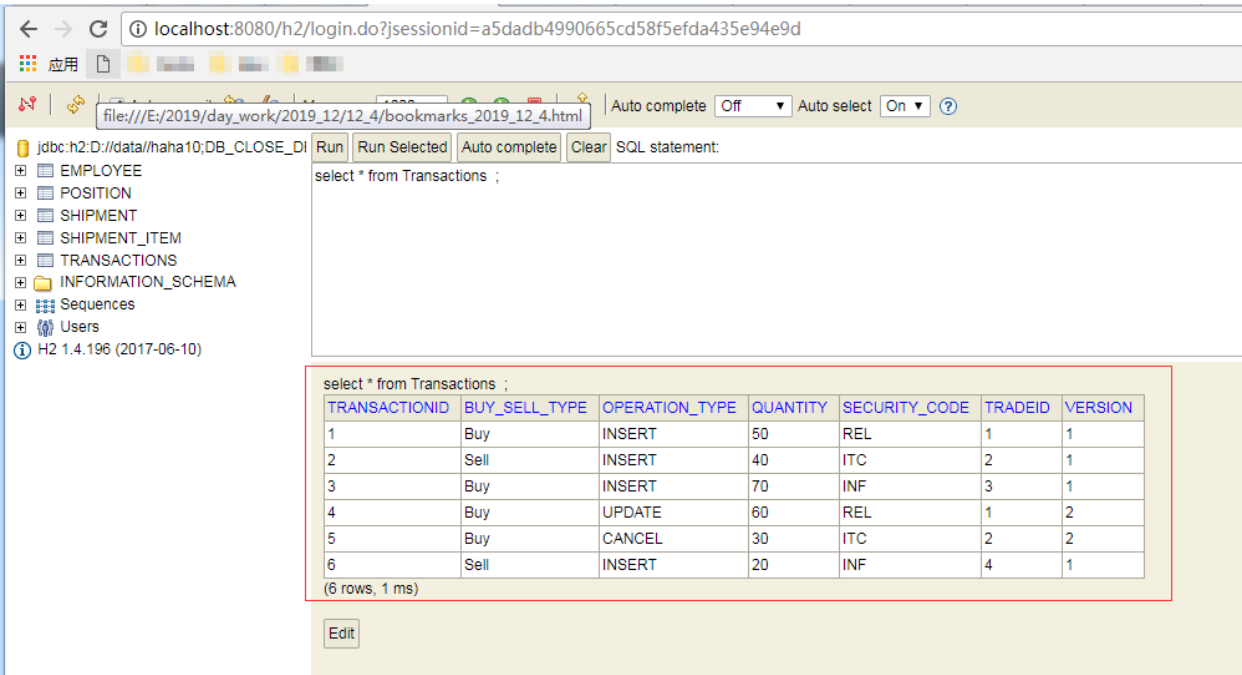


题2-Equity Positions

设计

【1】初始化Transactions 数据

接口具体见postman倒出的附件文档



【2】全部查出来list

```
select transactionid,tradeid,version,security_code,quantity,operation_type,buy_sell_type from Transactions order by transactionid asc;
```

【3】根据SecurityCode分组

```
SECURITY_CODE = REL    （目前有2条） -- 放入list1
SECURITY_CODE =ITC     （目前有2条） -- 放入list2
SECURITY_CODE =INF     （目前有2条） -- 放入list3
```

【4】遍历，根据Buy/Sell 的标记，判断是加号+，还是减号-

Buy 则+Quantity

Sell 则-Quantity

【5】根据Insert/Update/Cancel 标记，路由不同的计算规则

insert 则 把上一步骤的Quantity加起来

update 则直接更新为本次动作的Quantity

cancel 则把所有前面步骤的Quantity修改为0

【6】把动态计算的结果保存到 Position 表

security\_code -- Quantity

REL -- +60

ITC -- 0

INF -- +50

【7】查询 Position 表，UI展示

接口具体见postman倒出的附件文档

测试输出结果如下：

The screenshot shows a web browser window with the address bar displaying `localhost:8080/h2/login.do?jsessionid=a5dad4990665cd58f5efda435e94e9d`. Below the browser window, a database client interface (likely DBeaver) is visible. It shows a list of database tables on the left, including `EMPLOYEE`, `POSITION`, `SHIPMENT`, `SHIPMENT_ITEM`, `TRANSACTIONS`, `INFORMATION_SCHEMA`, `Sequences`, and `Users`. The `POSITION` table is selected. The SQL statement entered is `select id,quantity,security_code from Position order by quantity desc;`. The result is displayed in a table with three rows:

ID	QUANTITY	SECURITY_CODE
74	60	REL
76	50	INF
75	0	ITC

Below the table, it indicates `(3 rows, 1 ms)` and there is an `Edit` button.

---