

MSSQL

■■■■■■■■■■

01.■■■■ [SQL Server Management Studio](#)

02.■■■■■■■■■■

■■■■ table■■■

```
SELECT
    t.name AS TableName,
    SUM(p.rows) AS RowCounts
FROM sys.tables t
INNER JOIN sys.partitions p ON t.object_id = p.object_id
WHERE p.index_id IN (0,1) -- 0=heap, 1=clustered index
GROUP BY t.name
ORDER BY RowCounts DESC;
```

■■ comProdRec ■■ 10■

```
SELECT TOP 10 *
FROM comProdRec;
```

■■ Python■■ API

```
import pyodbc
from flask import Flask, render_template, request, jsonify, redirect, url_for, session

app = Flask(__name__)
app.secret_key = "yoursecertkey"

DB_CONFIG = {
    "CHIComp01": {
        "database": "yourdbname",
        "warehouses": [("G", "G ")],
        "default_warehouse": "G",
        "label": "■■ "
```

```
}  
}
```

```
ODBC_CONFIG = {  
    'server': 'yourserverip',  
    'username': 'sa',  
    'password': 'yourspassword',  
    'driver': '{ODBC Driver 17 for SQL Server}',  
}
```

```
API_KEY = "yourapikey"
```

```
def get_sku_map_from_db(database):
```

```
    conn_str = (  
        f"DRIVER={ODBC_CONFIG['driver']}";  
        f"SERVER={ODBC_CONFIG['server']}";  
        f"DATABASE={database}";  
        f"UID={ODBC_CONFIG['username']}";  
        f"PWD={ODBC_CONFIG['password']}"  
    )  
    conn = pyodbc.connect(conn_str)  
    cursor = conn.cursor()  
    cursor.execute("SELECT ProdID, ProdName FROM comProduct WHERE ProdID IS NOT NULL AND ProdName IS  
NOT NULL")  
    sku_map = {}  
    for row in cursor.fetchall():  
        sku_map[row.ProdID] = row.ProdName  
    cursor.close()  
    conn.close()  
    return sku_map
```

```
def query_stock_and_lend(database, warehouse_id, sku_list):
```

```
    conn_str = (  
        f"DRIVER={ODBC_CONFIG['driver']}";  
        f"SERVER={ODBC_CONFIG['server']}";  
        f"DATABASE={database}";  
        f"UID={ODBC_CONFIG['username']}";  
        f"PWD={ODBC_CONFIG['password']}"  
    )  
    conn = pyodbc.connect(conn_str)
```

```

cursor = conn.cursor()
sql_stock = """
SELECT ProdID, SUM(Quantity) AS NowStock
FROM StkYearMonthQty
WHERE WareID = ? AND ProdID = ?
GROUP BY ProdID
"""
sql_lend = """
SELECT ProdID, SUM(QtyRemain) AS NowLendQty
FROM stkBorrowSub
WHERE WareHouseID = ? AND ProdID = ?
GROUP BY ProdID
"""
result = {}
for sku in sku_list:
    cursor.execute(sql_stock, (warehouse_id, sku))
    row_stock = cursor.fetchone()
    now_stock = float(row_stock.NowStock) if row_stock and row_stock.NowStock is not None else 0
    cursor.execute(sql_lend, (warehouse_id, sku))
    row_lend = cursor.fetchone()
    now_lend = float(row_lend.NowLendQty) if row_lend and row_lend.NowLendQty is not None else 0
    real_stock = now_stock - now_lend
    result[sku] = {
        "NowStock": now_stock,
        "NowLendQty": now_lend,
        "RealStock": real_stock
    }
cursor.close()
conn.close()
return result

```

```

def search_sku(sku_map, keyword):
    keyword = keyword.lower()
    return [(sku, name) for sku, name in sku_map.items()
            if keyword in sku.lower() or keyword in name.lower()]

```

```
@app.route("/login", methods=["GET", "POST"])
```

```
def login():
```

```
    error = ""
```

```
    if request.method == "POST":
```

```
key = request.form.get("key", "")
if key == API_KEY:
    session['key'] = key
    return redirect(url_for('index'))
else:
    error = "XXXXXXXXXX"
return render_template("login.html", error=error, db_label="")
```

```
@app.route("/logout", methods=["POST"])
```

```
def logout():
```

```
    session.pop('key', None)
    return redirect(url_for('login'))
```

```
@app.route("/", methods=["GET", "POST"])
```

```
def index():
```

```
    if session.get("key") != API_KEY:
        return redirect(url_for("login"))
```

```
db_selected = "CHIComp01"
```

```
db_info = DB_CONFIG[db_selected]
```

```
warehouse_selected = "G"
```

```
keyword = request.form.get("keyword", "").strip()
```

```
sku_map = get_sku_map_from_db(db_info["database"])
```

```
warehouse_list = db_info["warehouses"]
```

```
result = []
```

```
searched = False
```

```
if keyword:
```

```
    matches = search_sku(sku_map, keyword)
```

```
    if matches:
```

```
        stock_data = query_stock_and_lend(
            db_info["database"], warehouse_selected, [sku for sku, _ in matches]
        )
```

```
        for sku, name in matches:
```

```
            now_stock = stock_data[sku]["NowStock"]
```

```
            now_lend = stock_data[sku]["NowLendQty"]
```

```
            real_stock = stock_data[sku]["RealStock"]
```

```
            result.append({
```

```
                "ProdID": sku,
```

```
                "Name": name,
```

```

        "NowStock": now_stock,
        "NowLendQty": now_lend,
        "RealStock": real_stock
    })
    searched = True

return render_template(
    "index.html",
    database_list=DB_CONFIG,
    db_selected=db_selected,
    warehouse_list=warehouse_list,
    warehouse_selected=warehouse_selected,
    result=result,
    keyword=keyword,
    searched=searched,
    db_label=db_info["label"]
)

@app.route("/sku/<sku_id>")
def get_sku_stock(sku_id):
    key = request.args.get("key", "")
    if key != API_KEY:
        return jsonify({"error": "API Key Error"}), 403

    db_selected = "CHIComp01"
    warehouse_selected = "G"
    sku_map = get_sku_map_from_db(DB_CONFIG[db_selected]["database"])

    if sku_id not in sku_map:
        return jsonify({"error": "SKU Not Found"}), 404

    stock_data = query_stock_and_lend(
        DB_CONFIG[db_selected]["database"], warehouse_selected, [sku_id]
    )
    real_stock = stock_data[sku_id]["RealStock"]
    return jsonify({"sku": sku_id, "RealStock": real_stock})

if __name__ == "__main__":
    app.run(debug=True, host="0.0.0.0", port=5098, ssl_context=('cert.pem', 'key.pem'))

```

