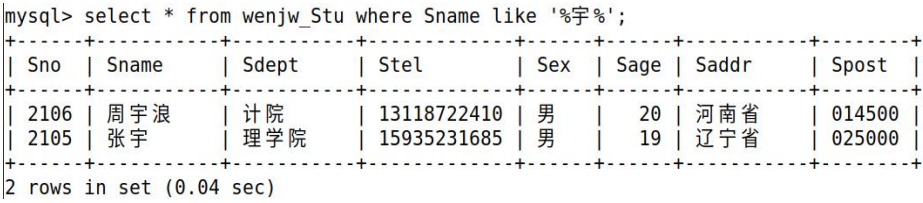
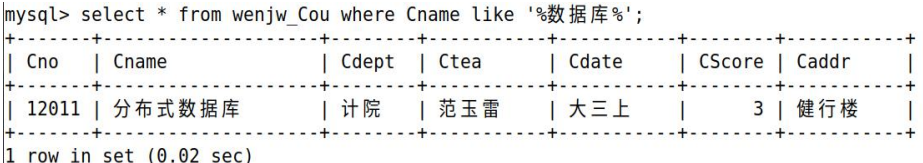
1、实验环境安装情况（用☑替换□即可）

- a) ☒ 自安装环境 ☐ 高校邦系统（如果选择后者，以下三个不用回答）
 b) ☐ 单机 ☒ 多机
 c) ☐ 单数据库实例 ☒ 两个数据库实例 ☐ 多个数据库实例
 d) ☒ 同类型数据库实例 ☐ 两个类型数据库实例 ☐ 多类型数据库实例
 e) ☐ Mycat ☒ ShardingSphere
 f) ☒ MySQL ☐ OpenGauss ☐ 其他_____

2、实验内容

- a) 针对实验运行环境（用☑替换□即可）
☒ 命令行下进行实验 ☐ Java 应用程序进行实验 ☐ Python 应用程序进行实验
☐ 其他_____
- b) 针对实验一和试验二的学生表 Stu、课程表 Cou 和学生选课表 SC，继续插入一些数据，尽量贴合实际，以备后续查询使用，插入数据之后完成如下两个查询：
 （1）利用 like 关键字查询整个库中学生表 Stu 中学生姓名 Sname 中包含“宇”字的所有学生信息；
 （2）利用 like 关键字查询整个库中课程表 Cou 中课程名 Cname 包含“数据库”三个字的所有课程信息。

能否实现：☒ 能 ☐ 不能（如果可以完成下表）

表	整个库（Mycat 或 ShardingSphere）
Stu	SQL 语句：select * from wenjw_Stu where Sname like '%宇%'; 结果（截图）： 
Cou	SQL 语句：select * from wenjw_Cou where Cname like '%数据库%'; 结果（截图）： 

- （1）利用 like 关键字查询每个库中学生表 Stu 中学生姓名 Sname 中包含“宇”字的所有学生信息；
 （2）利用 like 关键字查询每个库中课程表 Cou 中课程名 Cname 包含“数据库”三个字的所有课程信息。。

表	数据库 1	数据库 2
---	-------	-------

Stu	SQL 语句: select * from wenjw_Stu0 where Sname like '%字%'; 结果（截图）： <pre>select * from wenjw_Stu0 where Sname like '%字%'; +-----+-----+-----+-----+-----+ Sname Sdept Stel Sex Sage +-----+-----+-----+-----+-----+ 周宇浪 计院 13118722410 男 20 +-----+-----+-----+-----+-----+ 1 row in set (0.00 sec)</pre>	SQL 语句: select * from wenjw_Stu1 where Sname like '%字%'; 结果（截图）： <pre>[wenjw_db1]> select * from wenjw_Stu1 where Sname like '%字%'; +-----+-----+-----+-----+-----+-----+ Sname Sdept Stel Sex Sage Saddr +-----+-----+-----+-----+-----+-----+ 05 张宇 理学院 15935231685 男 19 辽宁省 +-----+-----+-----+-----+-----+-----+ 1 row in set (0.00 sec)</pre>
Cou	SQL 语句: select * from wenjw_Cou where Cname like '%数据库%'; 结果（截图）： <pre>mysql> select * from wenjw_Cou where Cname like '%数据库%'; +-----+-----+-----+-----+-----+ Cno Cname Cdept Ctea Cdate +-----+-----+-----+-----+-----+ 12011 分布式数据库 计院 范玉雷 大三上 +-----+-----+-----+-----+-----+ 1 row in set (0.00 sec)</pre>	SQL 语句: select * from wenjw_Cou where Cname like '%数据库%'; 结果（截图）： <pre>wenjw_db1> select * from wenjw_Cou where Cname like '%数据库%'; +-----+-----+-----+-----+-----+-----+ Cname Cdept Ctea Cdate CScore C +-----+-----+-----+-----+-----+-----+ 分布式数据库 计院 范玉雷 大三上 3 1 +-----+-----+-----+-----+-----+-----+ 1 row in set (0.00 sec)</pre>

- c) 前述实验大家都未使用 Begin Transaction/Commit/Abort 等关键字，其实大家使用的是隐式事务，默认每个 SQL 语句为一个事务。针对显示事务进行如下实验：首先要确定 Mycat 或 ShardingSphere 端口是否接受 Begin Transaction/Commit/Abort 等关键字：☒能 ☐不能
- 如果不能，允许同学在其中一个数据库上完成以下事务相关实验：
- 如果能，需要两个 Mycat 或 ShardingSphere 端口，用以分别执行两个事务：

<input checked="" type="checkbox"/> 整个库（Mycat 或 ShardingSphere）	<input type="checkbox"/> 数据库 1	<input type="checkbox"/> 数据库 2
事务 T1	事务 T2	
Begin Transaction		
按照学号查询自己的信息： SQL 语句： select * from wenjw_Stu where Sno = 2101; 查询结果截图： <pre>mysql> select * from wenjw_Stu where Sno = 2101; +-----+-----+-----+-----+-----+ Sno Sname Sdept Stel Sex +-----+-----+-----+-----+-----+ 2101 温家伟 理学院 19858180826 男 +-----+-----+-----+-----+-----+ 1 row in set (0.01 sec)</pre>		
	Begin Transaction	
	按照学号查询自己的信息： SQL 语句： select * from wenjw_Stu where Sno = 2101; 查询结果截图： <pre>mysql> select * from wenjw_Stu where Sno = 2101; +-----+-----+-----+-----+-----+ Sno Sname Sdept Stel Sex +-----+-----+-----+-----+-----+ 2101 温家伟 理学院 19858180826 男 +-----+-----+-----+-----+-----+ 1 row in set (0.01 sec)</pre>	

	<p>按照学号更新自己的手机号为任意值：</p> <p>SQL 语句：</p> <p>update wenjw_Stu set Stel = '88888888888'</p> <p>where Sno = 2101;</p> <p>更新结果截图：</p> <pre>mysql> select * from wenjw_Stu where Sno = 2101; +-----+-----+-----+-----+-----+ Sno Sname Sdept Stel Sex +-----+-----+-----+-----+-----+ 2101 温家伟 理学院 88888888888 男 +-----+-----+-----+-----+-----+ 1 row in set (0.01 sec)</pre>
<p>按照学号查询自己的信息：</p> <p>SQL 语句：</p> <p>select * from wenjw_Stu where Sno = 2101;</p> <p>查询结果截图：</p> <pre>mysql> select * from wenjw_Stu where Sno = 2101; +-----+-----+-----+-----+-----+ Sno Sname Sdept Stel Sex +-----+-----+-----+-----+-----+ 2101 温家伟 理学院 19858180826 男 +-----+-----+-----+-----+-----+ 1 row in set (0.02 sec)</pre>	
	Commit
<p>按照学号查询自己的信息：</p> <p>SQL 语句：</p> <p>select * from wenjw_Stu where Sno = 2101;</p> <p>查询结果截图：</p> <pre>mysql> select * from wenjw_Stu where Sno = 2101; +-----+-----+-----+-----+-----+ Sno Sname Sdept Stel Sex +-----+-----+-----+-----+-----+ 2101 温家伟 理学院 19858180826 男 +-----+-----+-----+-----+-----+ 1 row in set (0.02 sec)</pre>	
Commit	
	Abort
<p>按照学号查询自己的信息：</p> <p>SQL 语句：</p> <p>select * from wenjw_Stu where Sno = 2101;</p> <p>查询结果截图：</p> <pre>mysql> select * from wenjw_Stu where Sno = 2101; +-----+-----+-----+-----+-----+ Sno Sname Sdept Stel Sex +-----+-----+-----+-----+-----+ 2101 温家伟 理学院 88888888888 男 +-----+-----+-----+-----+-----+ 1 row in set (0.03 sec)</pre>	
Commit	

3、实验总结：本次实验主要是探究 MySQL 在分布式数据库搭建后的事务。

首先，我们查看隔离级别：

```
mysql> select @@transaction_isolation;
+-----+
| @@transaction_isolation |
+-----+
| REPEATABLE-READ         |
+-----+
1 row in set (0.01 sec)
```

我们可以看到这是可重复读，也就是说，一个事务里面，只要你自己不修改，什么时候读都是一样的，只有自己提交后，若其他事务对值做了修改，才会被观测到。

然后，我又测试了，insert 语句的情况，和上述实验结果相同。一般的数据库在可重复读情况的时候，无法屏蔽其他事务 insert 的数据。因为隔离性实现是对数据加锁完成的，而 insert 待插入的数据因为并不存在，那么一般加锁无法屏蔽这类问题，会造成虽然大部分内容是可重复读的，但是 insert 的数据在可重复读情况被读取出来，导致多次查找时，会多查找出来新的记录，也就是幻读。很明显，MySQL 在 RR 级别的时候，是解决了幻读问题的(解决的方式是用 Next-Key 锁(GAP+行锁))。同样的分布式的情况下，MySQL 也做到了解决幻读。

MySQL 有四种隔离方式，分别是可串行化（隔离级别最高，但效率低下）、可重复读、读提交、读未提交（效率最高，但几乎没隔离），并且，MySQL 的默认隔离级别是可重复读。