《数据库原理》大作业

班 级：

学 号：

姓 名：

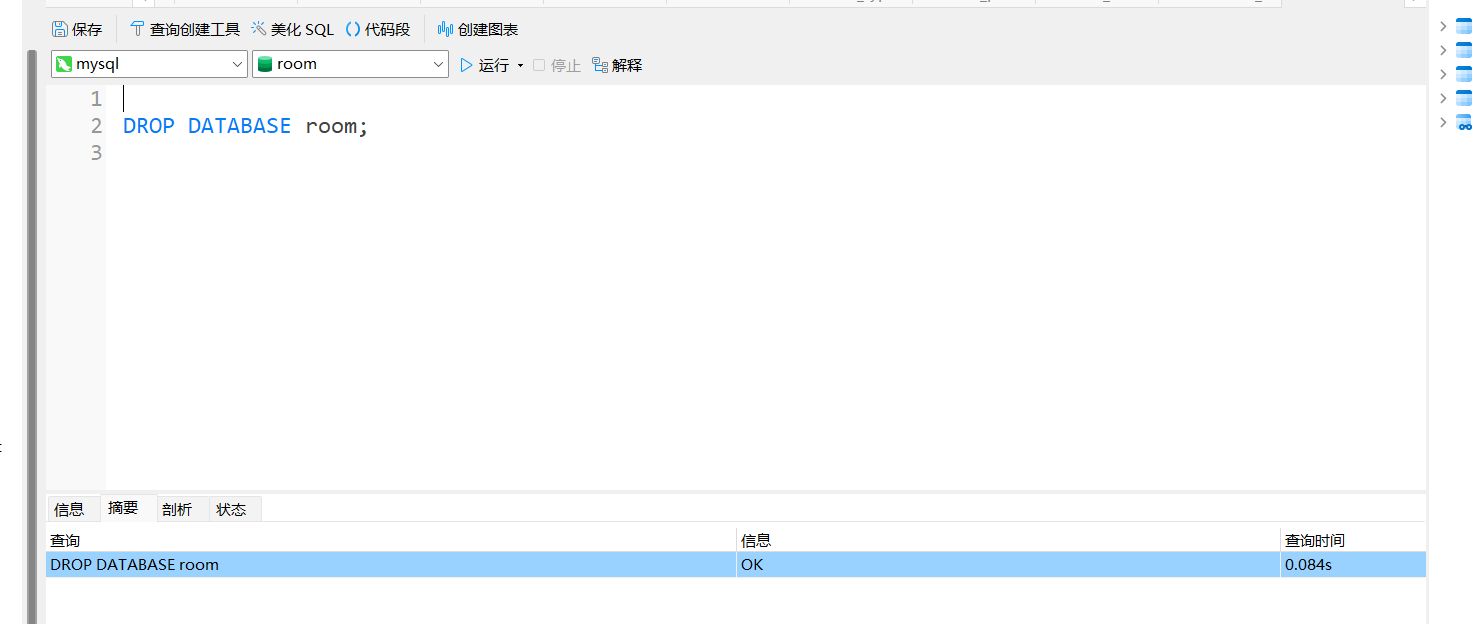
教 师：

学 期：

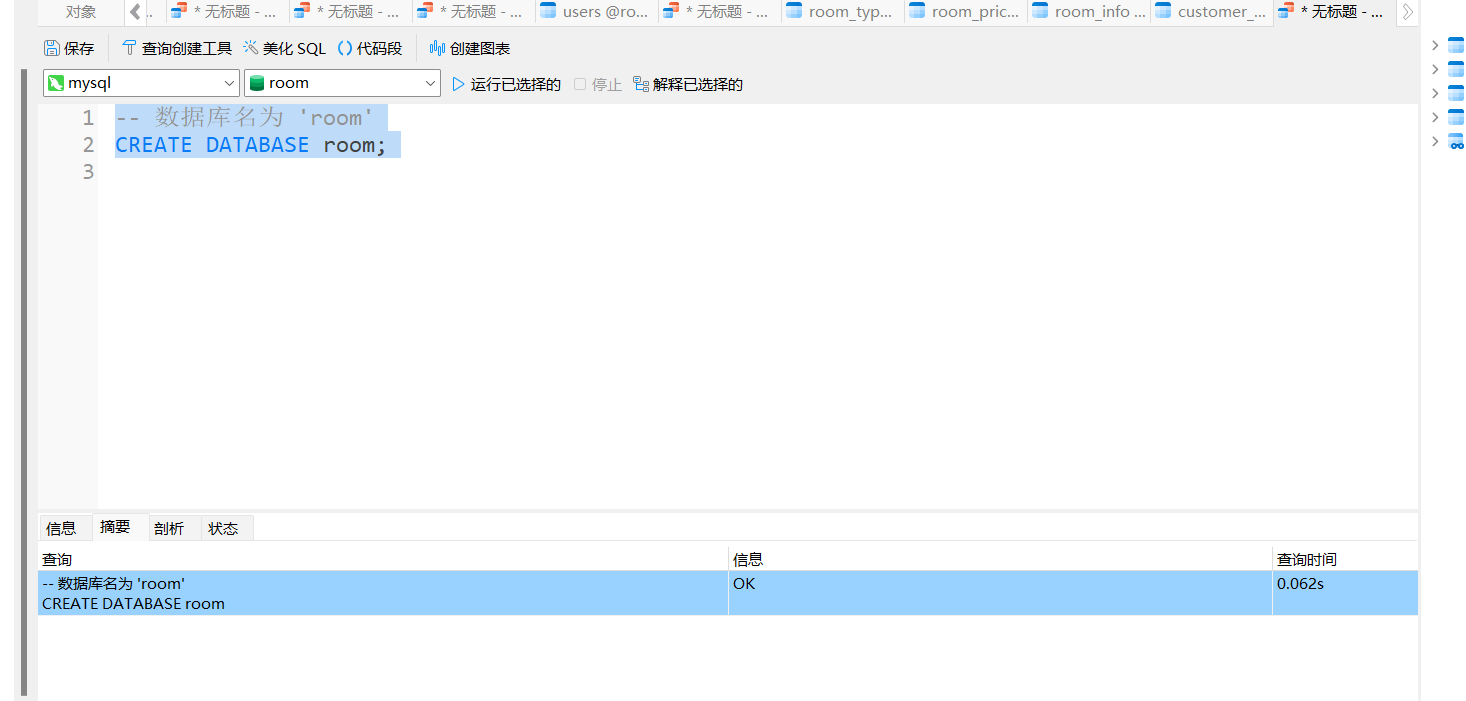
1数据库操作：建立数据库，删除数据库，备份数据库，还原数据库

* 删除数据库

DROP DATABASE room;



建立数据库



-- 备份数据库

mysqldump -u username -p room > backup.sql

--还原数据库

mysql -u username -p room < backup.sql

2 表的操作：建表，修改表，删除表，添加约束（每种约束都要添加）

-- 创建表（根据 `customer\_info` 表结构）

CREATE TABLE customer\_info (

customer\_id INT NOT NULL,

room\_id INT NOT NULL,

check\_in\_date DATE NOT NULL,

check\_out\_date DATE DEFAULT NULL,

total\_cost DECIMAL(10, 2) DEFAULT NULL,

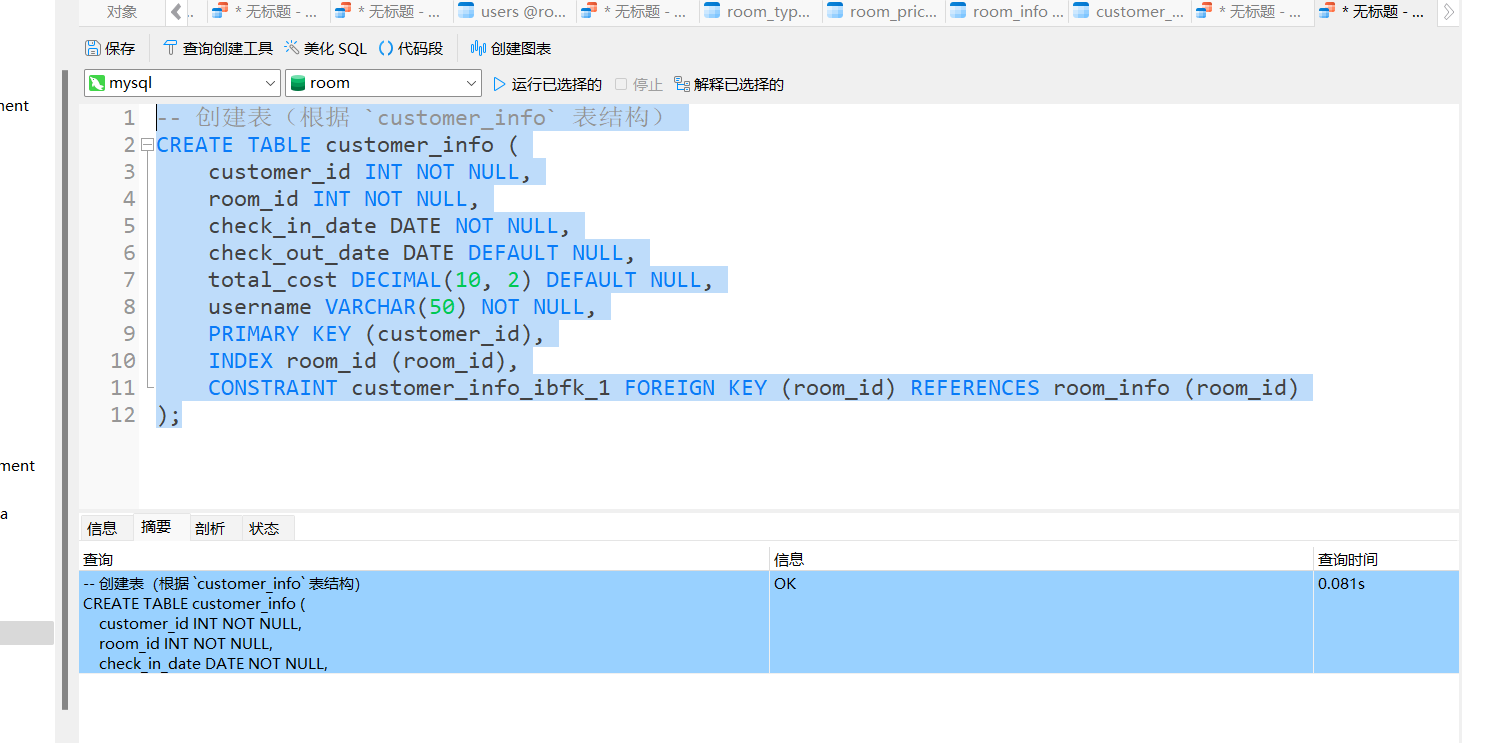
username VARCHAR(50) NOT NULL,

PRIMARY KEY (customer\_id),

INDEX room\_id (room\_id),

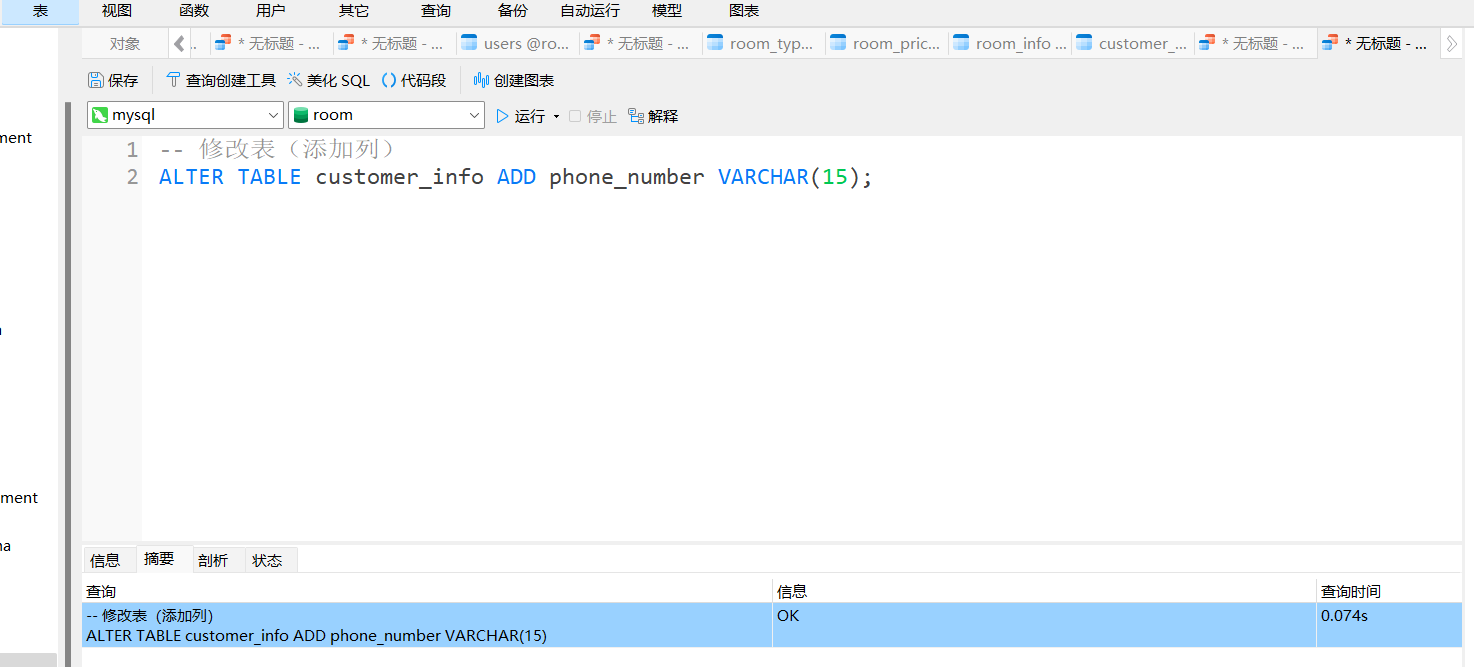
CONSTRAINT customer\_info\_ibfk\_1 FOREIGN KEY (room\_id) REFERENCES room\_info (room\_id)

);



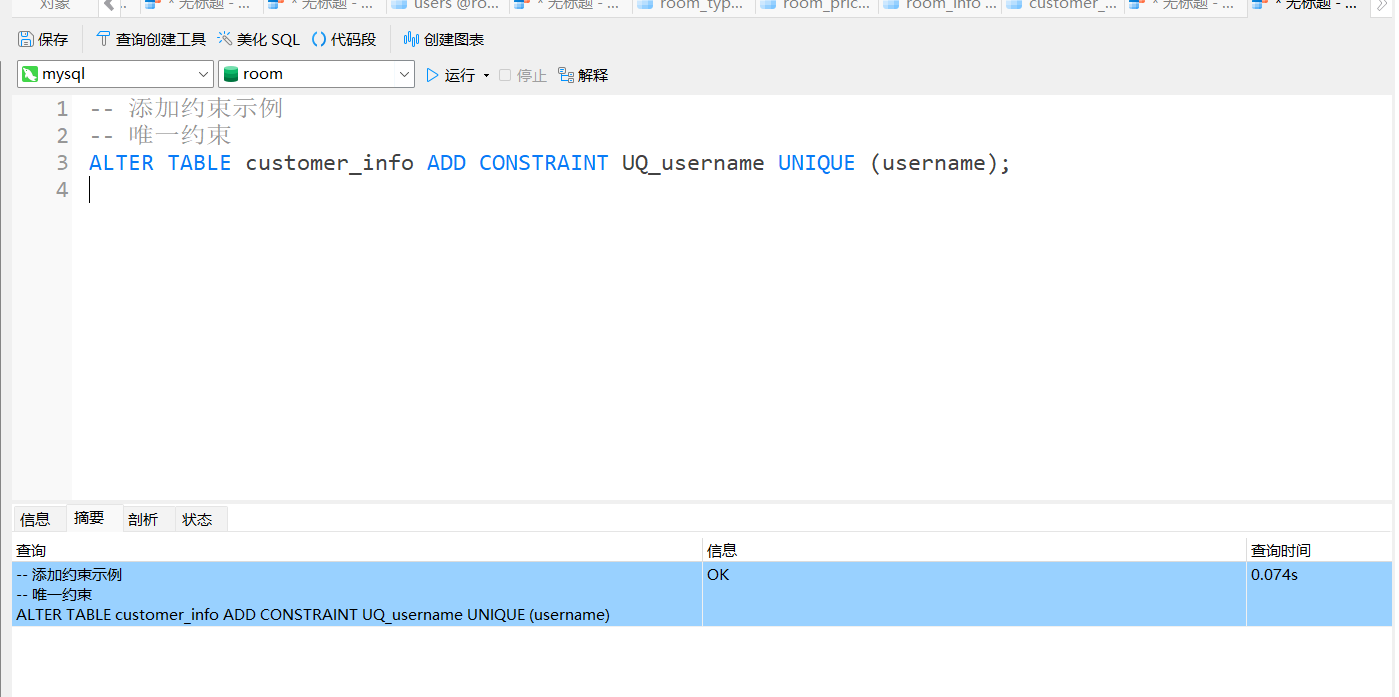
-- 修改表（添加列）

ALTER TABLE customer\_info ADD phone\_number VARCHAR(15);

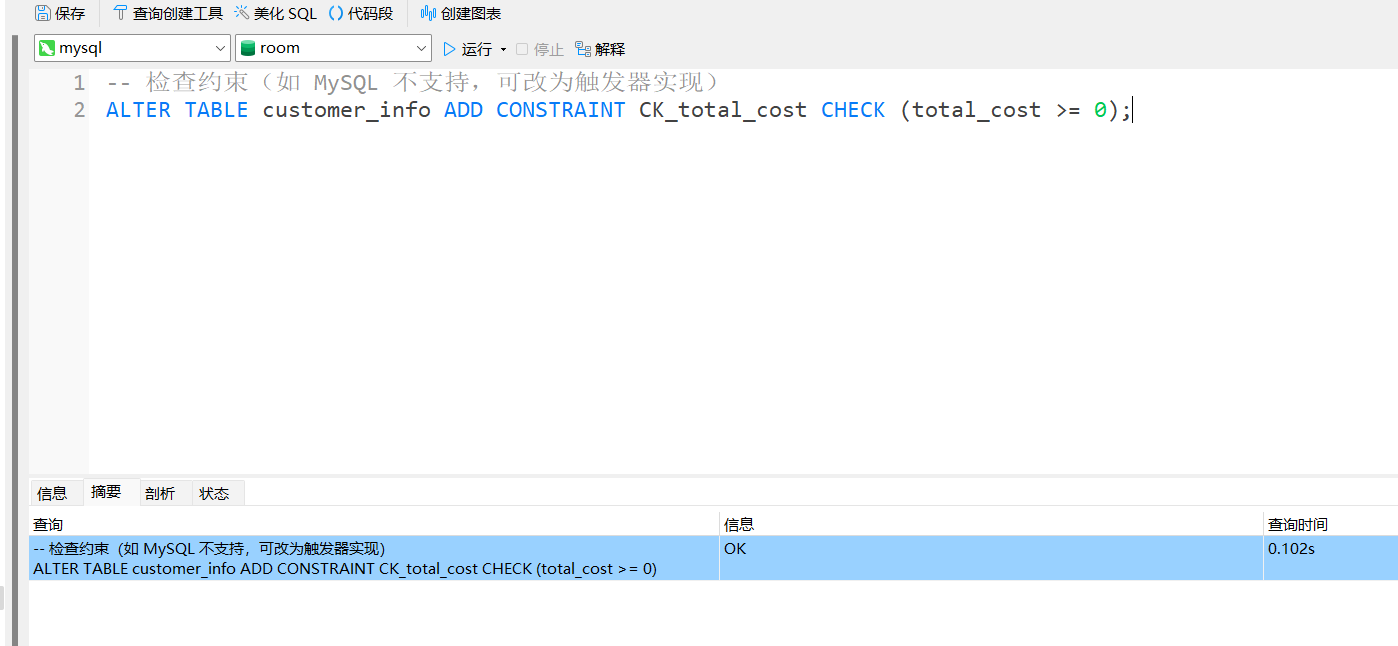


-- 添加约束示例

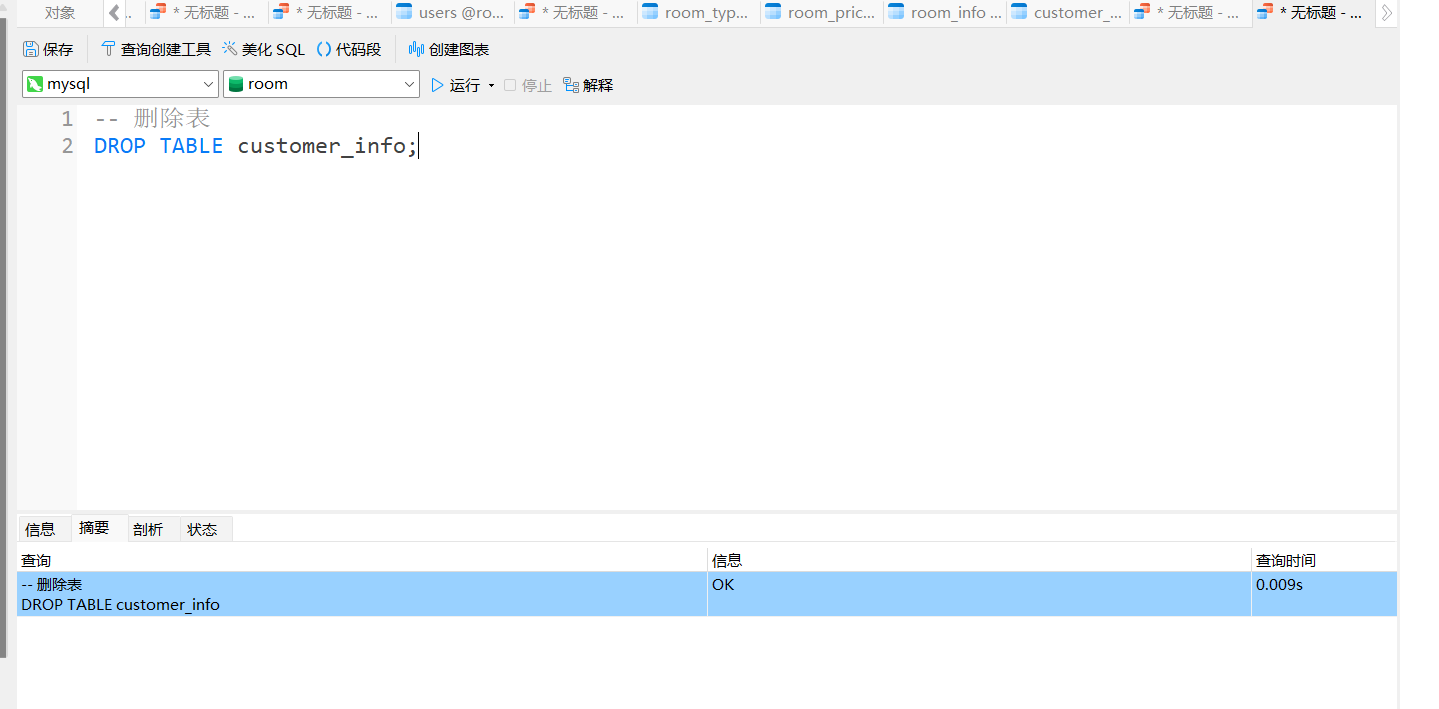
-- 唯一约束

ALTER TABLE customer\_info ADD CONSTRAINT UQ\_username UNIQUE (username); 

-- 检查约束（如 MySQL 不支持，可改为触发器实现） ALTER TABLE customer\_info ADD CONSTRAINT CK\_total\_cost CHECK (total\_cost >= 0);



-- 删除表 DROP TABLE customer\_info;



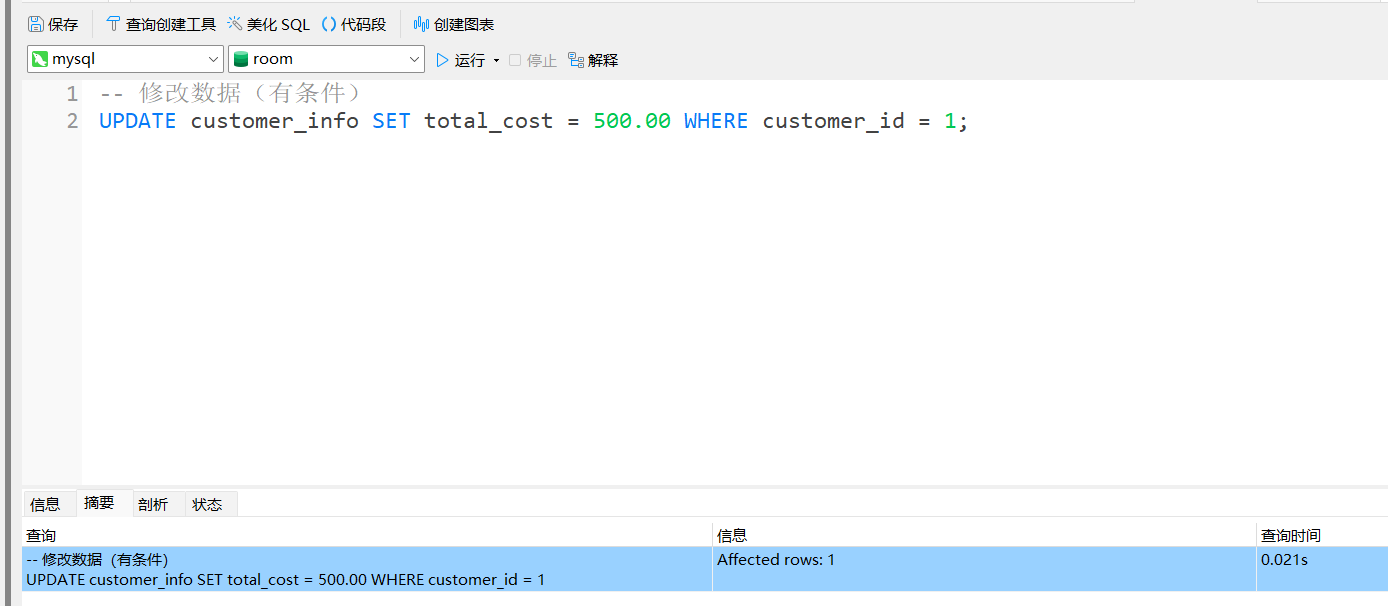
3 数据操作：插入数据，修改数据和删除数据（有条件无条件各一个）

-- 插入数据

INSERT INTO `customer\_info` VALUES (1, 1, '2023-06-21', '2023-06-23', 200.00, '张三');



-- 修改数据（有条件） UPDATE customer\_info SET total\_cost = 500.00 WHERE customer\_id = 1;

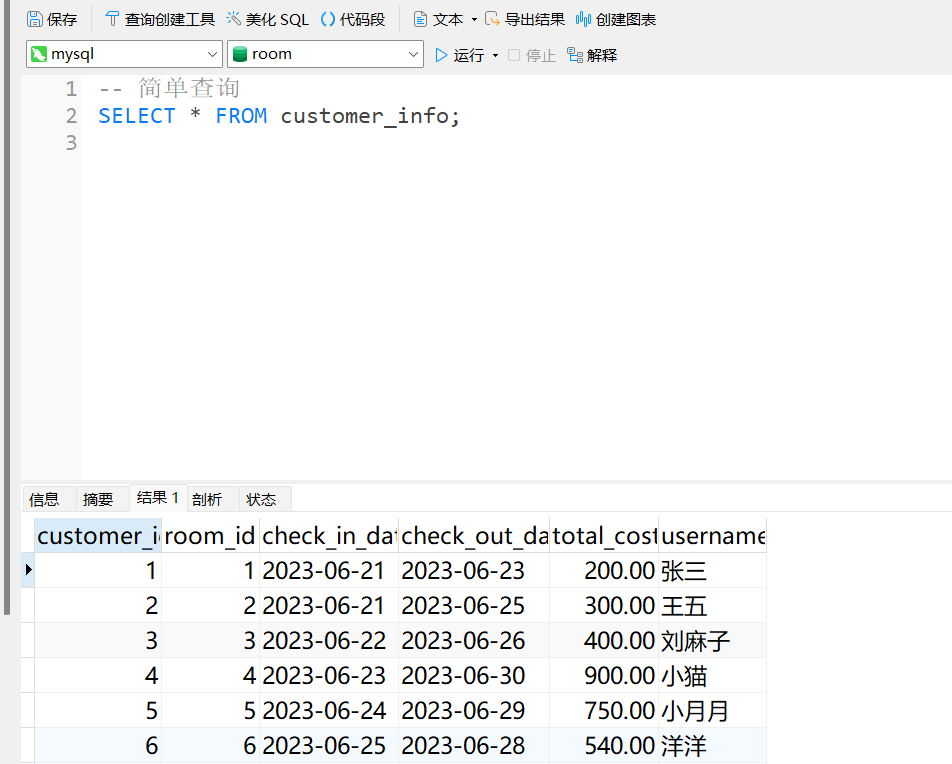


-- 修改数据（无条件） UPDATE customer\_info SET total\_cost = 600.00; -- 删除数据（有条件） DELETE FROM customer\_info WHERE customer\_id = 1; -- 删除数据（无条件） DELETE FROM customer\_info; 

4 数据查询（1）：简单查询，条件查询（每种条件一个查询）

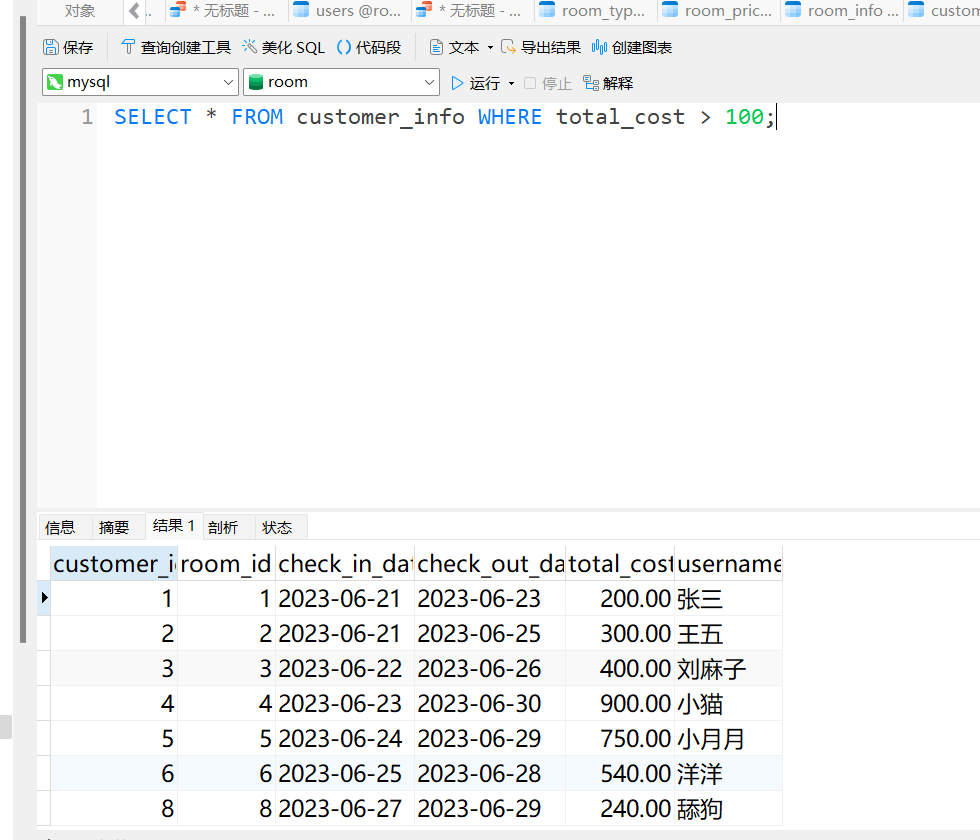
-- 简单查询

SELECT \* FROM customer\_info;



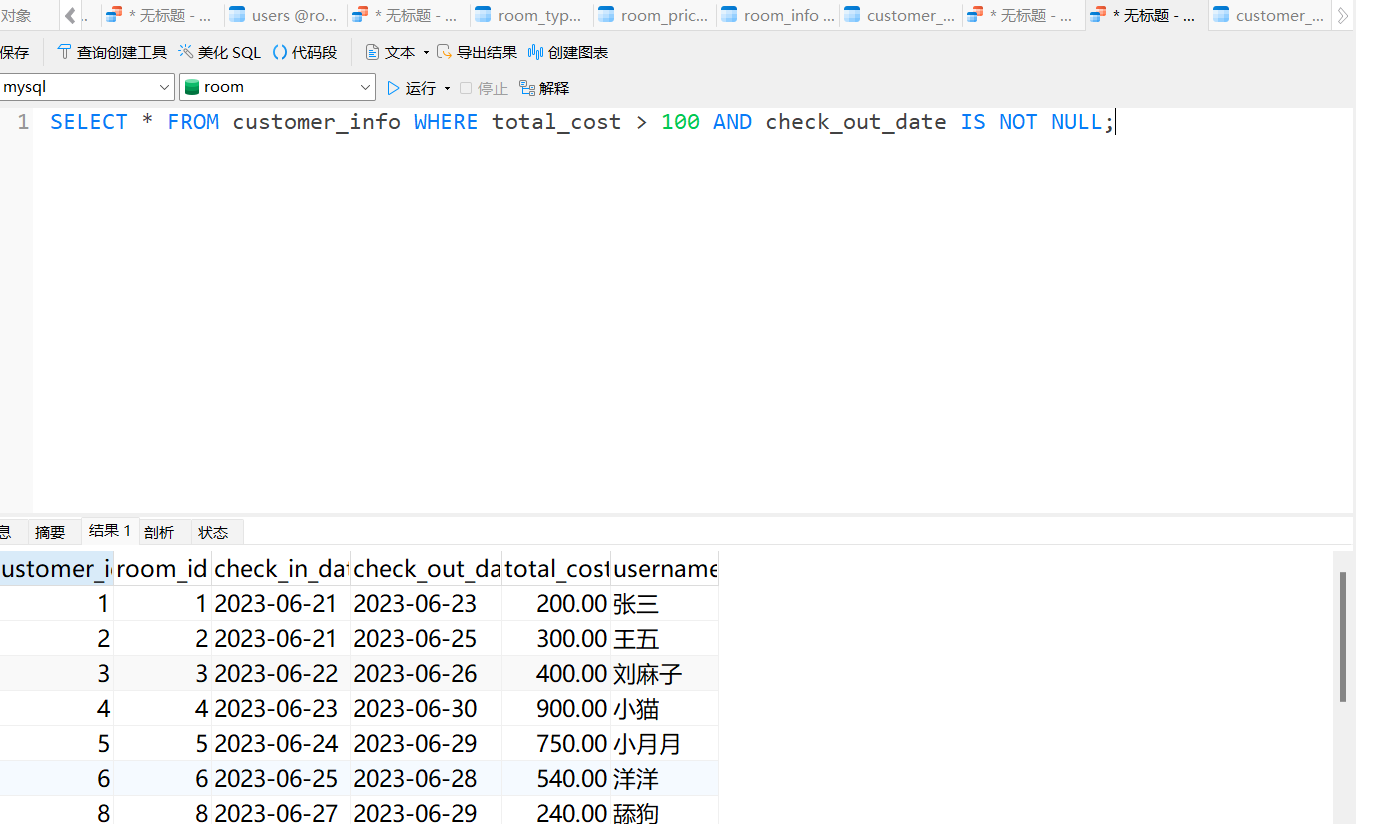
-- 条件查询

-- 比较运算符

SELECT \* FROM customer\_info WHERE total\_cost > 100; 

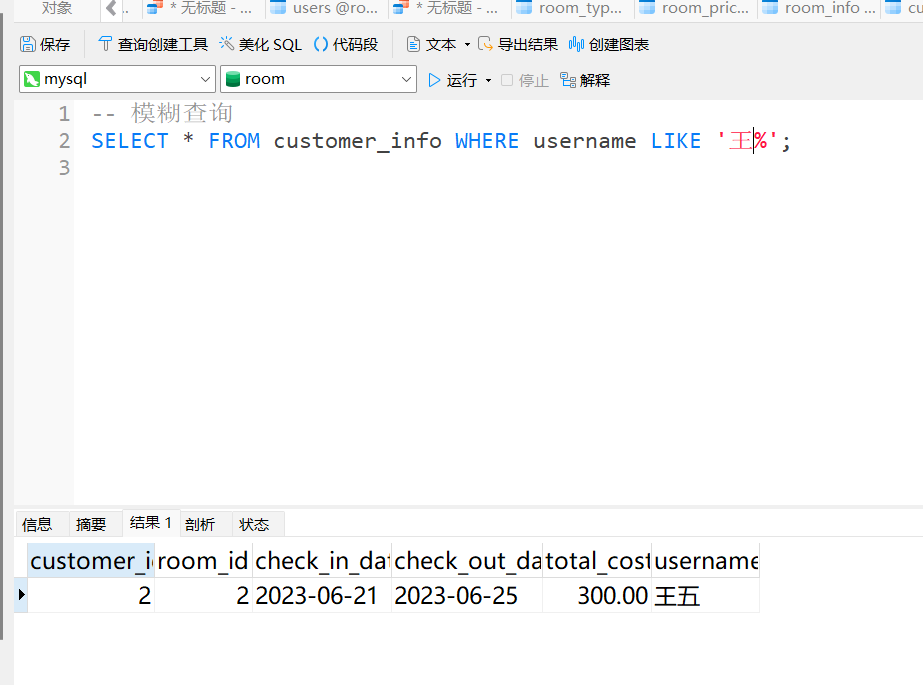
-- 逻辑运算符

SELECT \* FROM customer\_info WHERE total\_cost > 100 AND check\_out\_date IS NOT NULL;



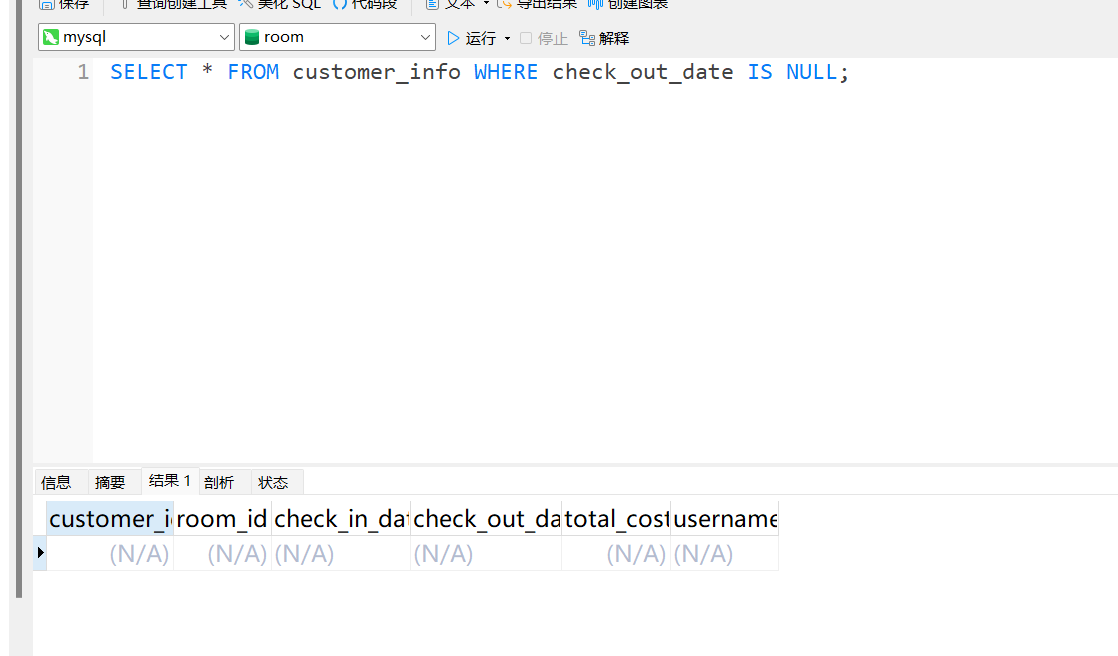
-- 模糊查询

SELECT \* FROM customer\_info WHERE username LIKE 'J%';



-- 空值查询

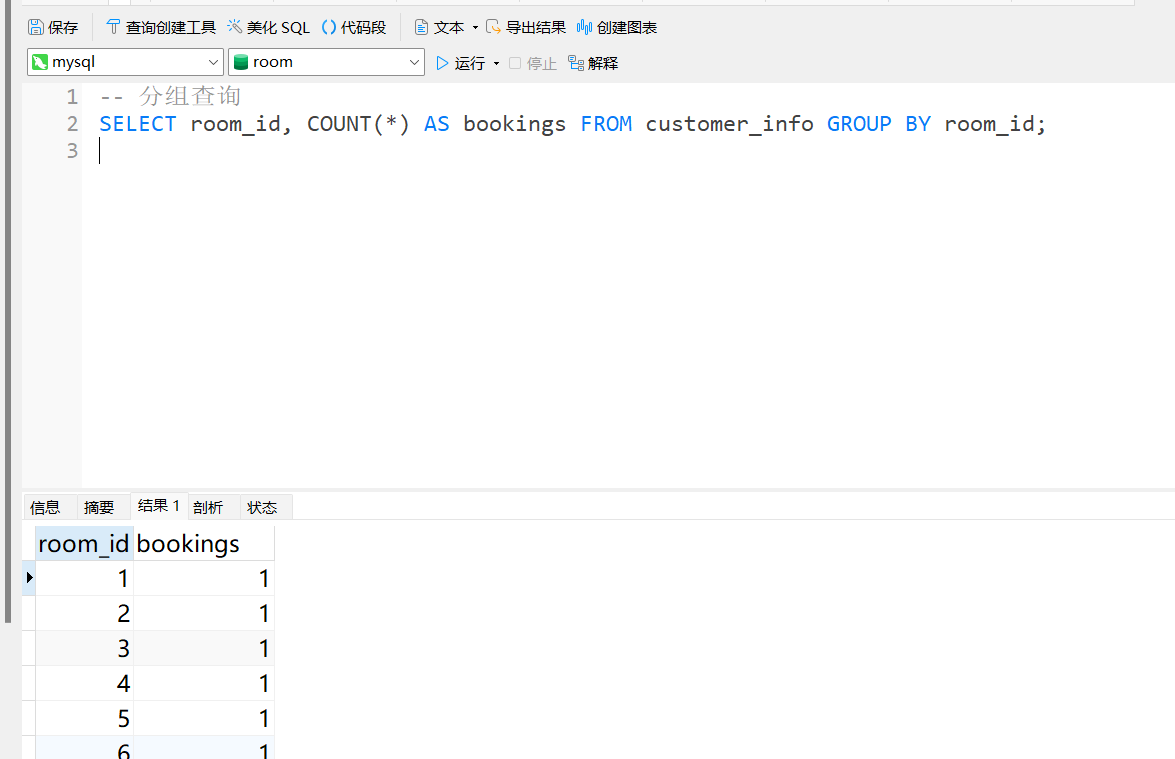
SELECT \* FROM customer\_info WHERE check\_out\_date IS NULL;



5数据查询（2）：高级查询（分组，排序，聚合函数的使用，每种至少一个），多表查询（自然连接，外连接各一个），嵌套查询（in,>any,>all 各一个）

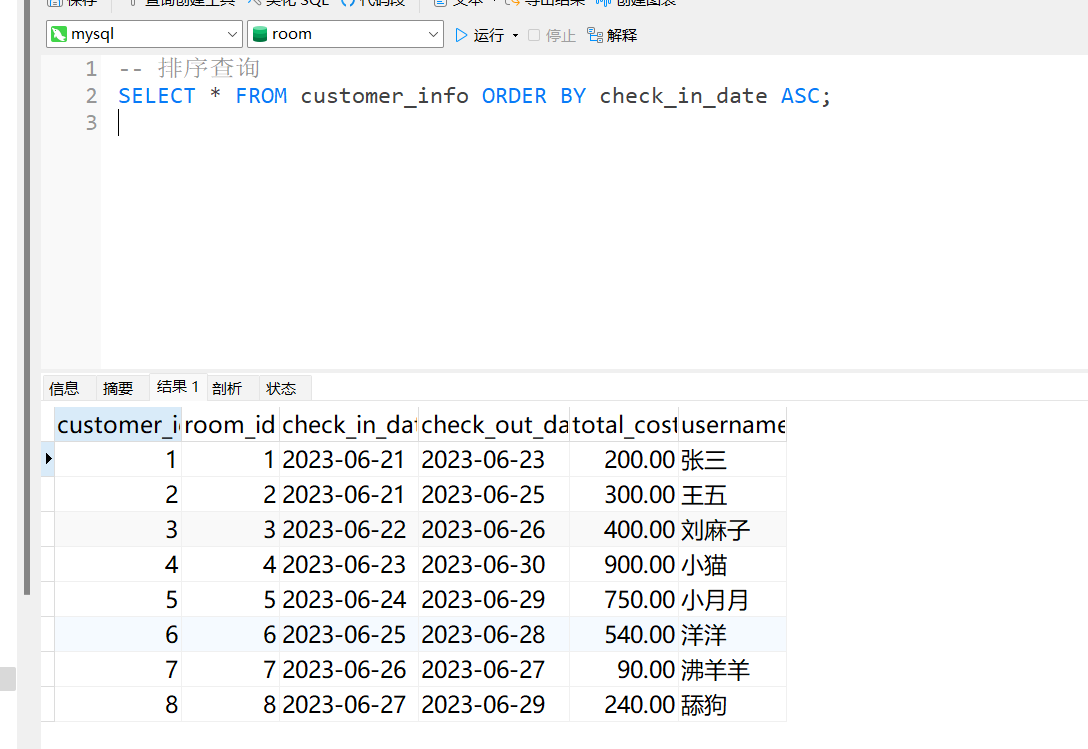
-- 分组查询

SELECT room\_id, COUNT(\*) AS bookings FROM customer\_info GROUP BY room\_id;



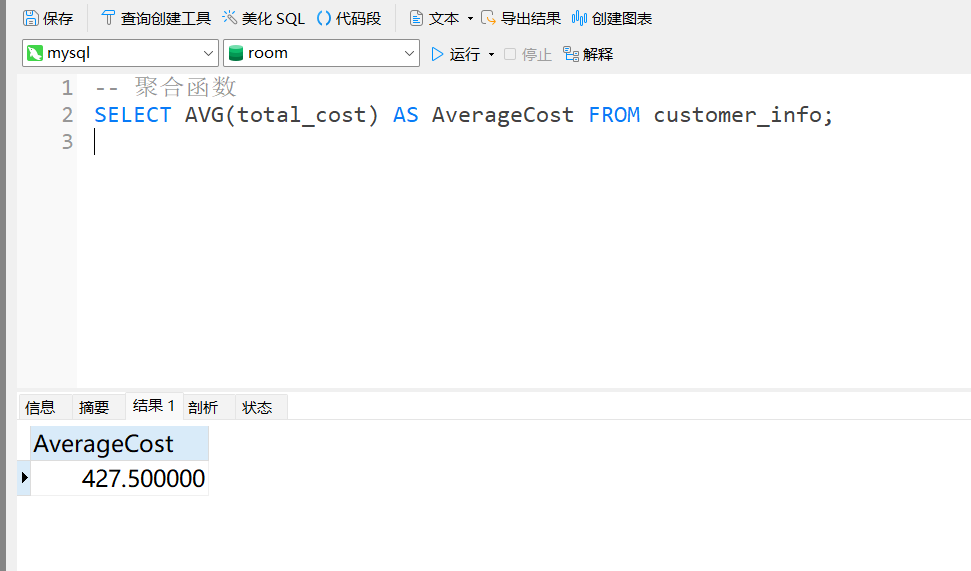
-- 排序查询

SELECT \* FROM customer\_info ORDER BY check\_in\_date ASC;



-- 聚合函数

SELECT AVG(total\_cost) AS AverageCost FROM customer\_info;

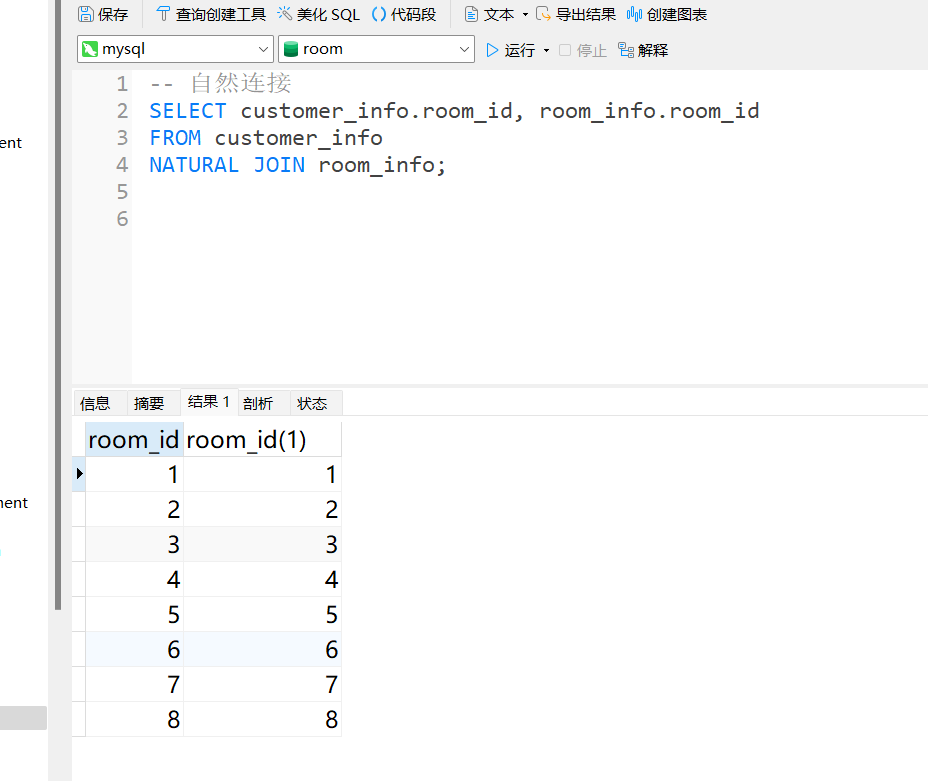


-- 自然连接

SELECT customer\_info.room\_id, room\_info.room\_id

FROM customer\_info

NATURAL JOIN room\_info;

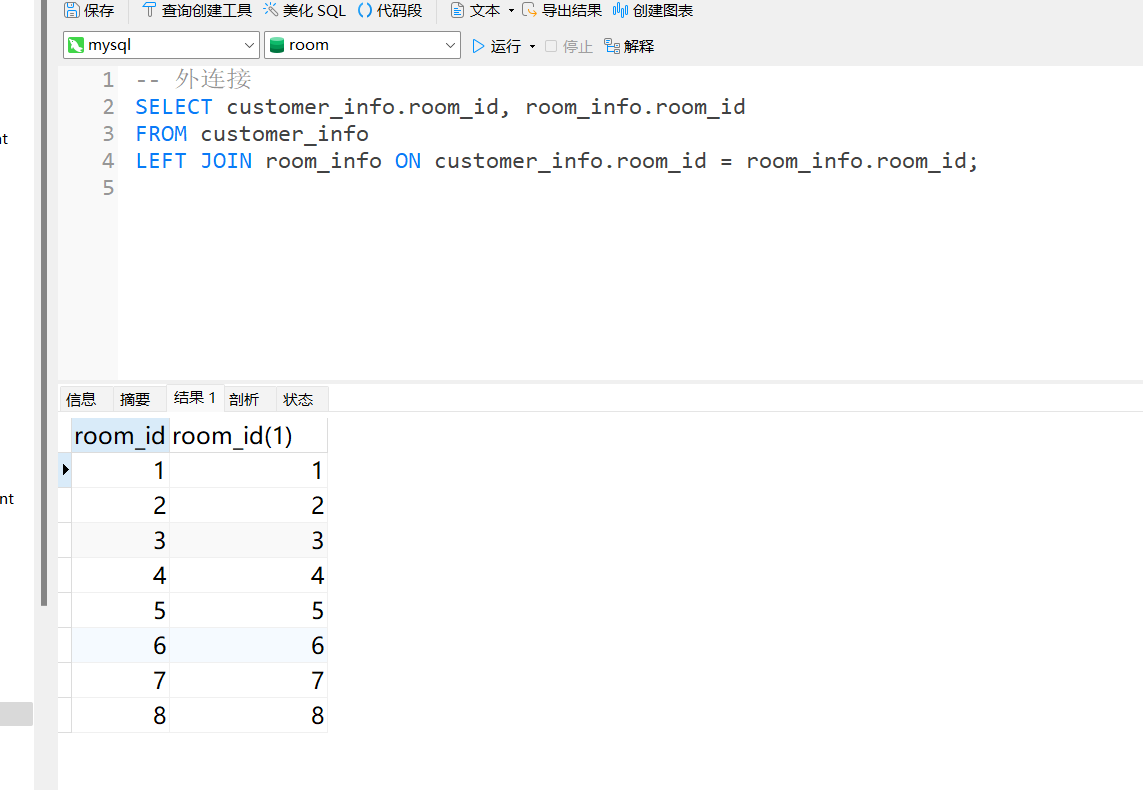
s

-- 外连接

SELECT customer\_info.room\_id, room\_info.room\_id

FROM customer\_info

LEFT JOIN room\_info ON customer\_info.room\_id = room\_info.room\_id;



-- 嵌套查询

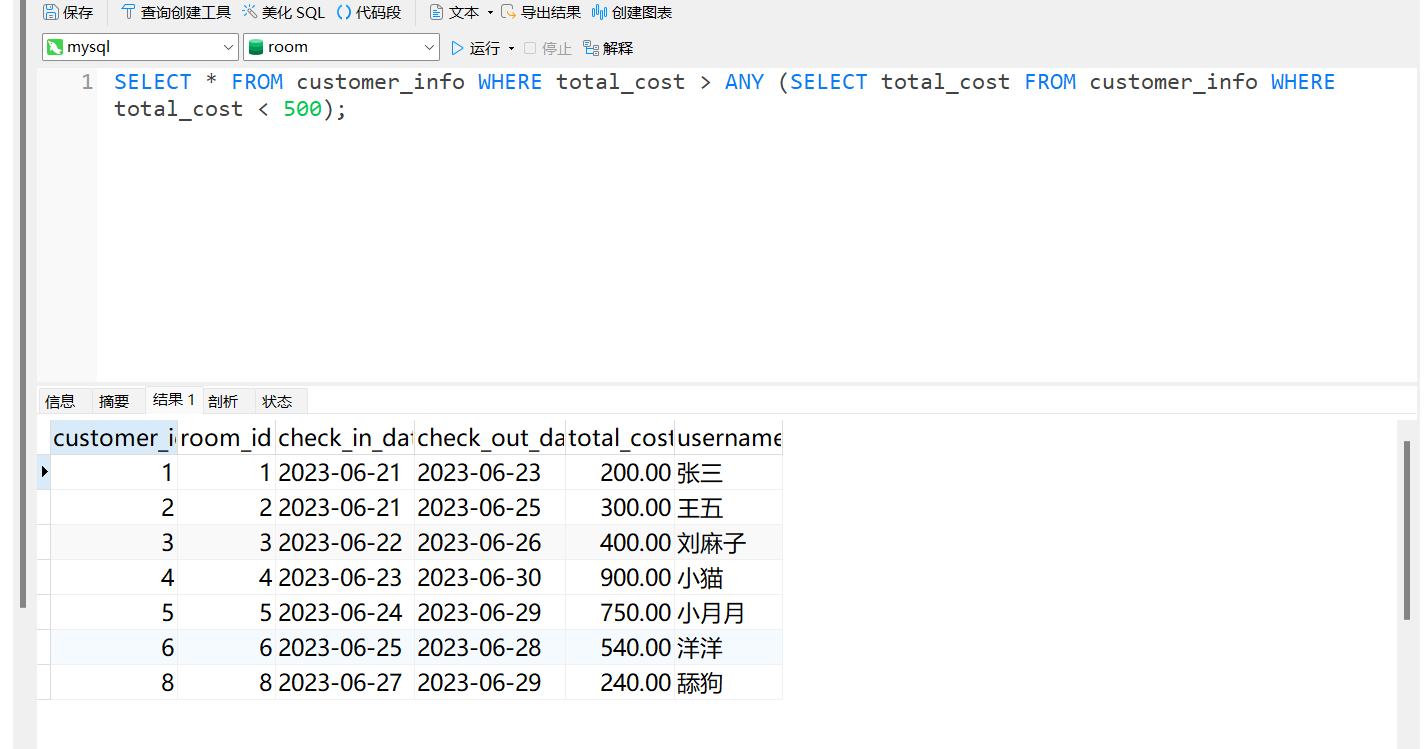
-- IN

SELECT \* FROM customer\_info WHERE room\_id IN (SELECT room\_id FROM room\_info WHERE room\_id = 1);



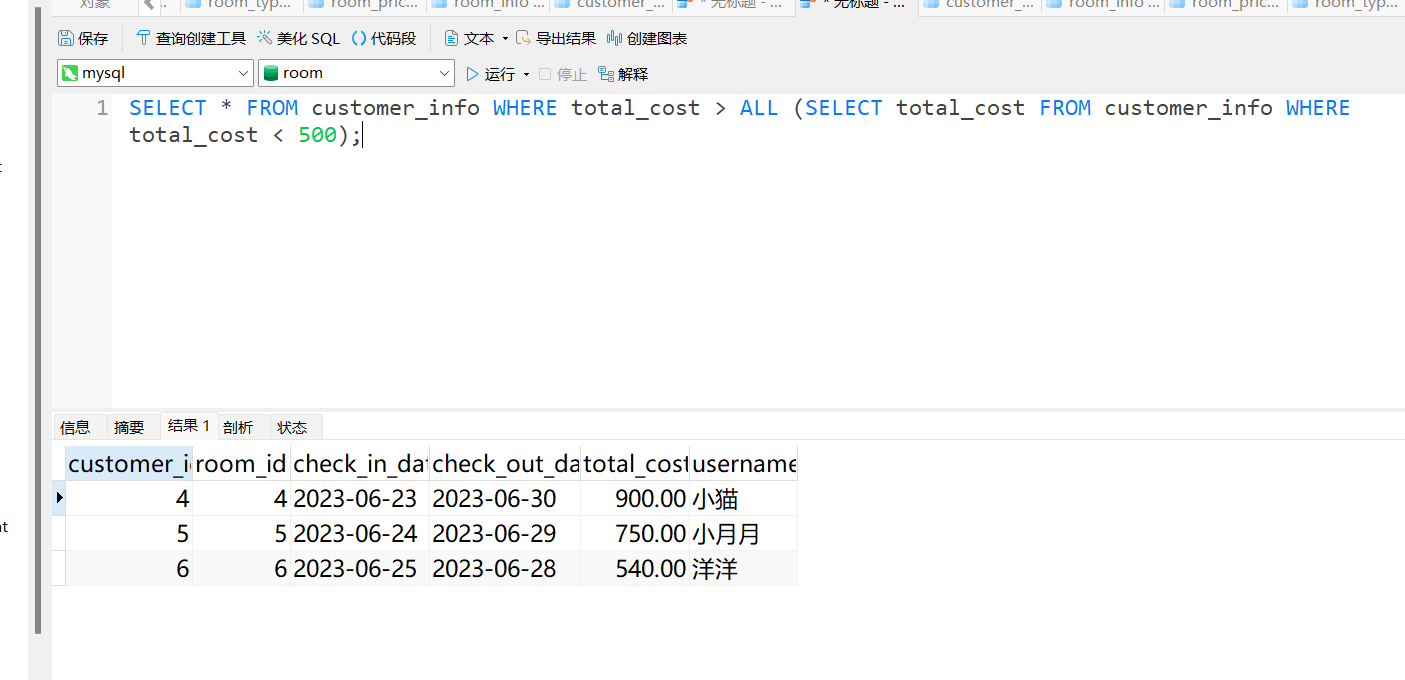
-- >ANY

SELECT \* FROM customer\_info WHERE total\_cost > ANY (SELECT total\_cost FROM customer\_info WHERE total\_cost < 500);



-- >ALL

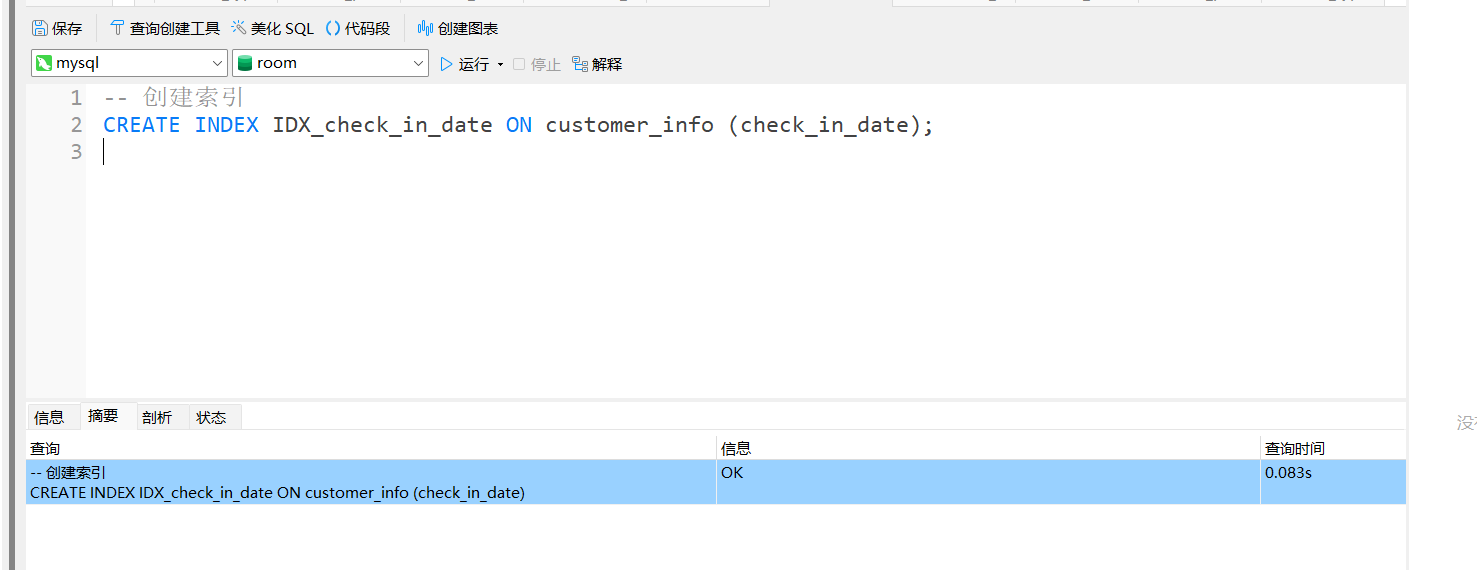
SELECT \* FROM customer\_info WHERE total\_cost > ALL (SELECT total\_cost FROM customer\_info WHERE total\_cost < 500);



6索引操作：建立和删除索引

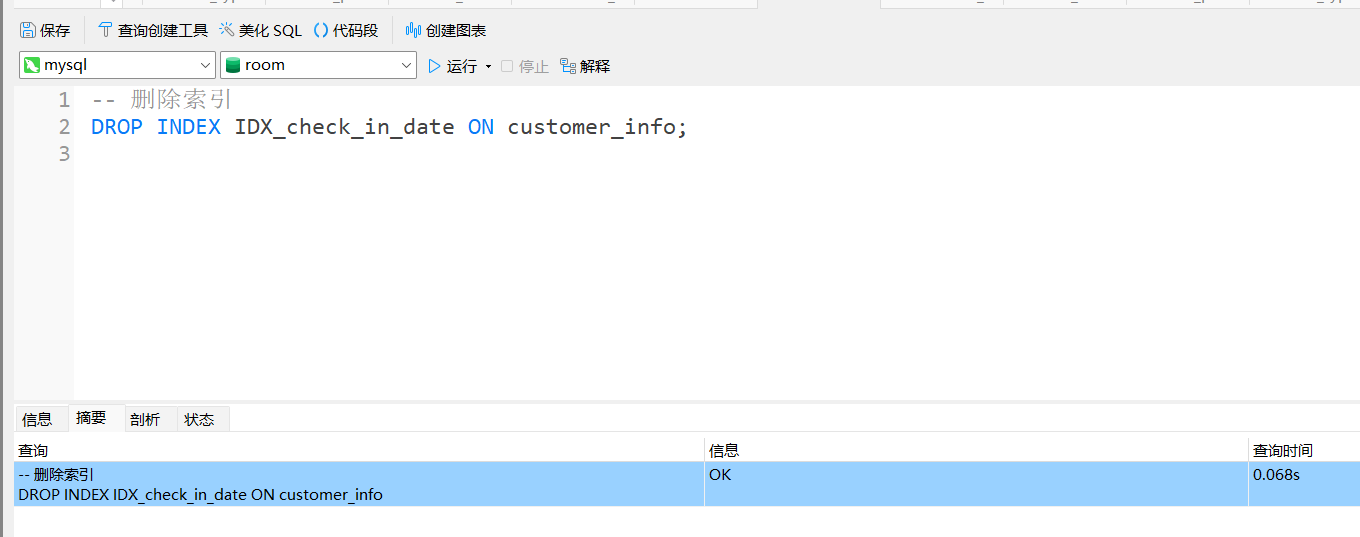
-- 创建索引

CREATE INDEX IDX\_check\_in\_date ON customer\_info (check\_in\_date);



-- 删除索引

DROP INDEX IDX\_check\_in\_date ON customer\_info;



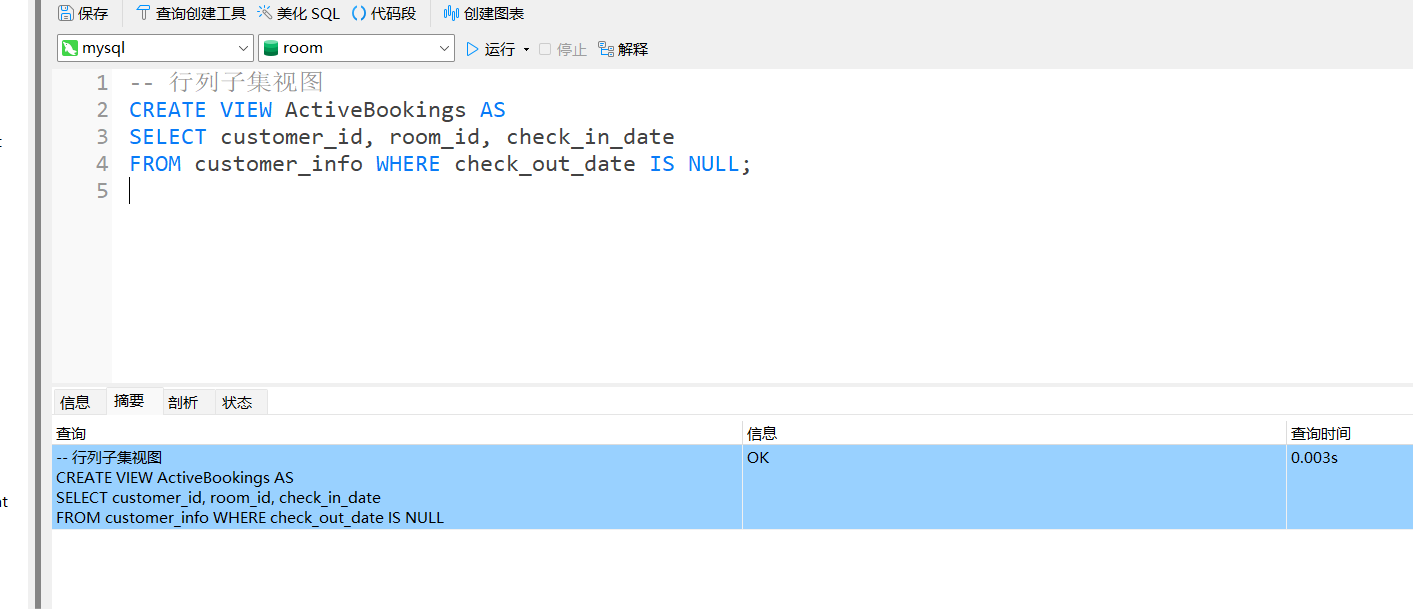
7 视图操作：视图的建立（行列子集一个，非行列子集一个），查询和删除

-- 行列子集视图

CREATE VIEW ActiveBookings AS

SELECT customer\_id, room\_id, check\_in\_date

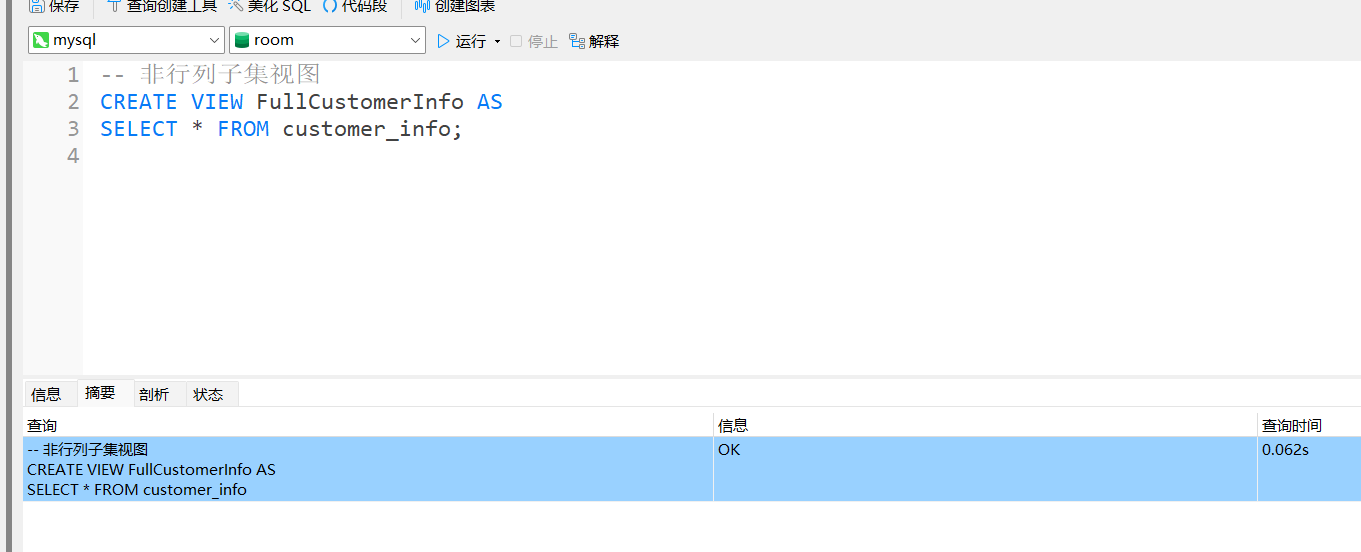
FROM customer\_info WHERE check\_out\_date IS NULL;



-- 非行列子集视图

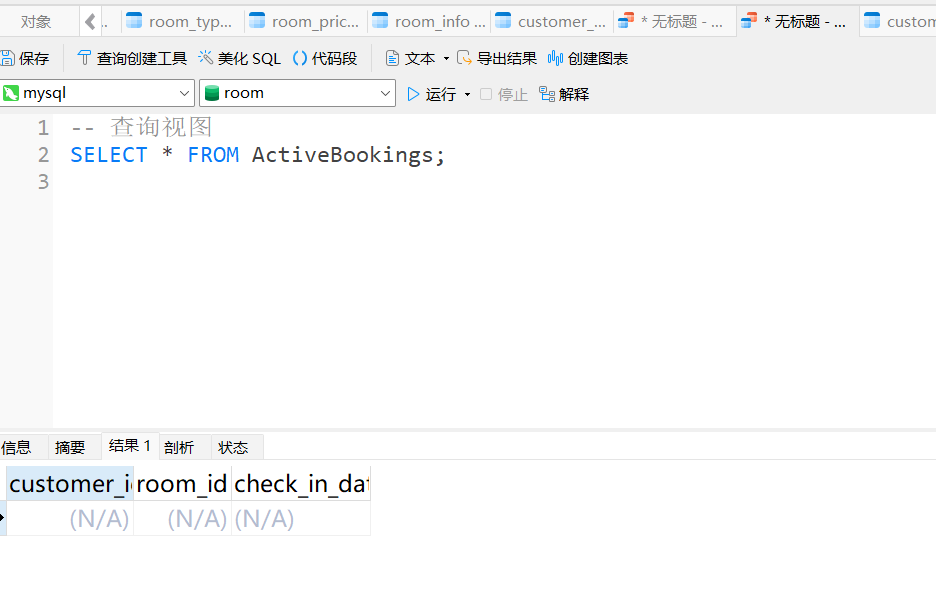
CREATE VIEW FullCustomerInfo AS

SELECT \* FROM customer\_info;



-- 查询视图

SELECT \* FROM ActiveBookings;



-- 删除视图

DROP VIEW ActiveBookings;



8存储过程操作：存储过程的建立和调用（有参无参各一个）和删除

-- 创建存储过程（无参）

CREATE PROCEDURE GetAllCustomers()

BEGIN

SELECT \* FROM customer\_info;

END;



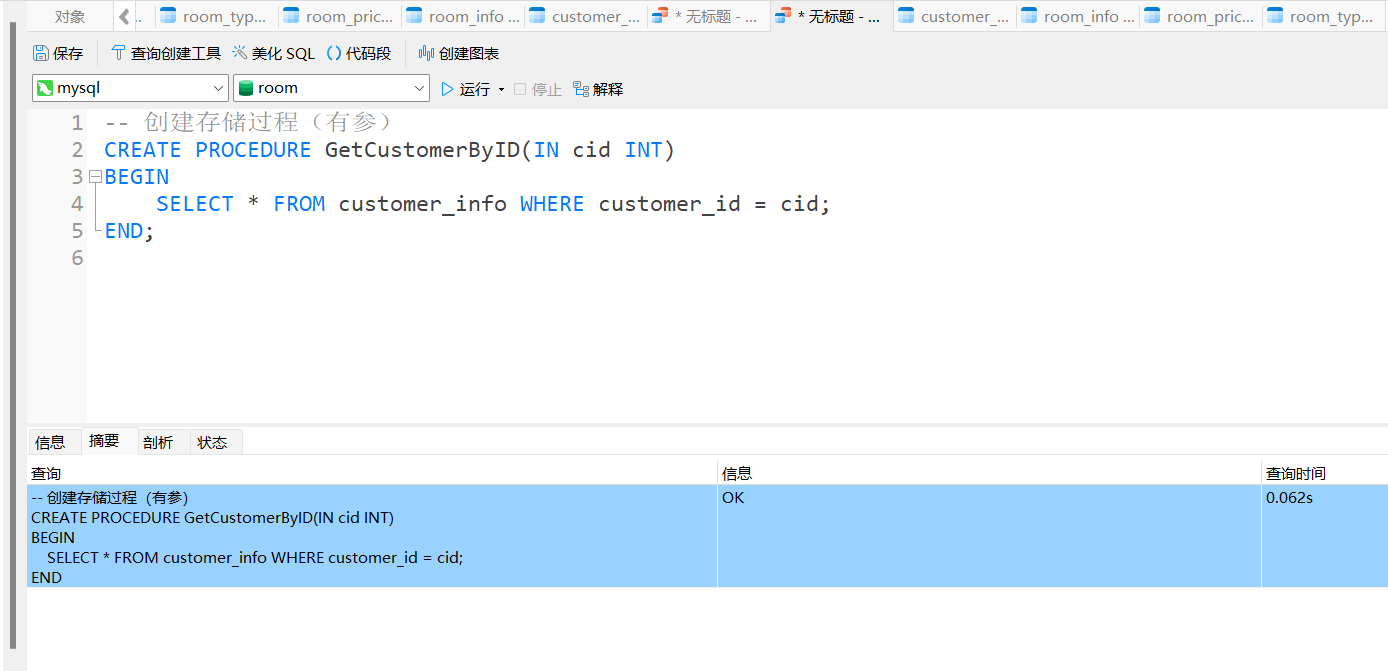
-- 创建存储过程（有参）

CREATE PROCEDURE GetCustomerByID(IN cid INT)

BEGIN

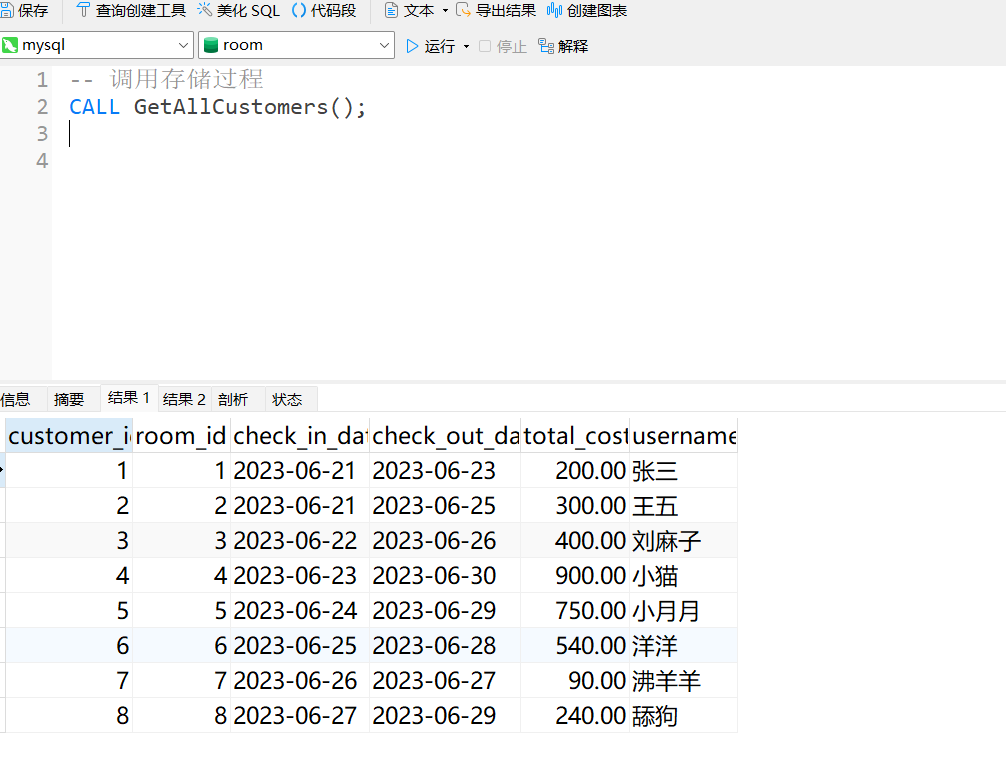
SELECT \* FROM customer\_info WHERE customer\_id = cid;

END;



-- 调用存储过程

CALL GetAllCustomers();



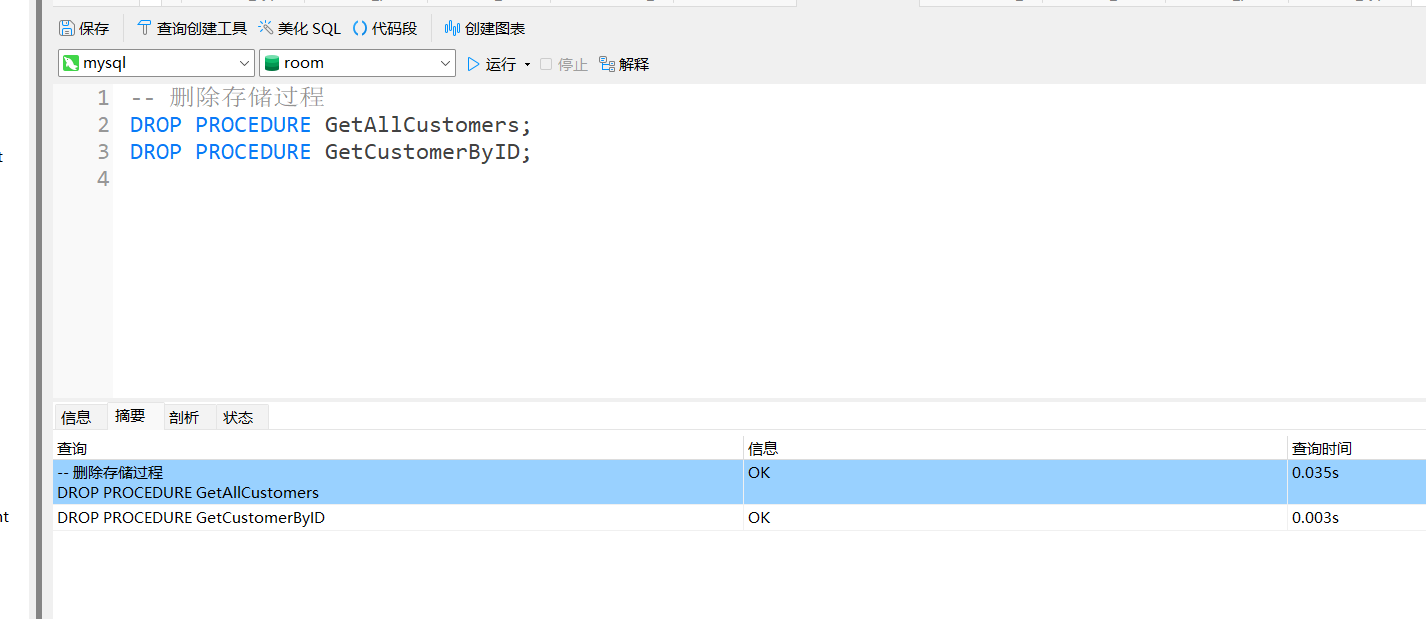
CALL GetCustomerByID(1);



-- 删除存储过程

DROP PROCEDURE GetAllCustomers;

DROP PROCEDURE GetCustomerByID;



9 触发器的操作：建立（insert，update，delete事件各写一个）和删除

-- 插入触发器

CREATE TRIGGER AfterCustomerInsert

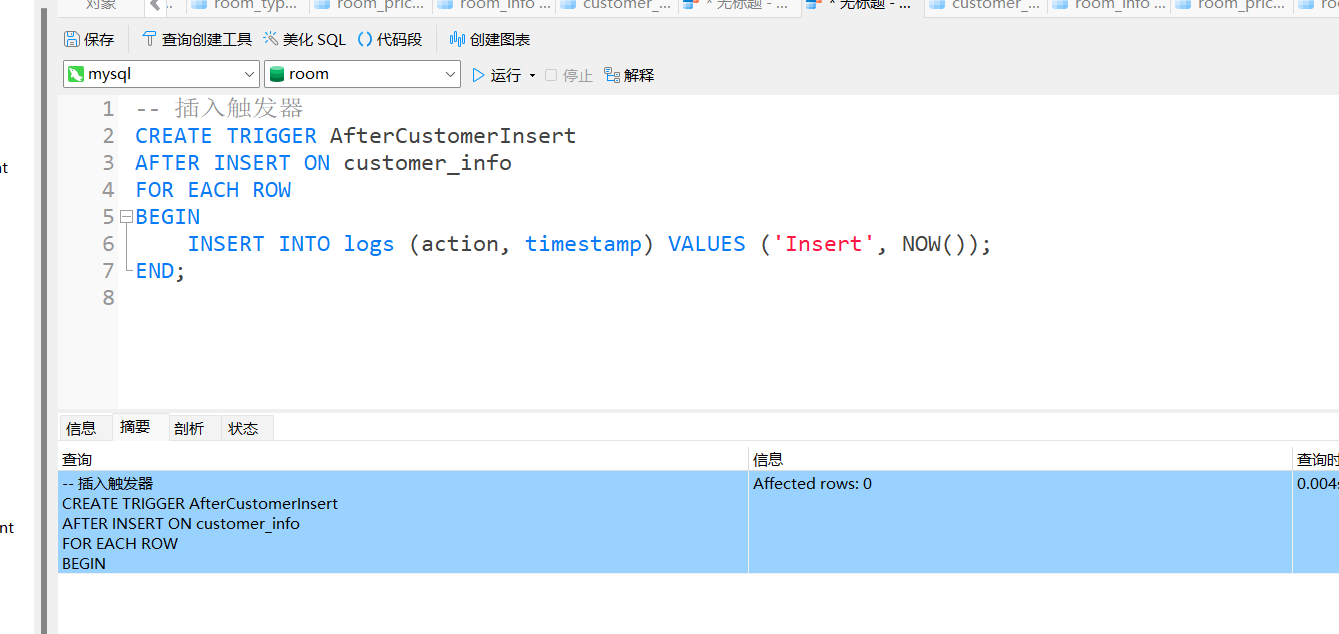
AFTER INSERT ON customer\_info

FOR EACH ROW

BEGIN

INSERT INTO logs (action, timestamp) VALUES ('Insert', NOW());

END;



-- 更新触发器

CREATE TRIGGER BeforeCustomerUpdate

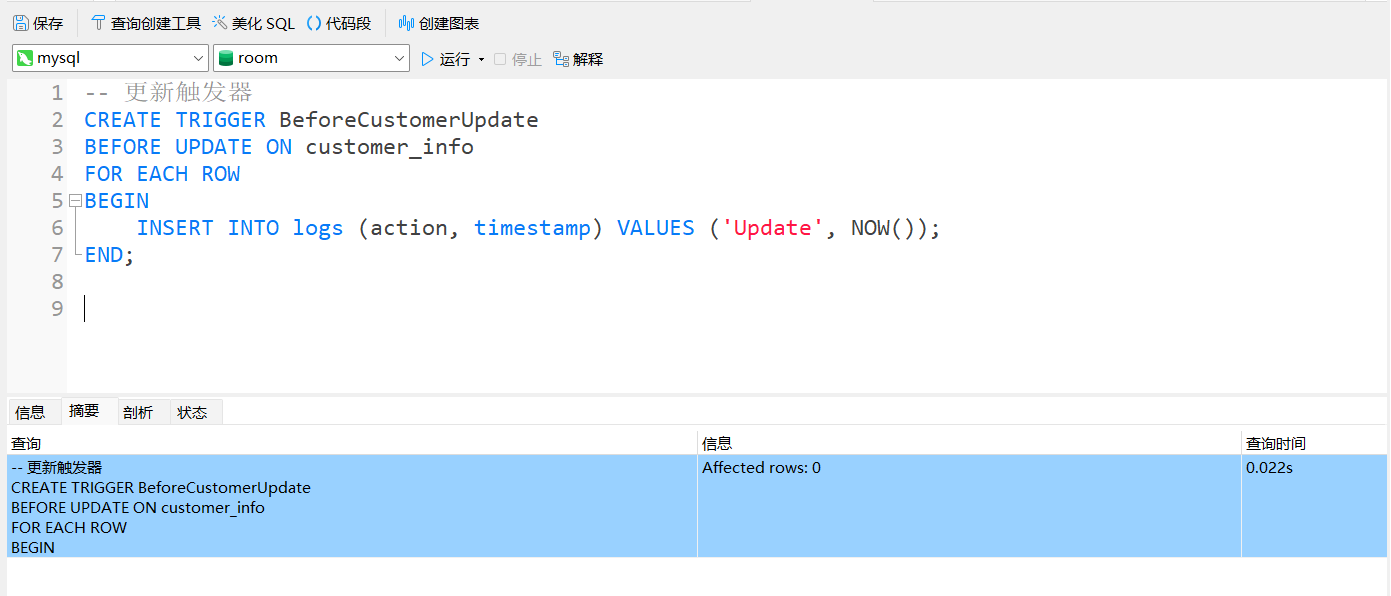
BEFORE UPDATE ON customer\_info

FOR EACH ROW

BEGIN

INSERT INTO logs (action, timestamp) VALUES ('Update', NOW());

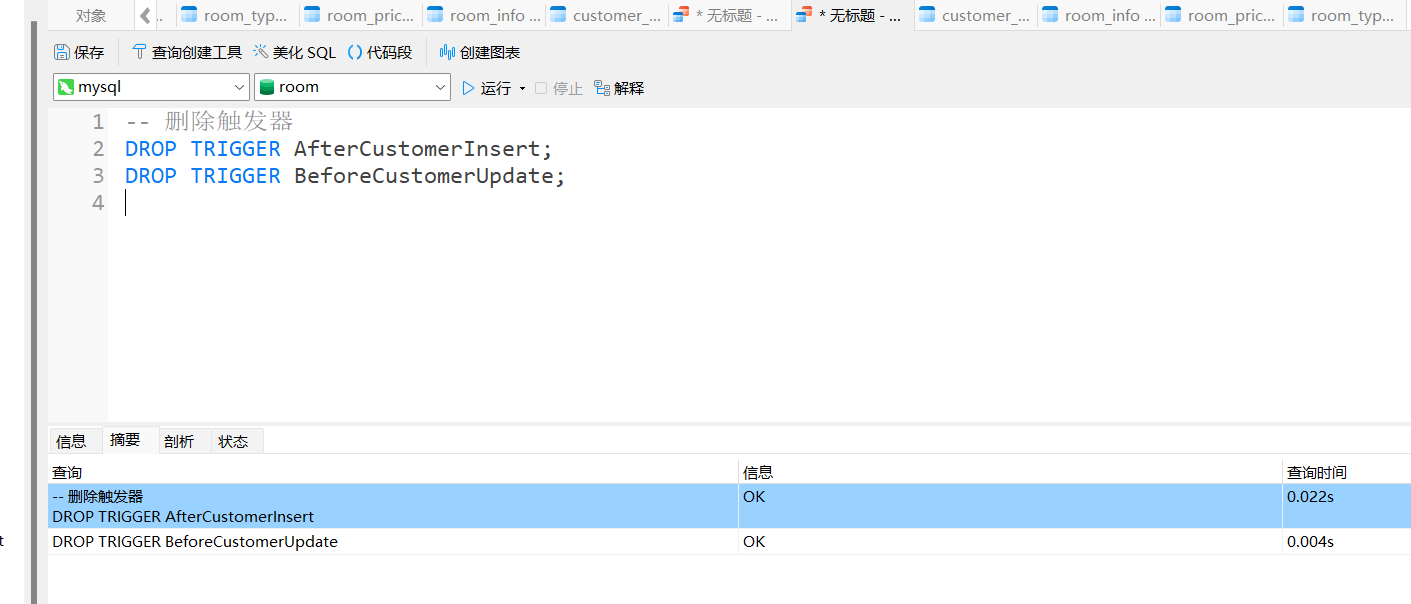
END;



-- 删除触发器

DROP TRIGGER AfterCustomerInsert;

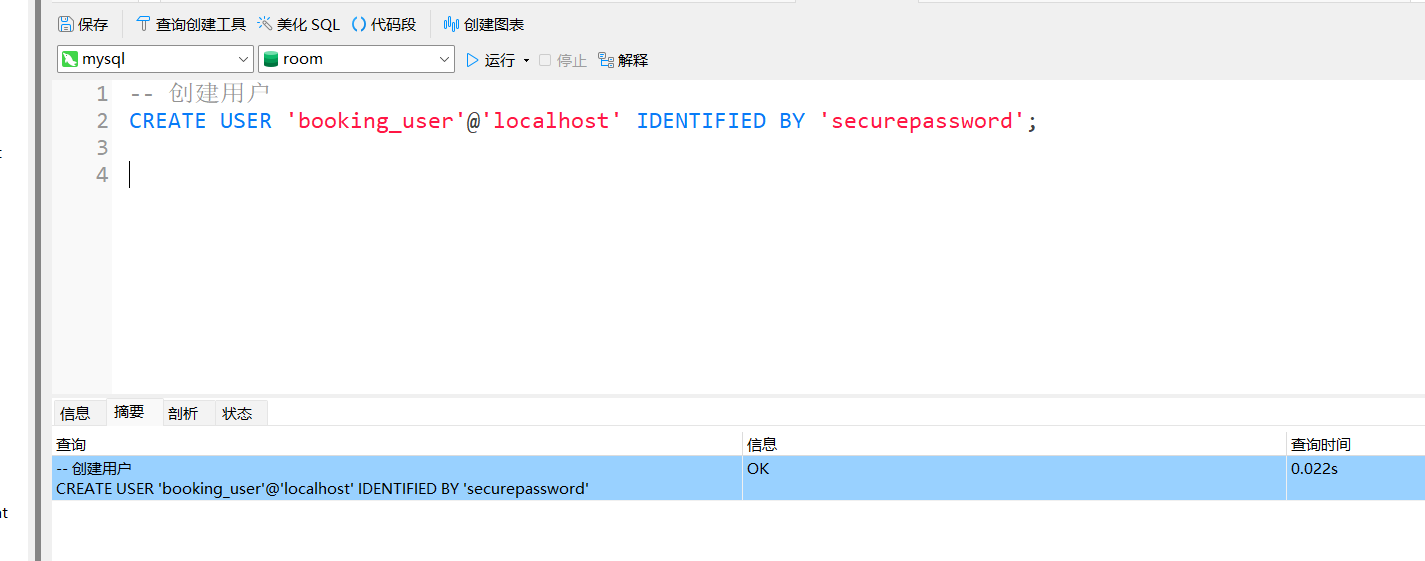
DROP TRIGGER BeforeCustomerUpdate;



10 权限管理：用户的添加和修改（用户名和密码都要修改），权限的授予和撤销

-- 创建用户

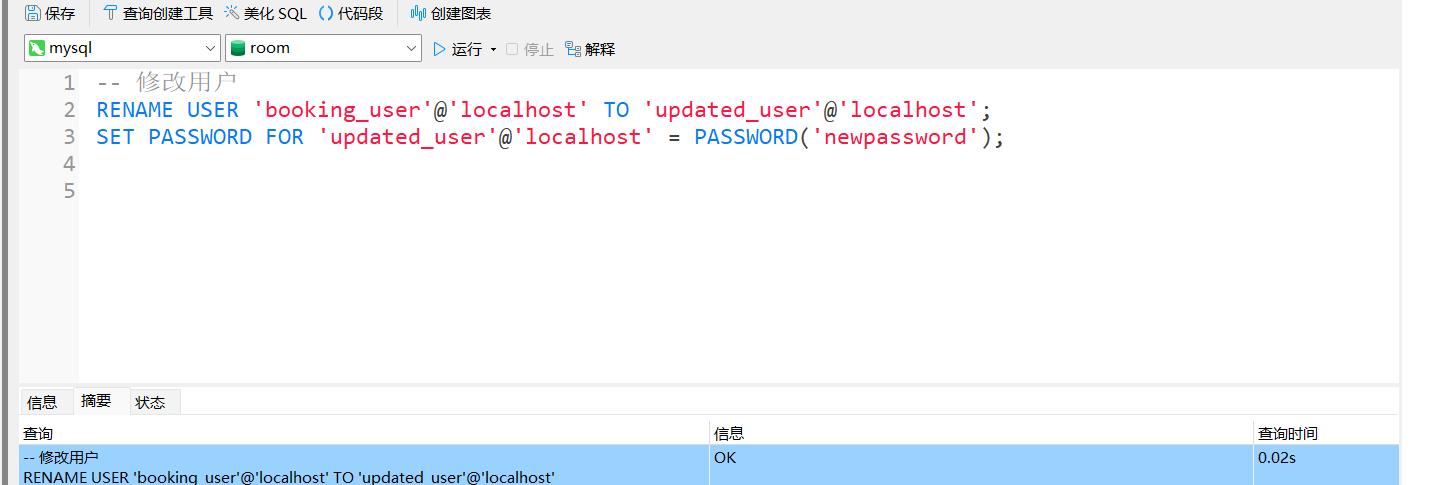
CREATE USER 'booking\_user'@'localhost' IDENTIFIED BY 'securepassword';



-- 修改用户

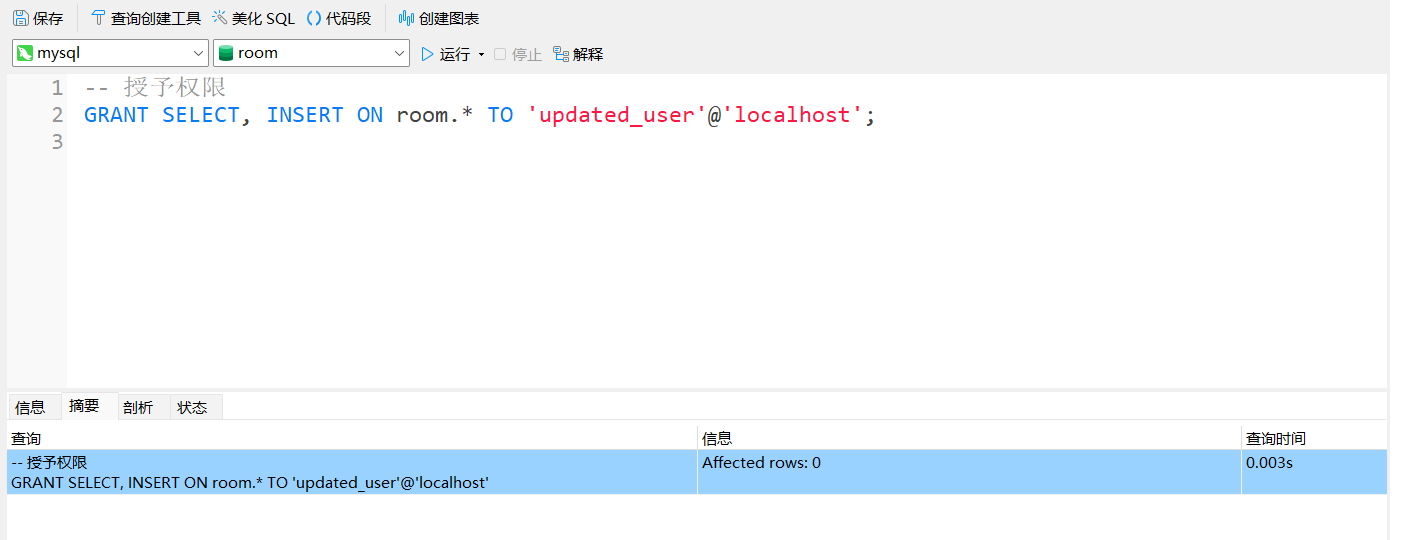
RENAME USER 'booking\_user'@'localhost' TO 'updated\_user'@'localhost';

SET PASSWORD FOR 'updated\_user'@'localhost' = PASSWORD('newpassword');



-- 授予权限

GRANT SELECT, INSERT ON room.\* TO 'updated\_user'@'localhost';



-- 撤销权限

REVOKE INSERT ON room.\* FROM 'updated\_user'@'localhost';

