

MySQL No Database Error



你提到了一个非常好的问题！我们再仔细看一下分号在**存储过程或事务**中的使用，特别是 `END` 之前或之后。

存储过程和块结构的分号使用

在 MySQL 中，存储过程的结构通常是用 `BEGIN` 和 `END` 来定义块结构。存储过程中的每一条 SQL 语句都必须以分号 (`;`) 结束，但 `END` 本身也可以看作是一个语句的结束。因此：

- 每个独立的 SQL 语句（如 `SELECT`、`INSERT`）需要用分号结束。
- `END` 之前的最后一条 SQL 语句也需要用分号结束，因为它是存储过程的最后一部分。
- `END` 本身不需要分号，但存储过程定义的结尾需要分号（这就是为什么我们要在 `END` 后面加分号来结束整个存储过程的定义）。

举例说明

假设你正在定义一个存储过程：

```
sql

DELIMITER $$

CREATE PROCEDURE example_procedure()
BEGIN
    SELECT 'Hello, World!'; -- 这里需要分号结束SQL语句
    INSERT INTO example_table (column1) VALUES ('value1'); -- 这里也需要分号
END $$ -- 这里的 `END` 标志着块的结束，块结构定义的结束符是 $$，没有加分号

DELIMITER ;
```

在这个例子中：

- `SELECT` 和 `INSERT` 语句分别以分号结束。
- `END` 之前的最后一条 SQL 语句也以分号结束。
- 存储过程的定义是用 `DELIMITER $$` 包裹的，因此 `END` 之后使用了 `$$` 结束整个存储过程定义。
- 在改变回默认分隔符后，任何普通 SQL 语句都会用分号结束。

总结

- 在块结构中，`END` 之前的 SQL 语句需要分号结束，因为每个 SQL 语句必须以分号结束。
- `END` 本身需要有一个分隔符（通常是分号或其他你定义的分隔符）来标志整个存储过程、函数或事务的结束。如果你使用了 `DELIMITER` 来改变默认分隔符，那么在 `END` 后面用那个自定义的分隔符（如 `$$`）结束存储过程定义。如果没有改变 `DELIMITER`，则 `END` 后会加分号。

换句话说，当存储过程的定义完成后，必须有一个分隔符（通常是分号）来标志整个过程的结束。