***SellingGetChestList:***

Code:

router.get('/SellingGetChestList',

    Authentication,

    isMyProvider,

    get\_\_All\_\_Uuid\_departmentGroup,

    // get\_\_All\_\_Uuid\_department,

    get\_\_All\_\_Uuid\_chest,

    (req, res) => {

    const pageIndex = req.query.pageIndex;

    const pageSize = req.query.pageSize;

    const all\_uuid\_chest = req.all\_uuid\_chest;

    const all\_uuid\_chest\_sq = [];

    for (let i = 0; i < all\_uuid\_chest.length; i++) {

        const uuid\_chest\_\_ = {uuid\_chest: all\_uuid\_chest[i]};

        all\_uuid\_chest\_sq.push(uuid\_chest\_\_);

    }

    chestCRUD.sellingRead(all\_uuid\_chest\_sq, Number(pageIndex), Number(pageSize), (chestList, err) => {

        if (err) {

            logEvents(`${req.url}---${req.method}---${err}`);

            return res.status(500).send(err);

        } else {

            if (chestList && chestList!==null) {

                return res.status(200).json({

                    chestList: chestList,

                    success: true,

                    message: `Chest list is read successly (${service}) !`

                });

            } else {

                return res.status(200).json({

                    chestList: chestList,

                    success: false,

                    message: `Chest list is NOT read successly (${service}) !`

                })

            }

        }

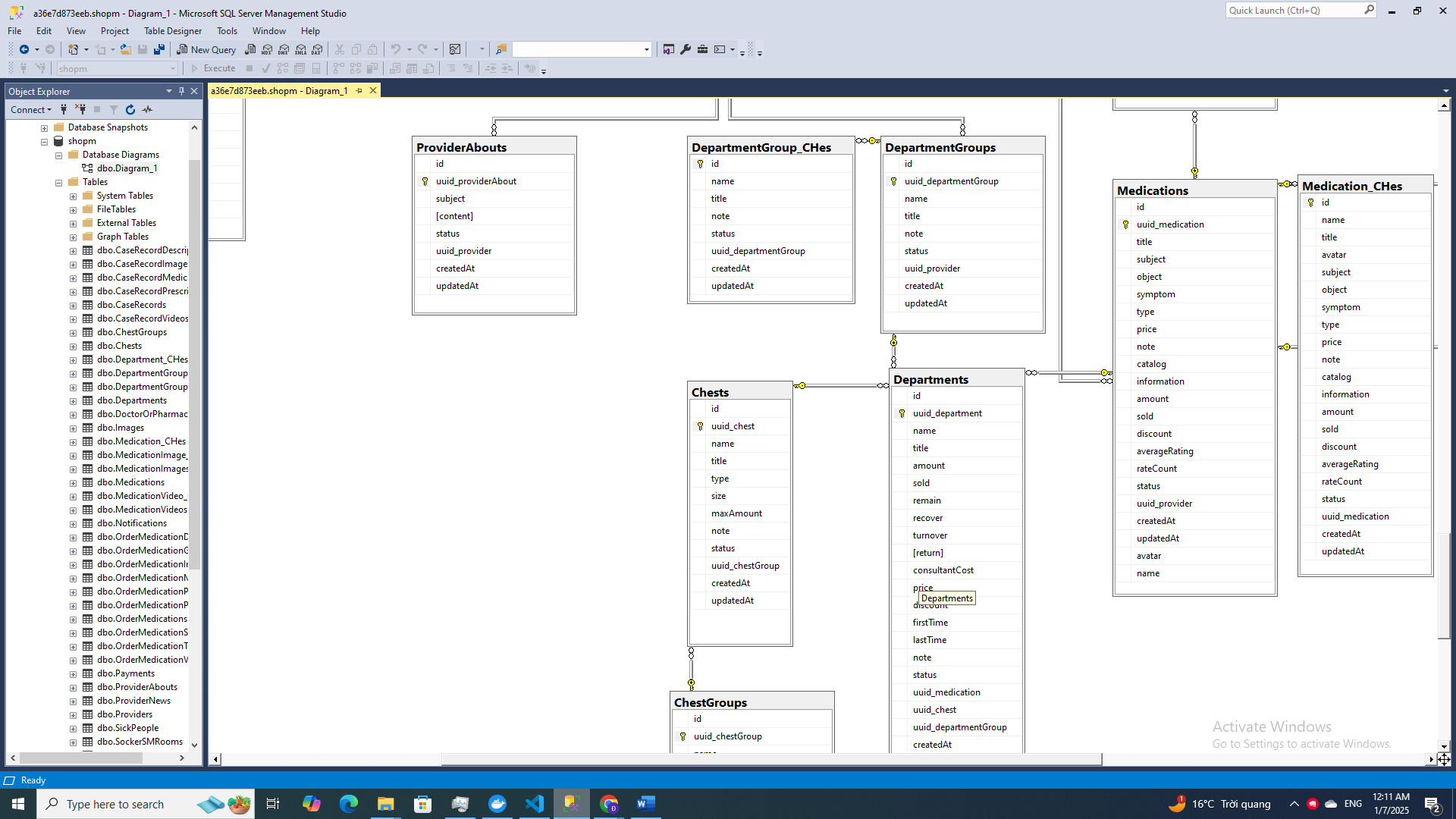
    })

})

Middleware:

* **isMyProvider**: Ensure provider that you store in cookie is your.
* **get\_\_All\_\_Uuid\_departmentGroup**: only take uuid\_departmentGroup
* **get\_\_All\_\_Uuid\_chest:** take uuid\_chest
* **main code:** implement main task.

Database:

* 
* We can’t get chest directly. It must via other table. Steps is middles.