### 《数据库系统原理》实验报告

**实验题目：数据库的安全性设计**

**姓名：** **实验日期：2023 年 12 月 1 日**

**实验内容及完成情况：**（可续页）

**二、实验内容**

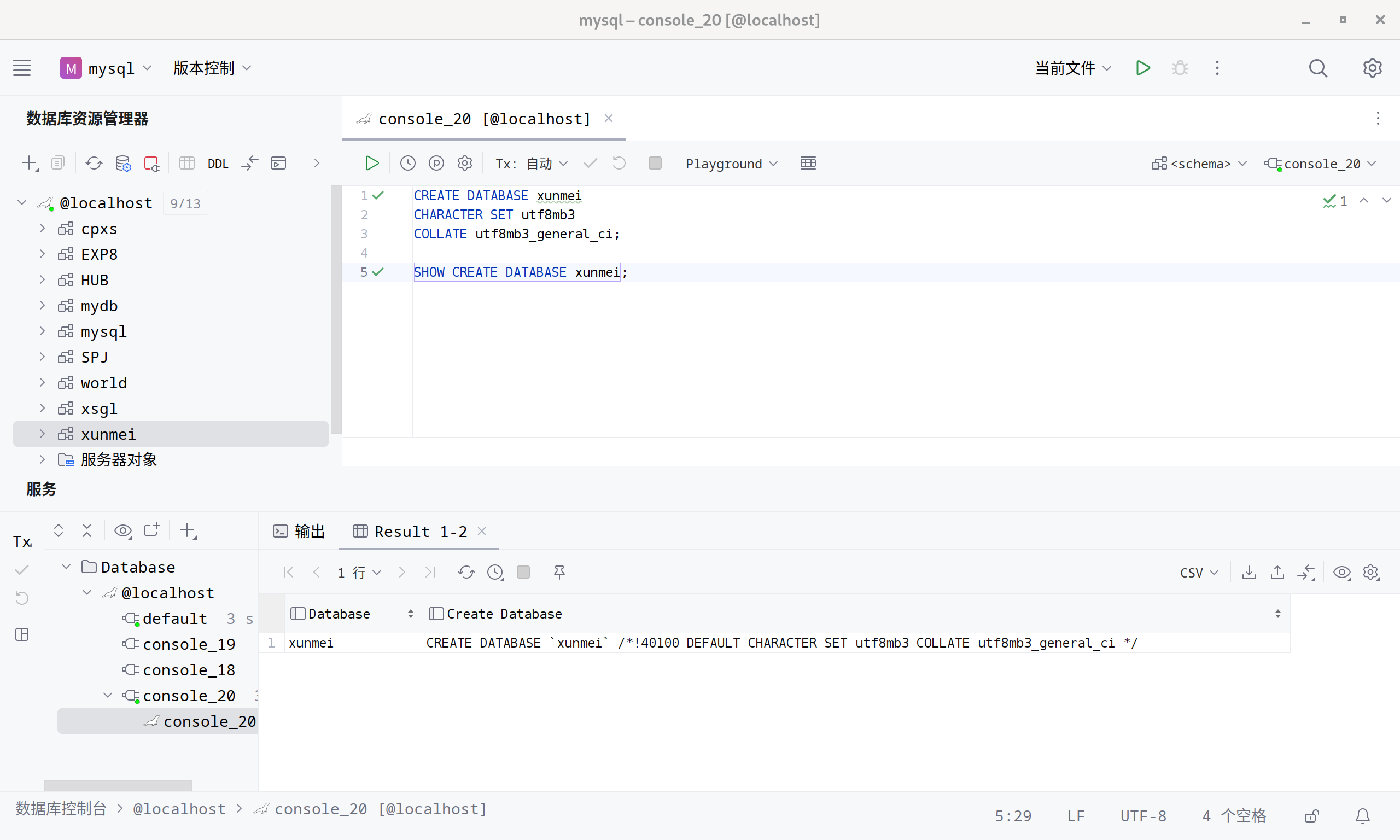
1、在Navicat中，管理建立用户和分配用户权限。

在这里，我们可以发现root是MySQL最高级别权限的用户，它拥有查看、修改和删除MySQL软件中所有数据库的权限。当需要有多个数据库，并且分配给不同的用户使用，多个用户之间只有查看自己对应数据库的权限，不相互干扰, 需要建立多个数据库和用户，给用户设置管理指定数据库的权限。

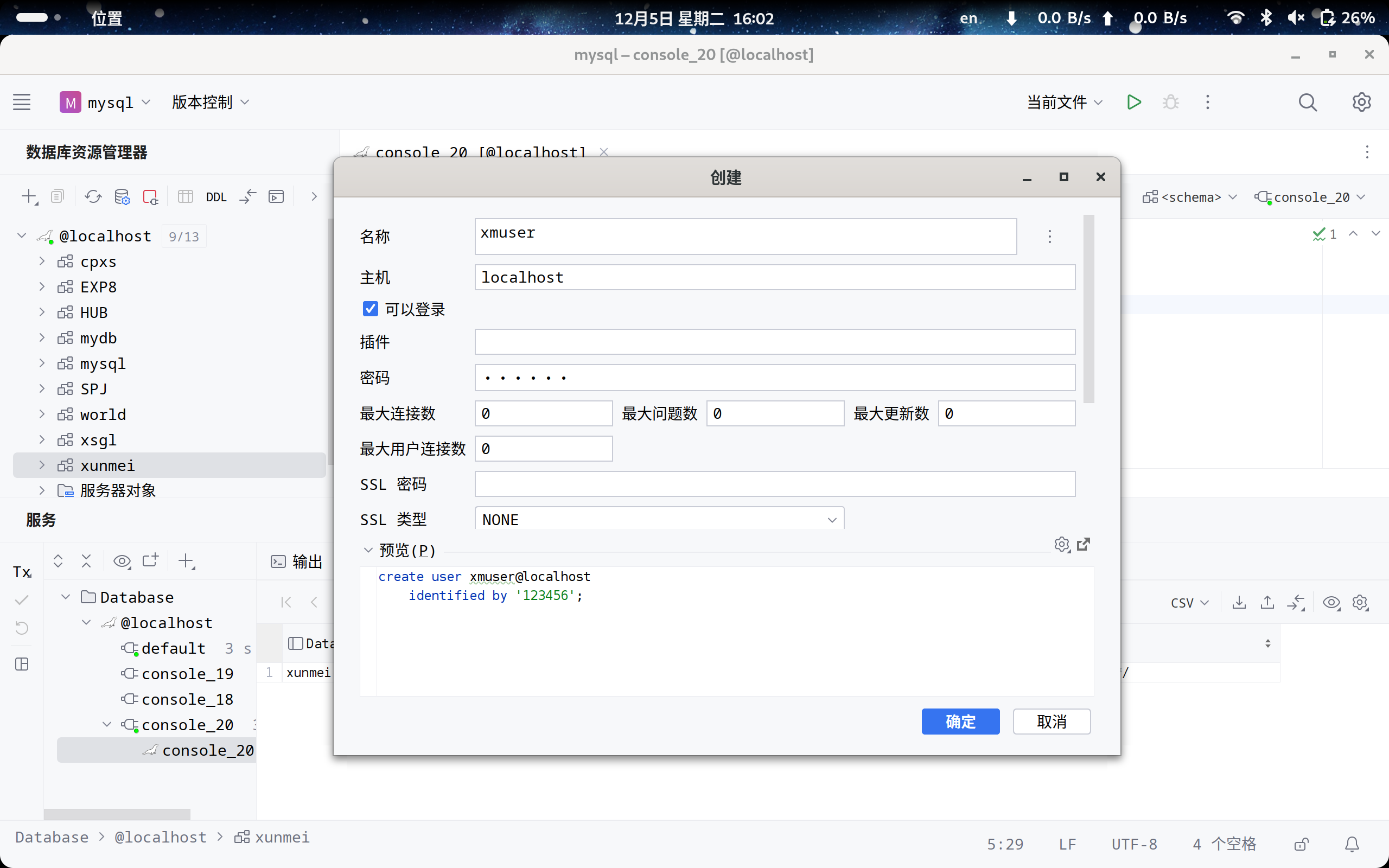
操作方法如下：

①右键点击“MySQL” 连接，选择“连接属性”，将“保存密码”取消。

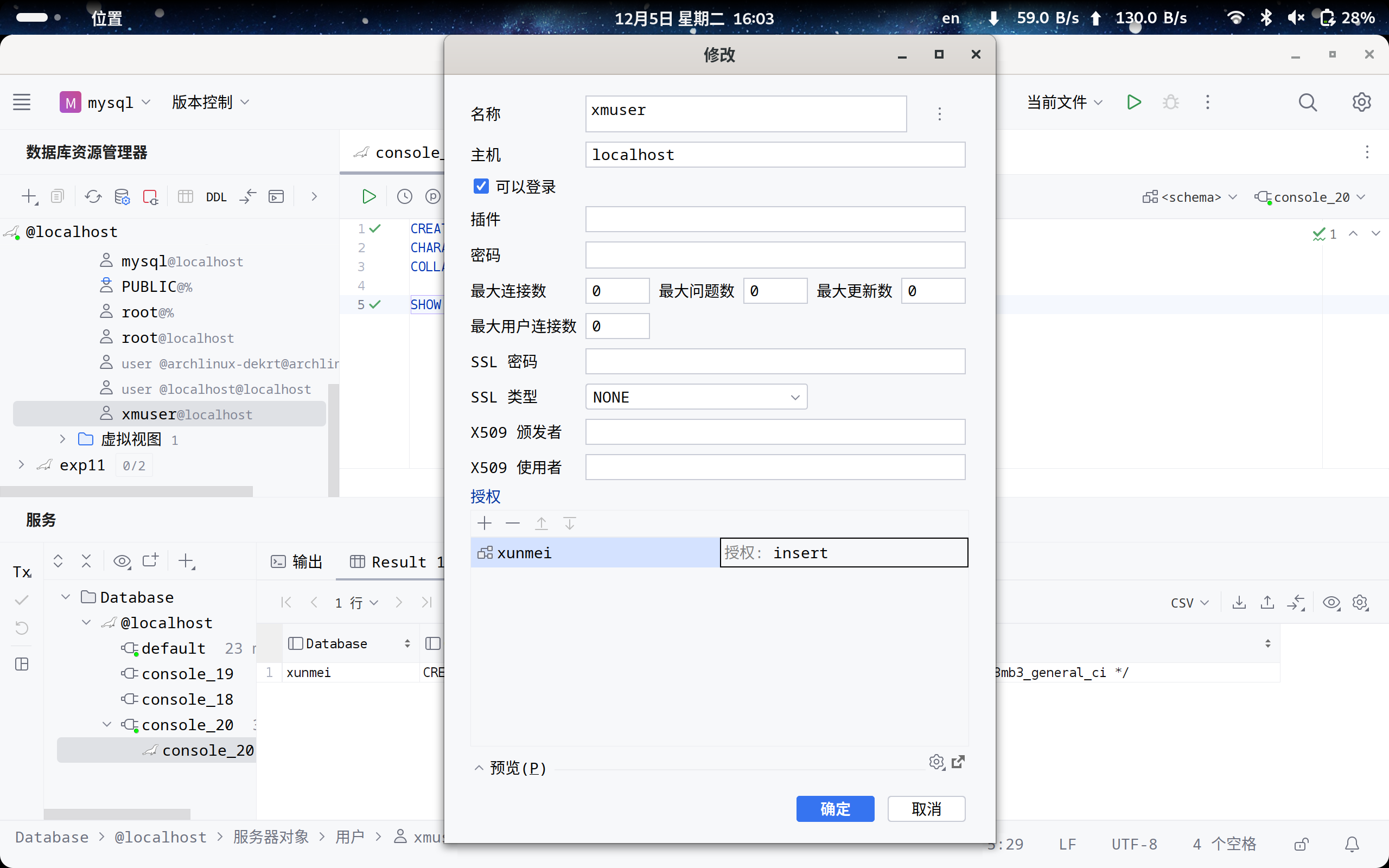
②鼠标右键点击“MySQL” 连接，选择“创建数据库”，创建名为xunmei的数据库，字符集选择utf8mb4排序规则选择utf8mb4\_general\_ci。



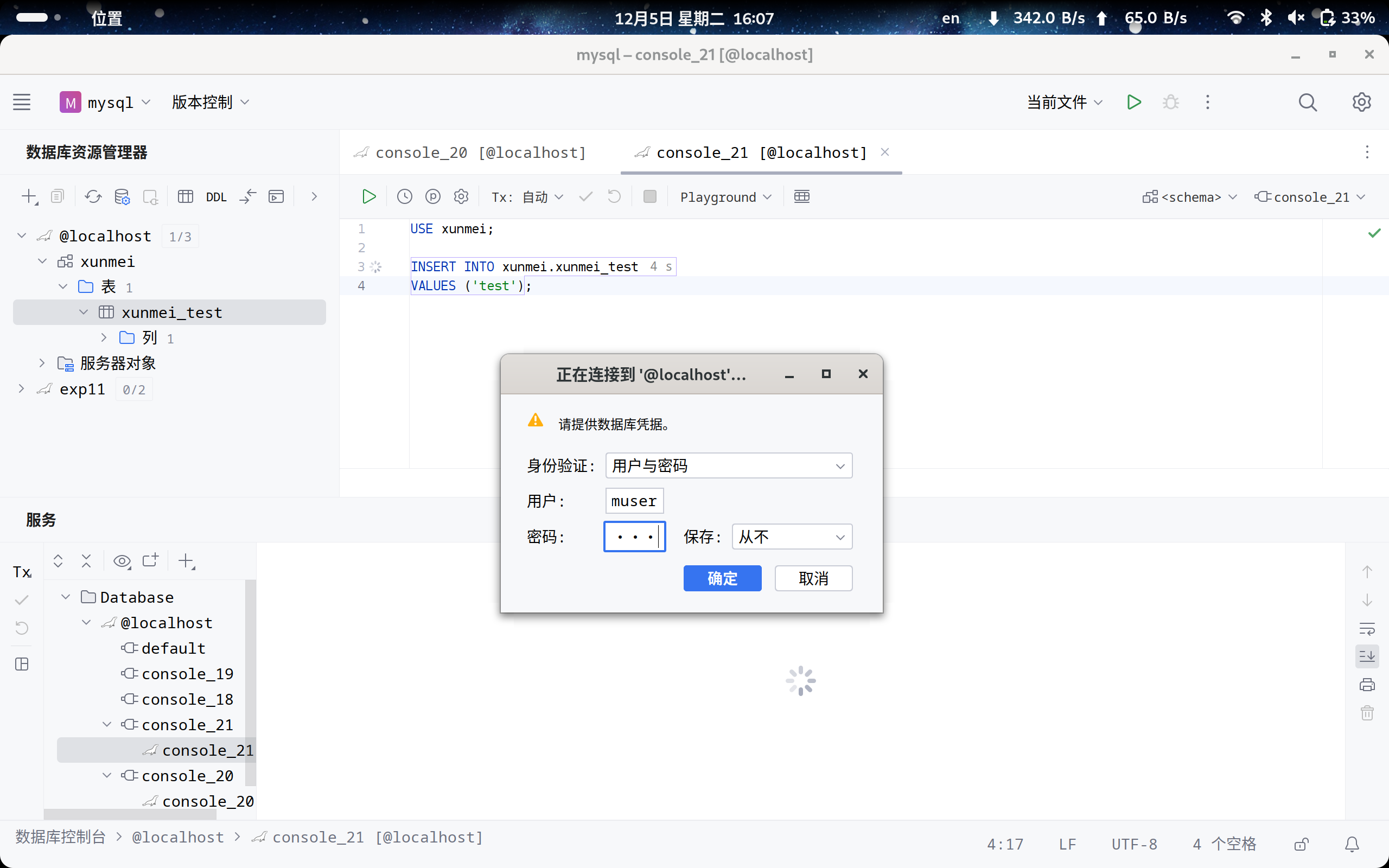
③打开数据库xunmei后，点击界面上方的“用户”—“新建用户”—输入用户名“xmuser”、主机“localhost”、密码“123456”—保存（服务器权限自行定sh义）。

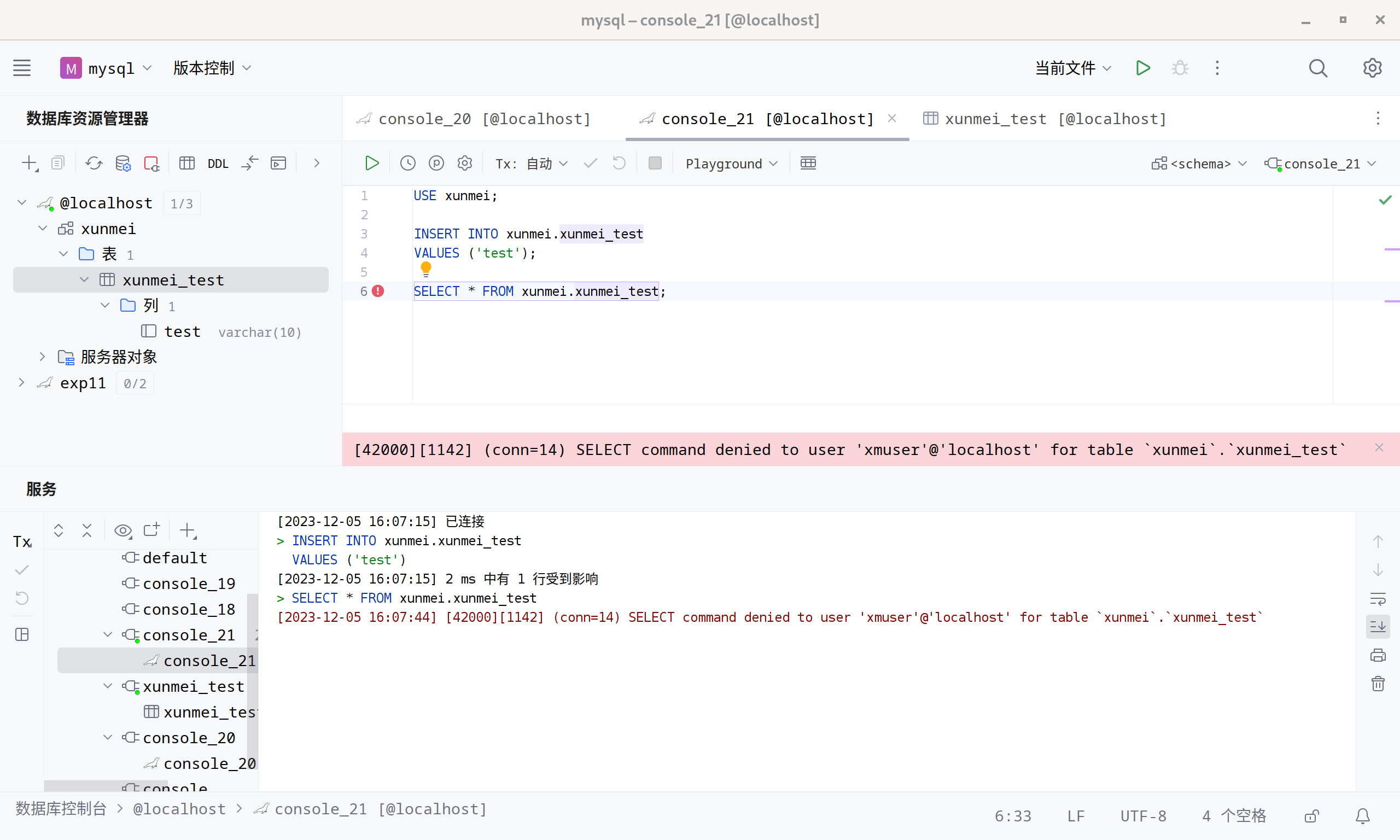


④点选新建的用户xmuser—编辑用户—权限—添加权限--选择数据库xunmei，添加相应权限（可不全选），如图所示：



⑤设置到此完成。为了验证我们的设置是否正确，我们可以用刚才建立的用户xmuser和密码123456来连接localhost主机地址（方法：关闭连接MySQl，该连接属于最高权限用户root。我们用新用户新建一个到localhost的连接来模拟不同用户使用数据库的过程）。连上后，可使用用户xmuser对数据库xunmei或其他数据库进行操作，看是否与自己设置的权限相吻合。（可只添加添加少量权限，然后尝试在xunmei里进行操作，看是否能成功）。





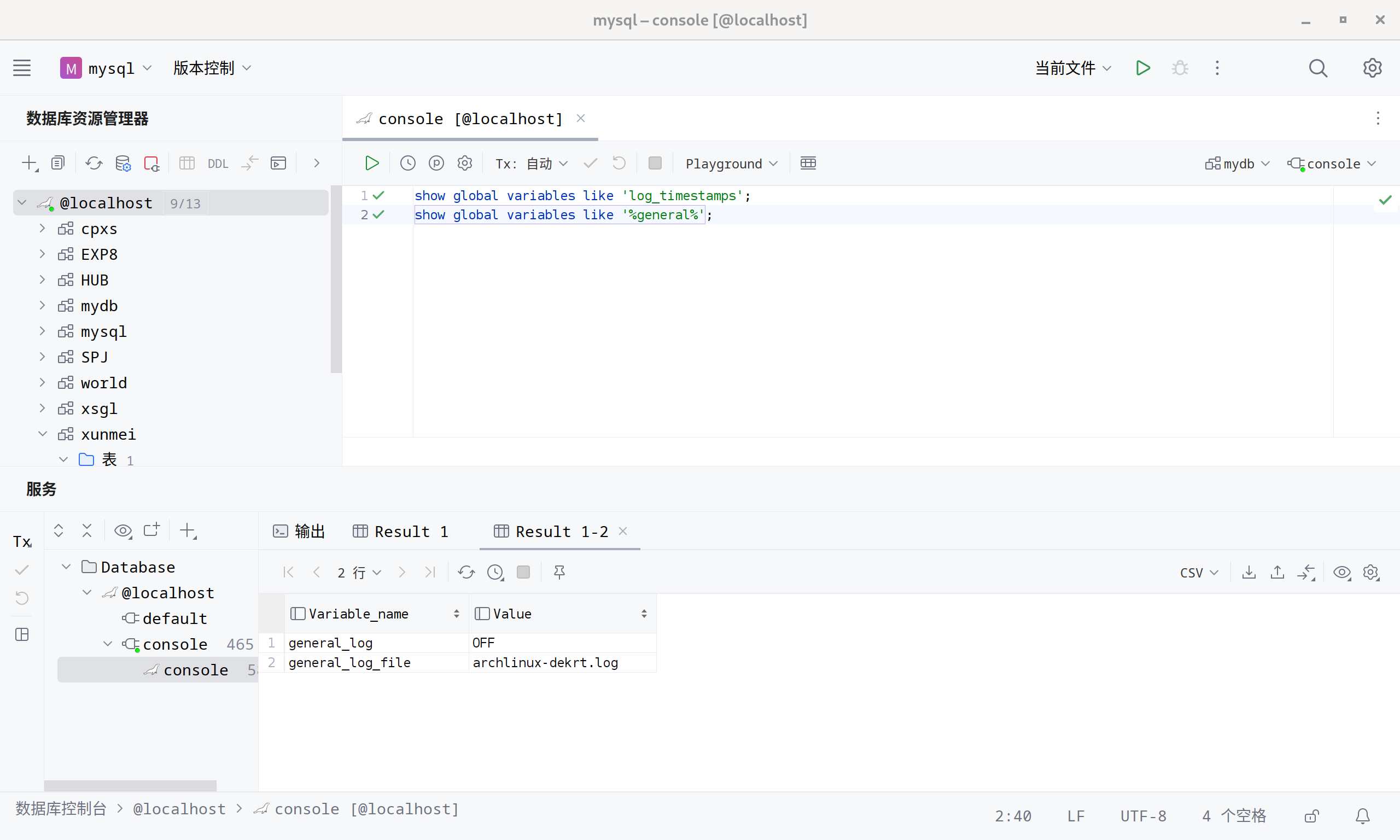
2、打开数据库审计开关，并查看存储审计信息的存储文件所在位置。

①显示当前审计开关状态。

其中log\_timestamps表示记录审计日志的时间是从那里获取，general\_log表示审计功能是否开启，general\_log\_file表示审计信息的存储文件。

show global variables like 'log\_timestamps';

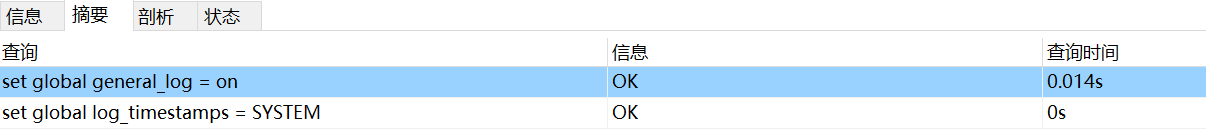
show global variables like '%general%';

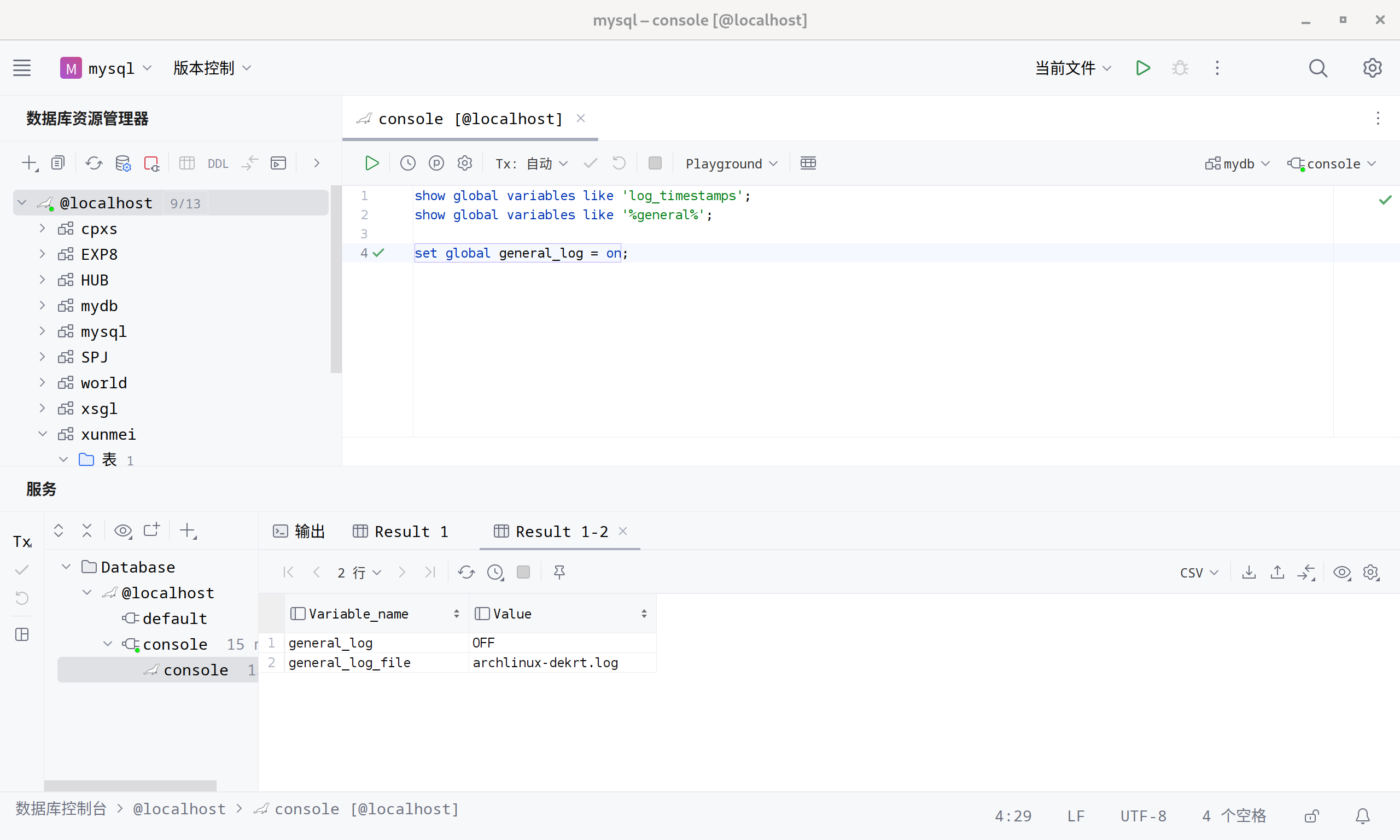


②打开审计开关

set global general\_log = on;

set global log\_timestamps = SYSTEM;

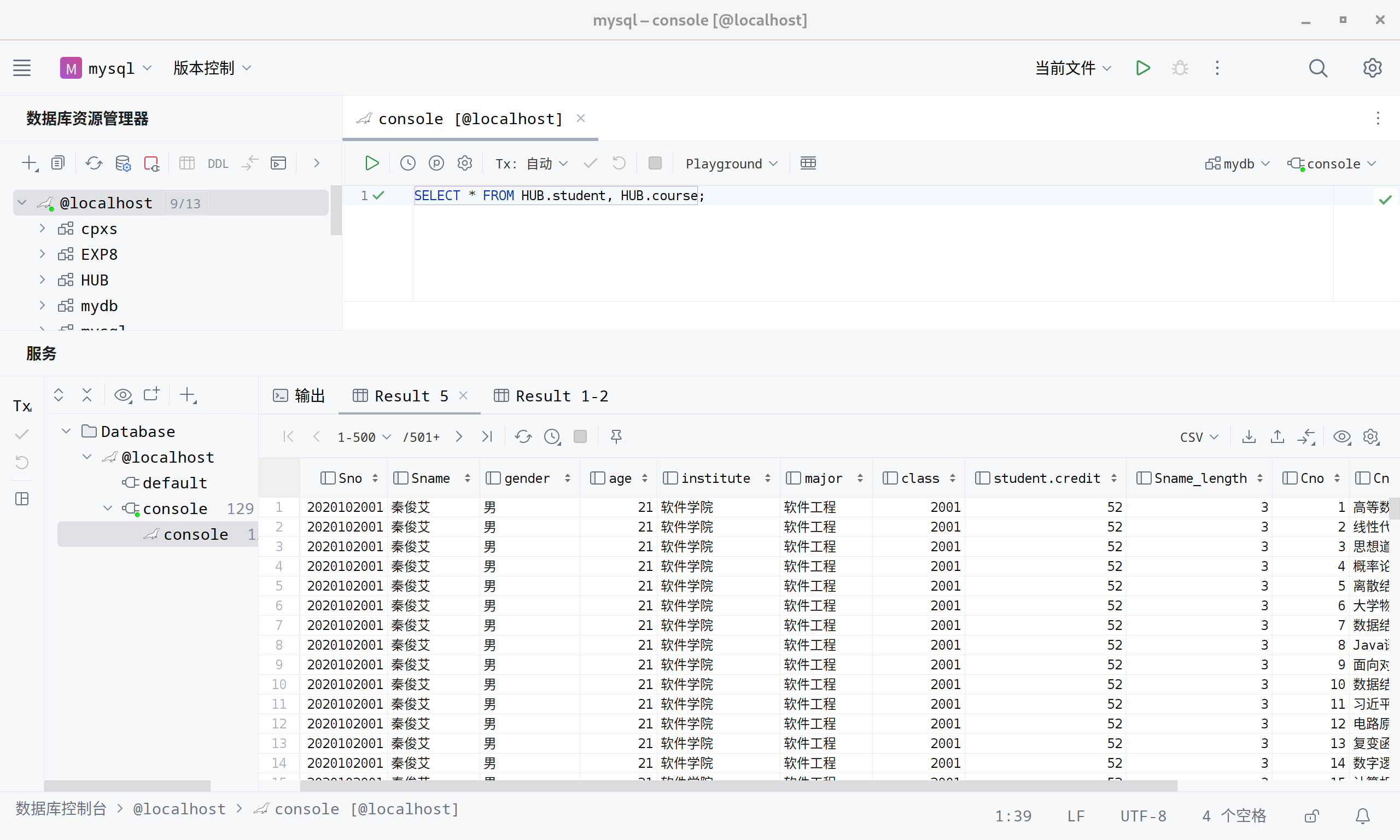




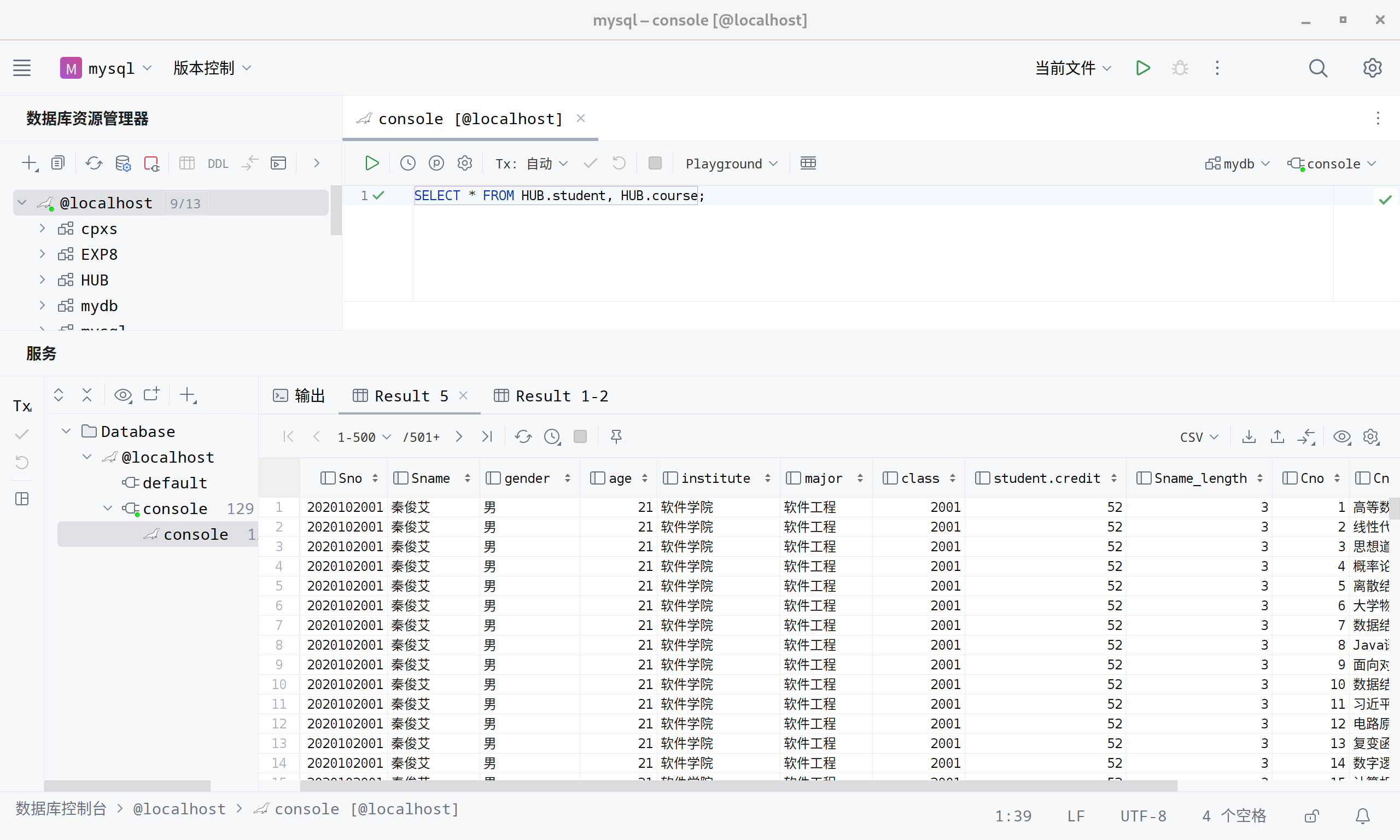
**三、课后练习题**

1、使用两个不同的用户通过Navicat和查询分析器查看hub数据库中student、course两个表的所有数据。

root用户：



test用户：

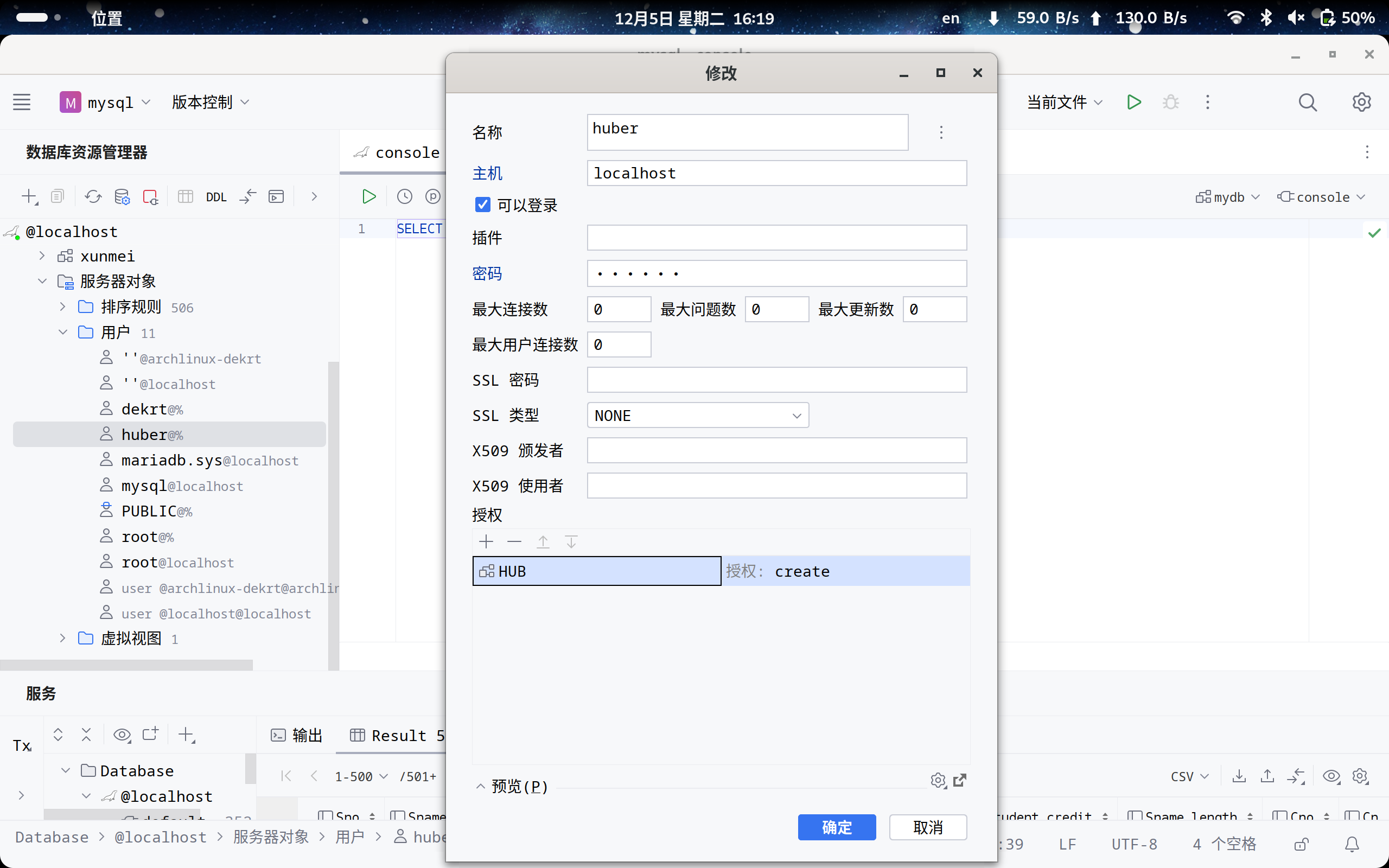


2、删除用户xmuser，尝试新建另一个用户来管理数据库、表等：

添加新用户对数据库hub的权限，比较在有无相关权限的情况之下进行操作的区别：

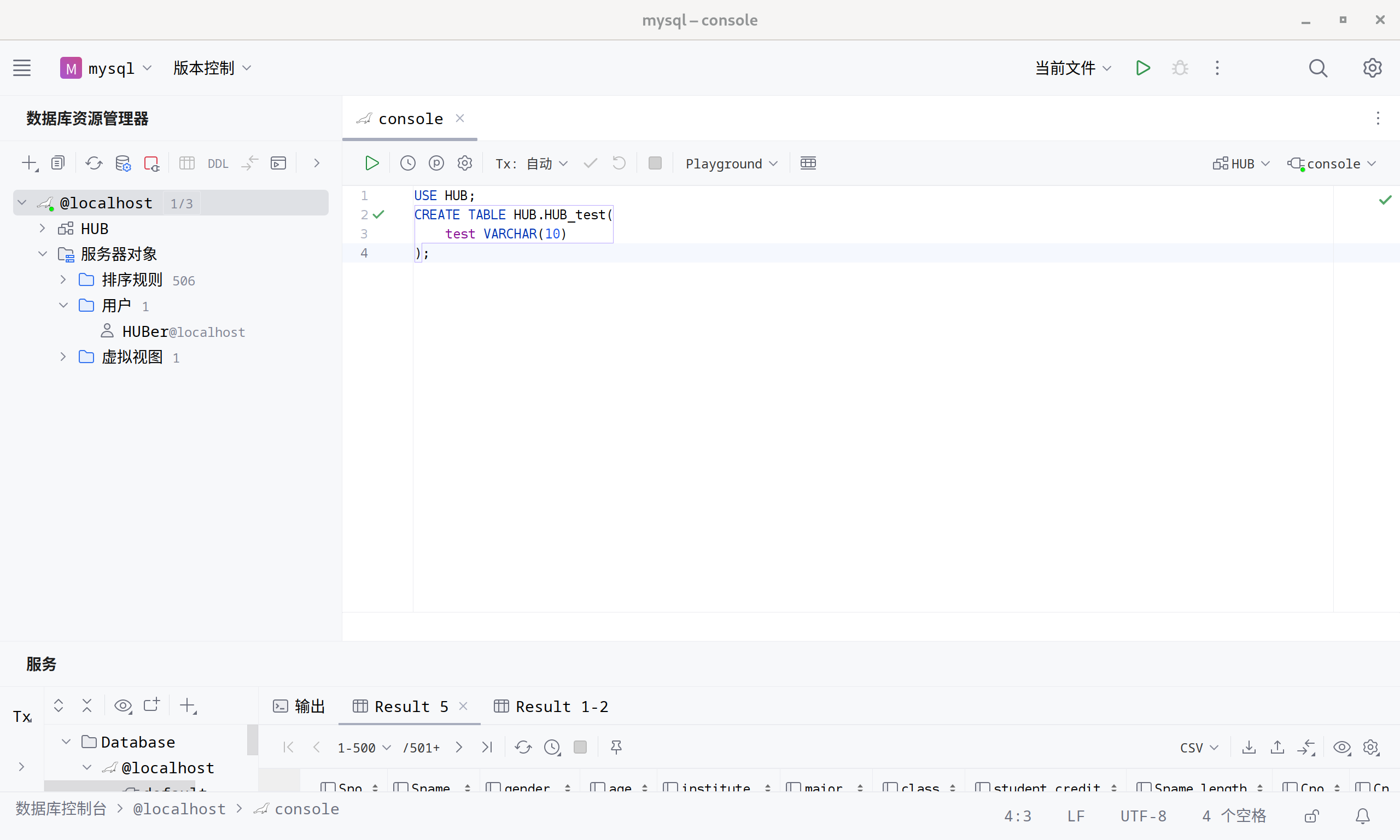
1. Create。尝试新建表
2. Update。尝试修改表数据或结构
3. ……..（自行设计，总共尝试5组，进行对比）

创建用户HUBer，授权create：



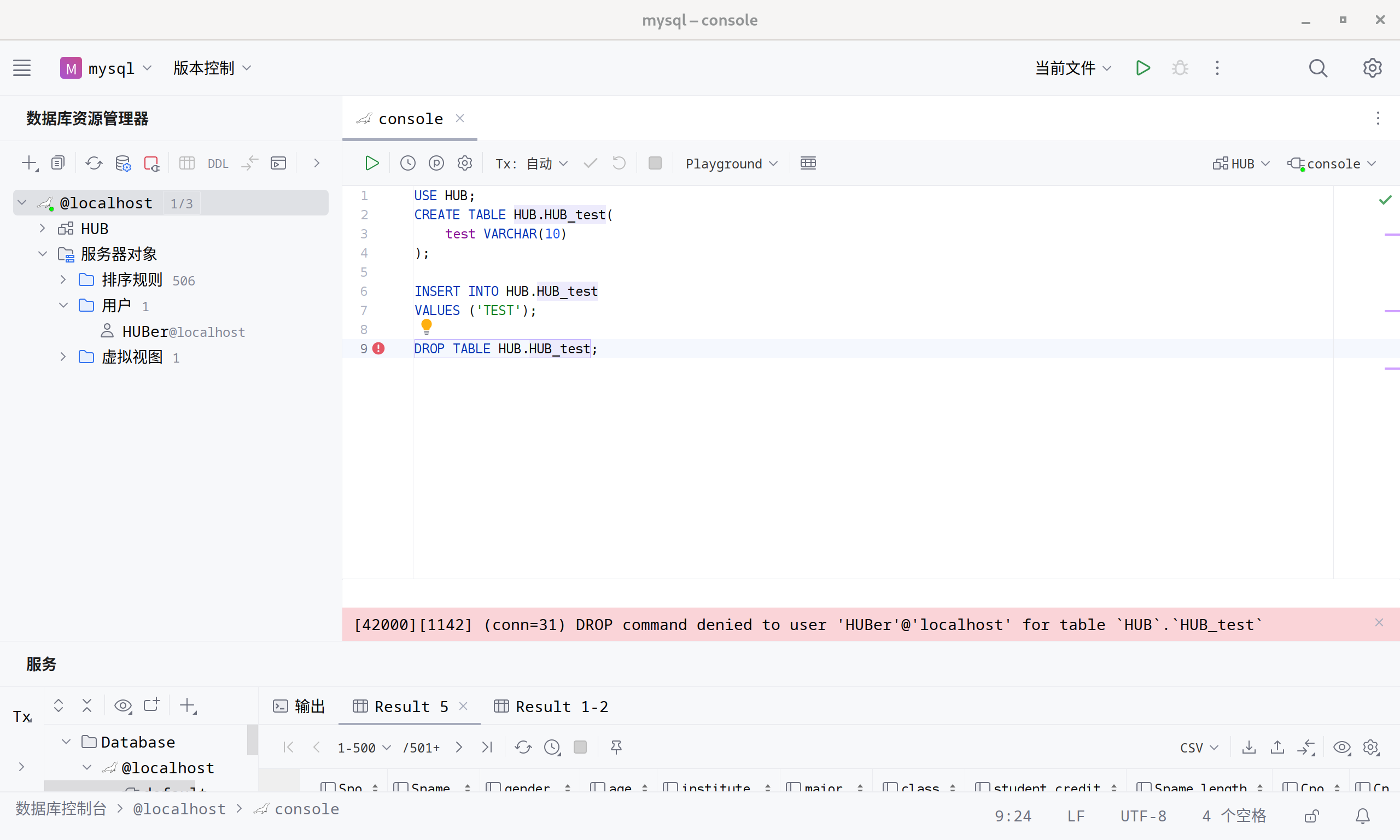


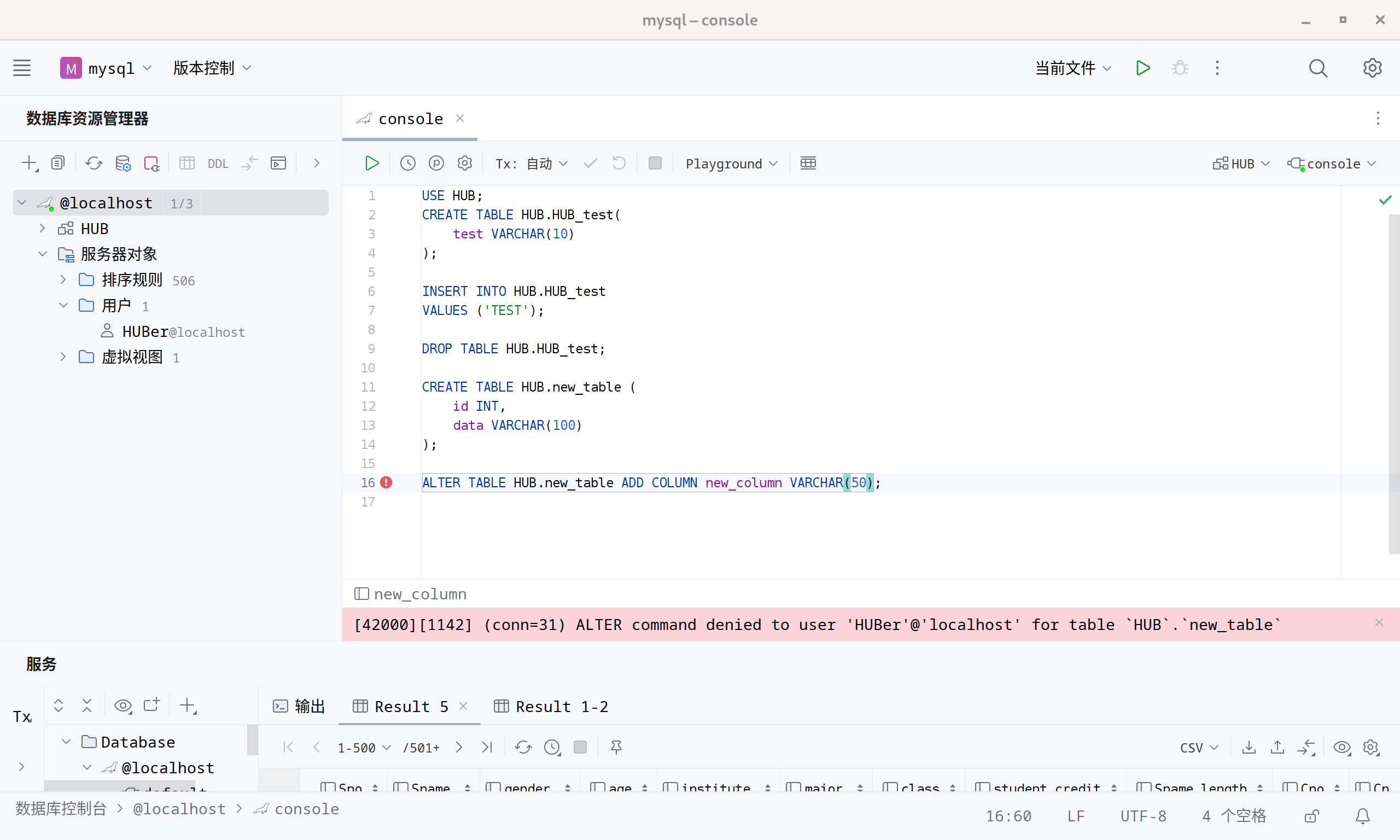
创建：

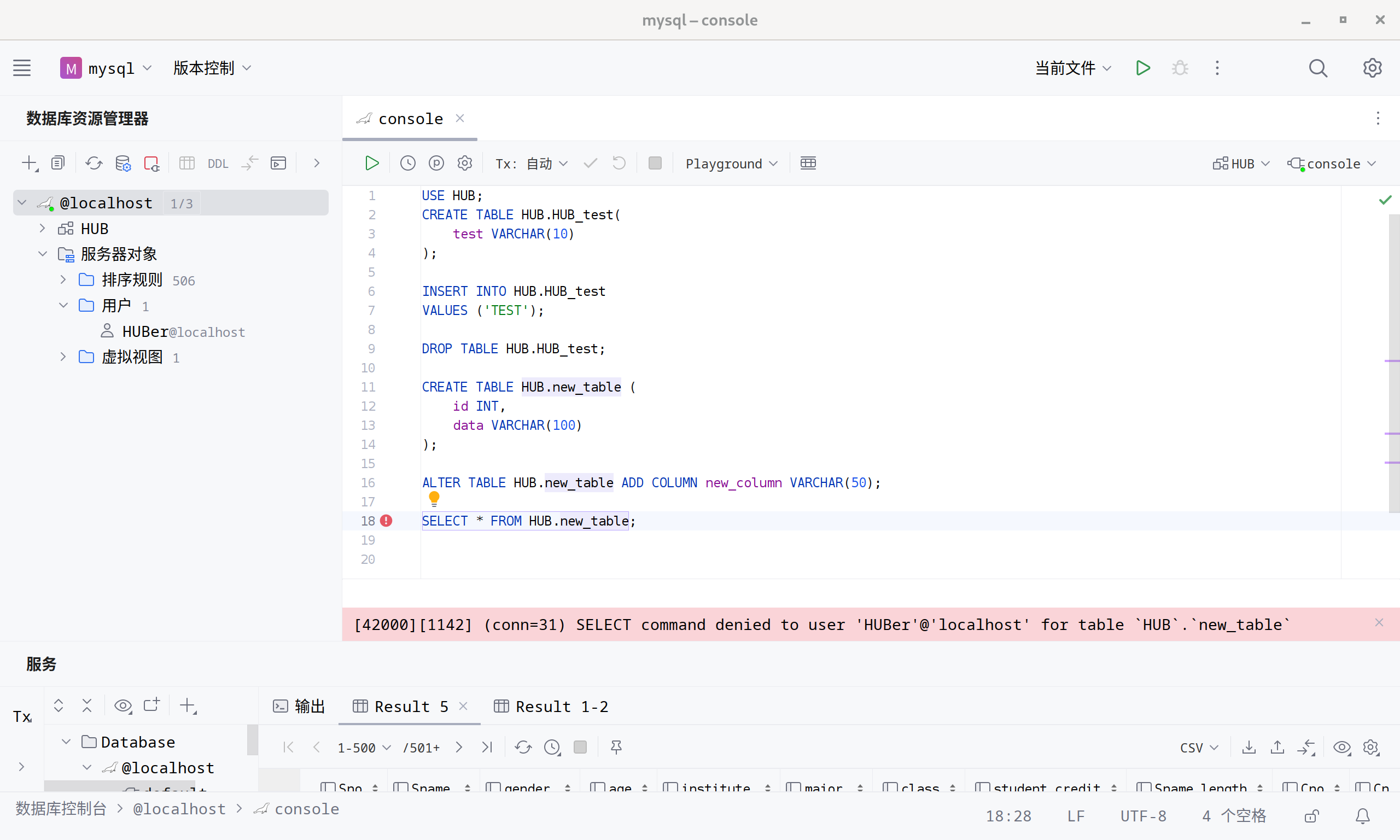


插入：

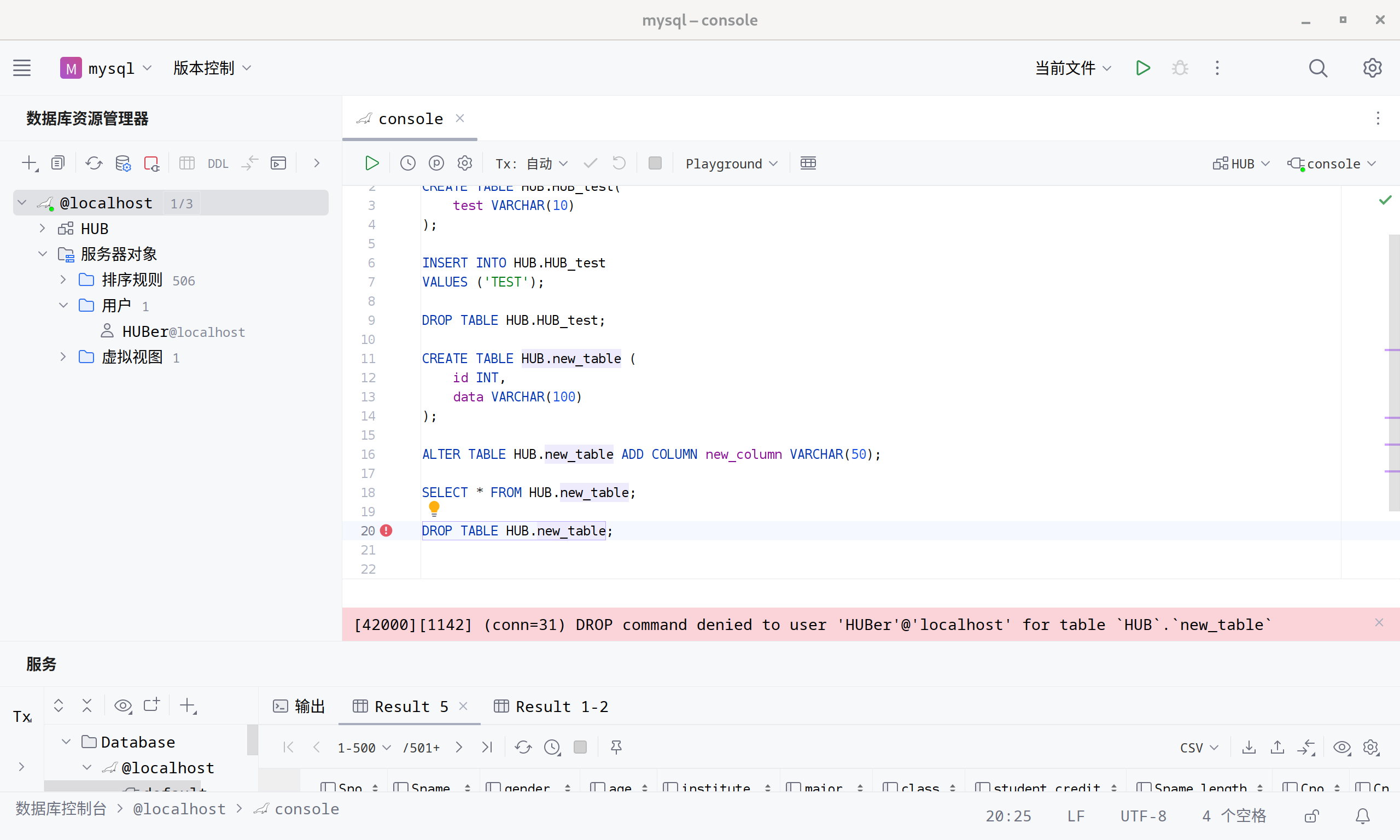


删除：  


更新表：  


查看数据：  


删除表：



**四、思考题**

1、解释10个数据库操作权限的含义和影响。

1. SELECT: 允许用户从表中读取数据。这是最基本的权限，通常授予大多数用户，用于查询和检索数据。
2. INSERT: 允许用户向表中添加新行。这个权限对于需要更新或添加新数据的用户来说很重要，但可能导致数据膨胀或意外插入。
3. UPDATE: 允许用户修改表中现有行的数据。这是一个强大的权限，可以更改数据，但如果不慎使用，可能导致数据损坏。
4. DELETE: 允许用户从表中删除行。这个权限需要谨慎授予，因为它可以永久移除数据。
5. CREATE: 允许用户创建新的数据库对象，如表、视图和存储过程。这个权限通常授予数据库设计者或管理员。
6. DROP: 允许用户删除数据库对象。这是一个高风险权限，因为它可以永久移除表、视图等结构。
7. ALTER: 允许用户修改现有数据库对象的结构，例如添加或删除表中的列。这个权限允许用户调整数据库结构，但也可能导致与现有应用程序的不兼容。
8. GRANT OPTION: 允许用户将他们拥有的权限授予其他用户。这是一种管理权限，需要非常谨慎地使用。
9. INDEX: 允许用户在表上创建或删除索引。索引对于提高查询性能至关重要，但也可能影响插入、更新和删除的性能。
10. EXECUTE: 允许用户执行存储过程或函数。这是对那些需要运行复杂业务逻辑的用户非常重要的权限。

2、试着设计一个例子，分析数据库审计对数据库性能的影响情况。

考虑一个在线零售数据库，该数据库记录用户交易、库存管理和用户行为。为了保障安全性和符合法规要求，企业决定启用数据库审计功能，记录所有关键表的读取、修改和删除操作。

实施审计: 数据库开始记录每一次对关键表的访问和更改，包括用户身份、时间戳、执行的SQL语句等。

性能影响分析:

1. 存储开销增加: 随着审计日志的增长，数据库需要更多的存储空间来存放这些日志。
2. 查询性能下降: 审计操作增加了数据库的工作负载，尤其是在高频更新、插入或删除操作的表上。这可能导致查询响应时间变长。
3. 写入延迟: 每当对数据库进行修改操作时，审计系统都需要记录相关信息，这可能导致写入操作的延迟。
4. 系统资源占用: 审计过程需要CPU和内存资源来处理和记录操作，可能会影响到其他数据库操作的效率。

缓解策略:

1. 选择性审计: 只对最关键的操作进行审计，减少不必要的日志记录。
2. 定期归档: 定期归档旧的审计日志，释放存储空间。
3. 硬件优化: 增加服务器资源，如更快的CPU和更大的内存，以应对额外的负载。

**实验总结：**加深了对数据安全性的理解，并掌握Navicat中有关用户，角色及操作权限的管理方法。掌握数据库审计的设置和管理方法，以便监控数据库操作，维护数据库安全。

**教师评语及成绩**：