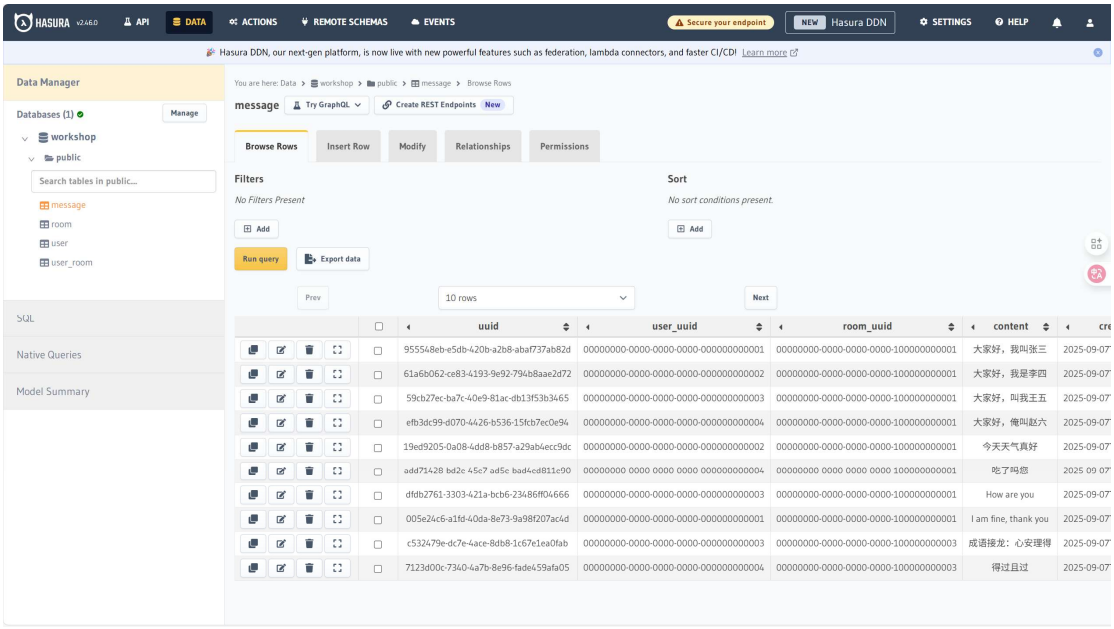
Hw_database

2.1:

2.1.1,2.1.2:已完成环境配置，在 setting 中导入 metadata 并打开数据库



已在 graphql 中实现了如下功能：

在房间内发信息，以及实时获取房间内的信息

新建房间，按邀请码查找房间，查询加入了哪些房间，加入房间

2.1.3：

前端通过如下语句生成可供调用的函数（虽然目前好像还没有被用）：

```
export type AddMessageMutationFn = Apollo.MutationFunction<AddMessageMutation, AddMessageMutationVariables>;
```

被调用之后则发送 post 请求到/register，被后端处理，发给 hasura

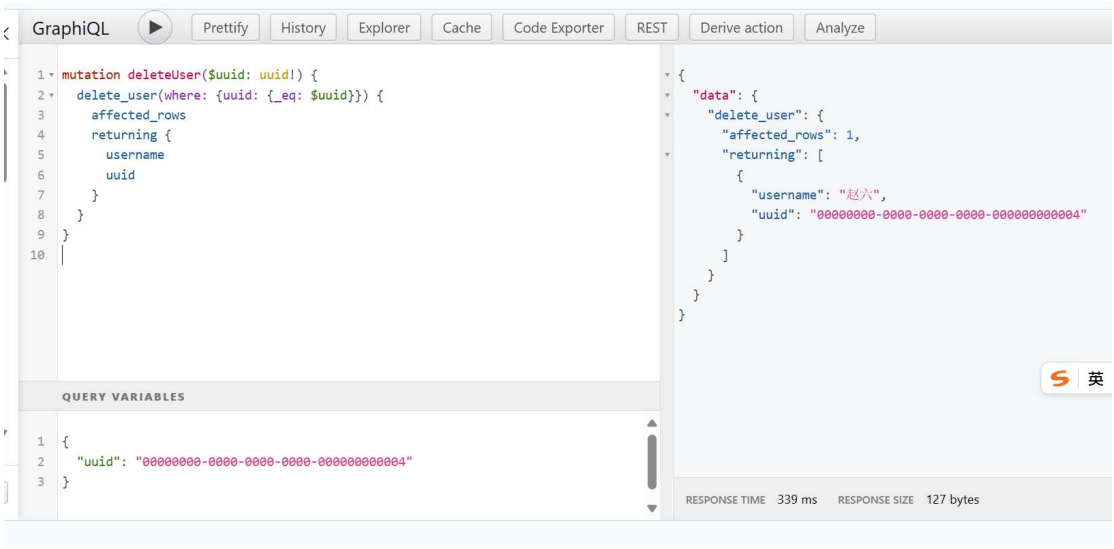
再被 hasura 经过权限认证后变成 sql 语言，返回数据。

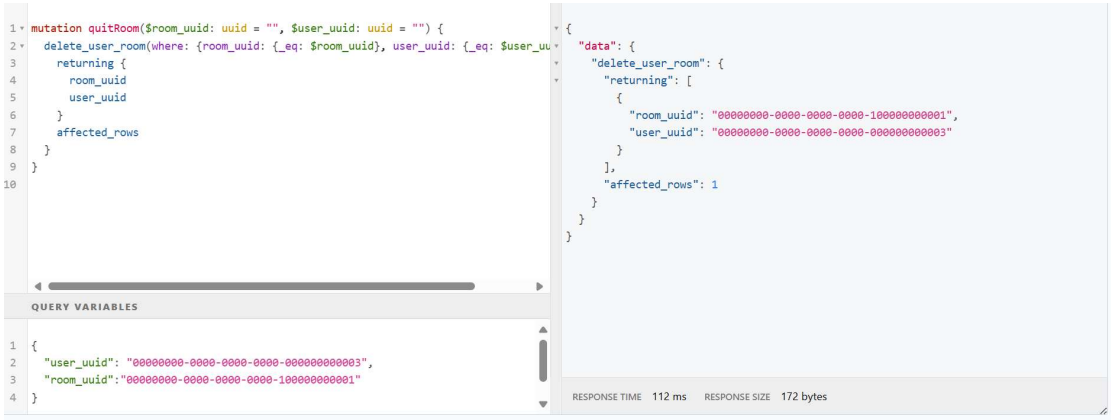
```
const mutationResult = await graphql.addUser({ username: username, password: password });

    const payload: userJWTPayload = {

      uuid: mutationResult.insert_user_one?.uuid,

      "https://hasura.io/jwt/claims": {

        "x-hasura-allowed-roles": ["admin", "user"],
```

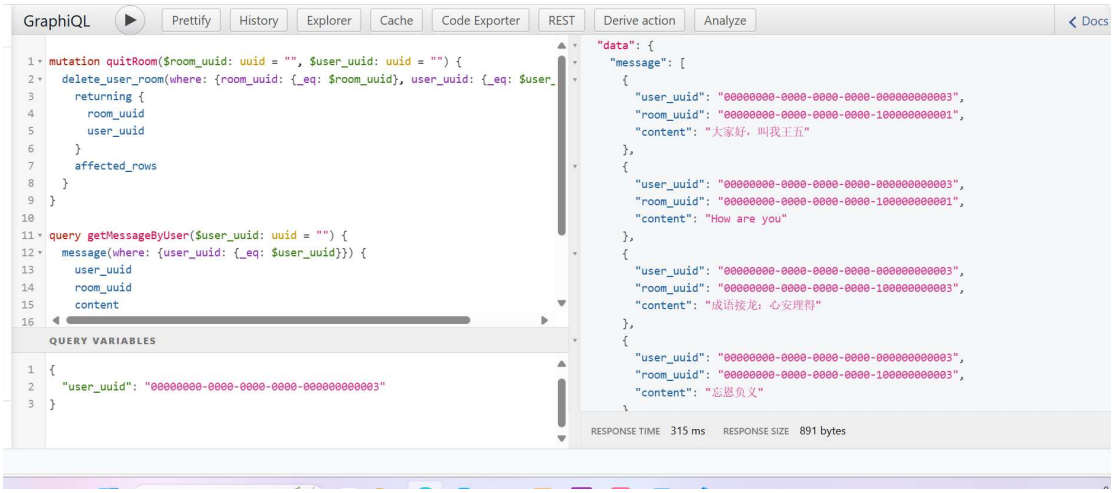
```
        "x-hasura-default-role": "user",
      },
    };
    const token = jwt.sign(payload, process.env.JWT_SECRET!, {
      expiresIn: "24h",
    });
```

2.2:

分别按要求向三个 graphql 中插入如下语句，已在 hasura api 中验证之：






GraphiQL
Prettify
History
Explorer
Cache
Code Exporter
REST
Derive action
Analyze
Docs
mutation quitRoom($room_uuid: uuid = "", $user_uuid: uuid = "") {
delete_user_room(where: {room_uuid: {_eq: $room_uuid}, user_uuid: {_eq: $user_
returning {
room_uuid
user_uuid
}
affected_rows
}
}
query getMessageByUser($user_uuid: uuid = "") {
message(where: {user_uuid: {_eq: $user_uuid}}) {
user_uuid
room_uuid
content
QUERY VARIABLES
{
"user_uuid": "00000000-0000-0000-0000-000000000003"
}
"data": {
"message": [
{
"user_uuid": "00000000-0000-0000-0000-000000000003",
"room_uuid": "00000000-0000-0000-0000-100000000001",
"content": "大家好，叫我王五"
},
{
"user_uuid": "00000000-0000-0000-0000-000000000003",
"room_uuid": "00000000-0000-0000-0000-100000000001",
"content": "How are you"
},
{
"user_uuid": "00000000-0000-0000-0000-000000000003",
"room_uuid": "00000000-0000-0000-0000-100000000003",
"content": "成语接龙：心安理得"
},
{
"user_uuid": "00000000-0000-0000-0000-000000000003",
"room_uuid": "00000000-0000-0000-0000-100000000003",
"content": "忘恩负义"
RESPONSE TIME 315 ms
RESPONSE SIZE 891 bytes