IFT 598: Middleware Prog & Database Sec (2022 Fall)

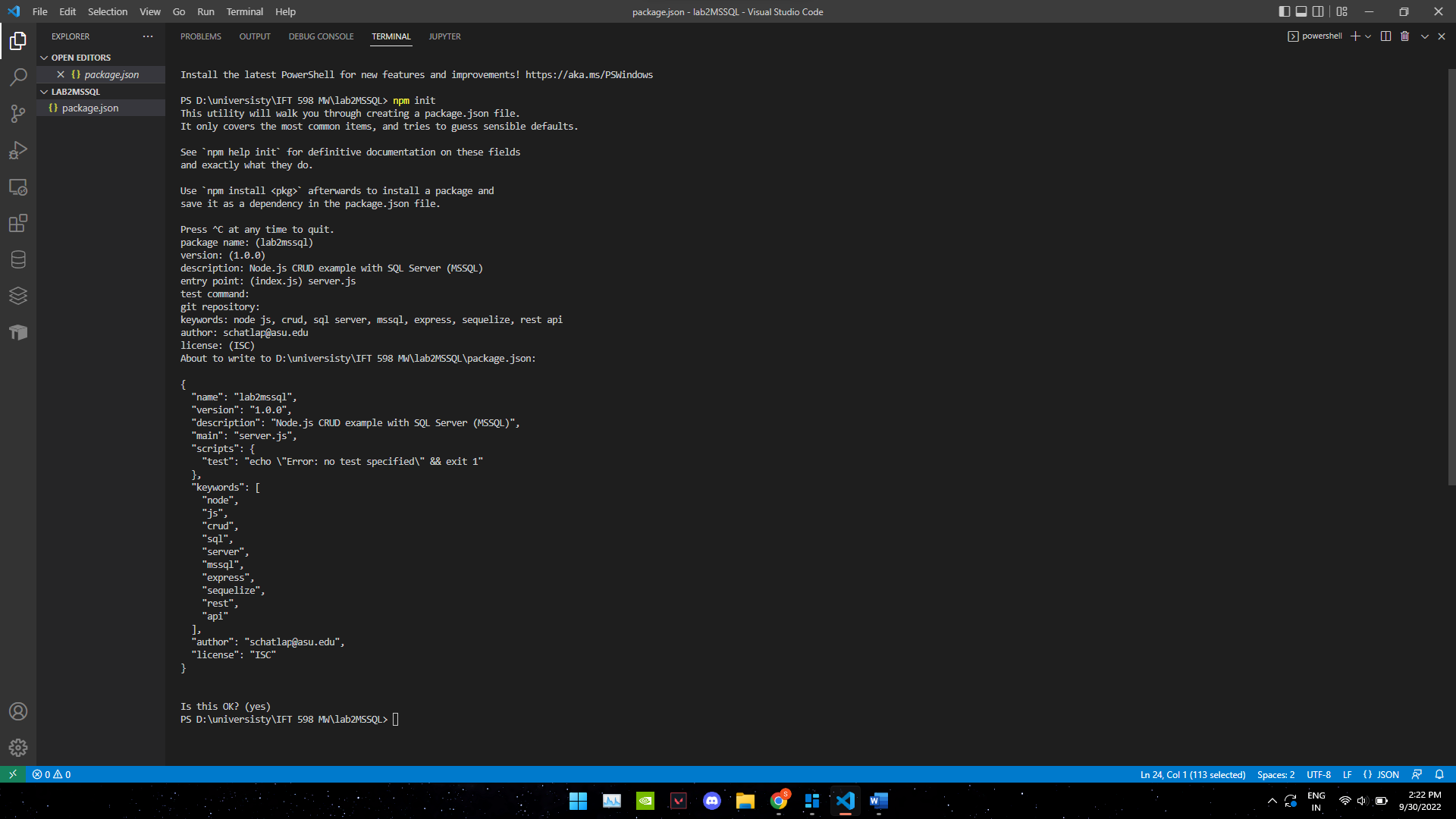
**Module 3 Assignment: Data Access**

Name: Sai Mrunaal Chatlapally

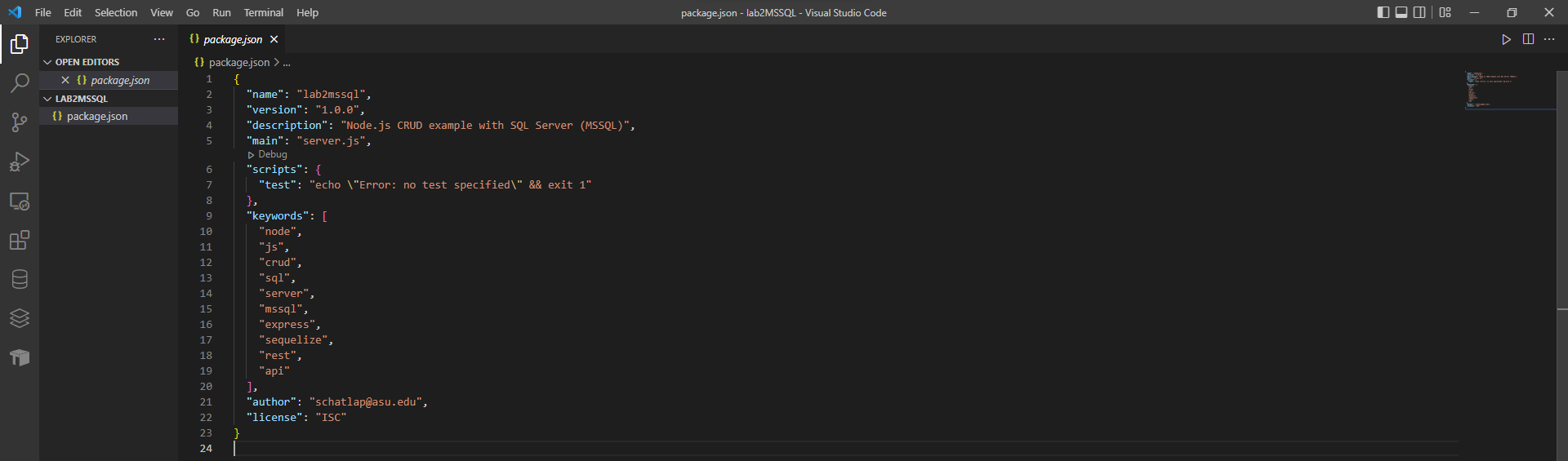
Class: 2022Fall-P-IFT458-IFT598-82341-82345

Date: 10-02-2022

* Npm init



* Package.json

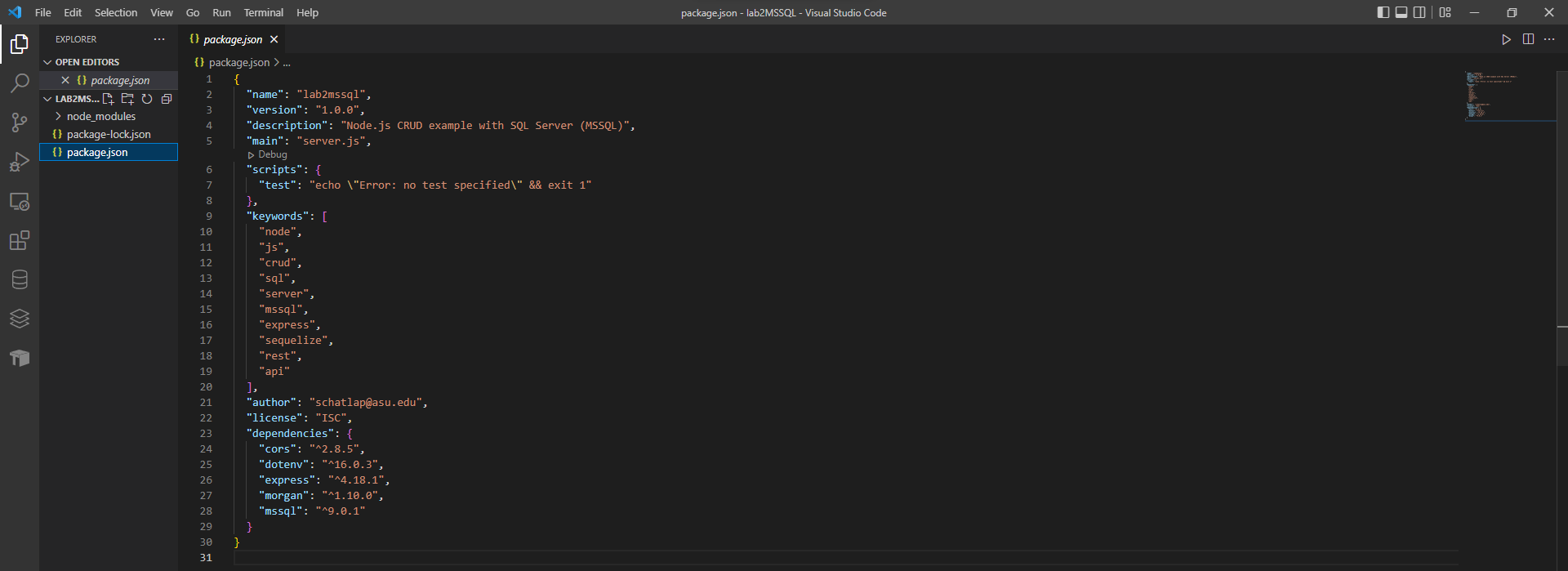


* Installing necessary modules:

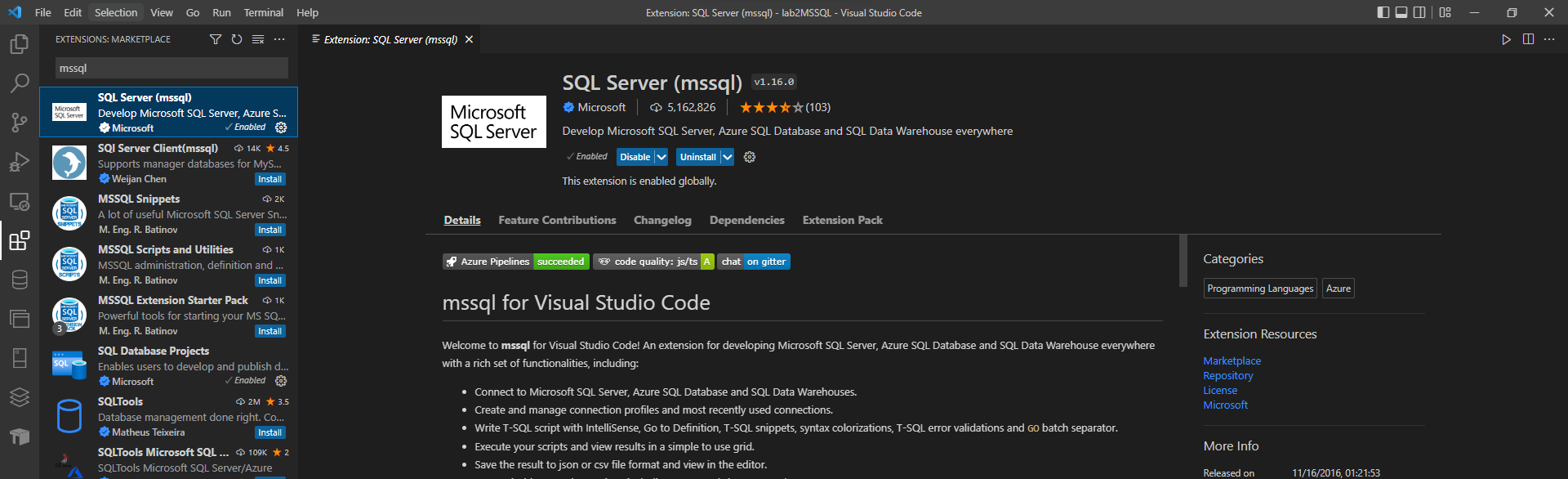
npm install express dotenv mssql morgan cors --save



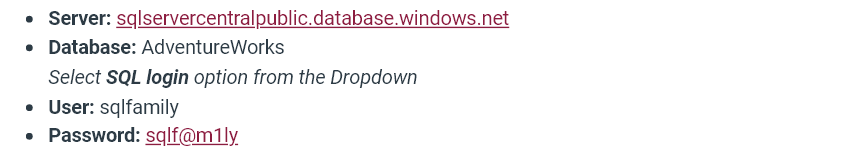
* + Updated package.json



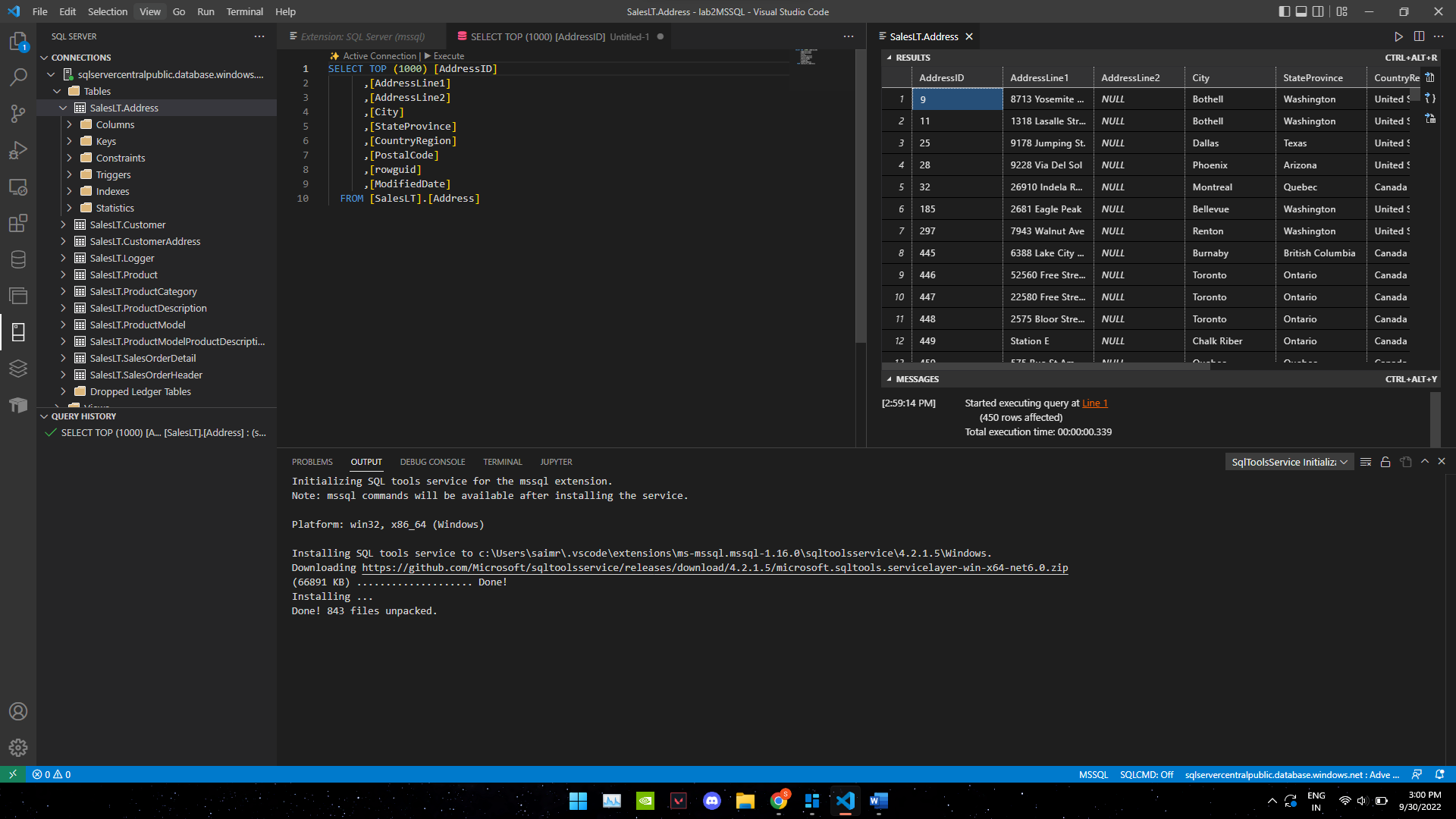
* Installing necessary extension (MSSQL):



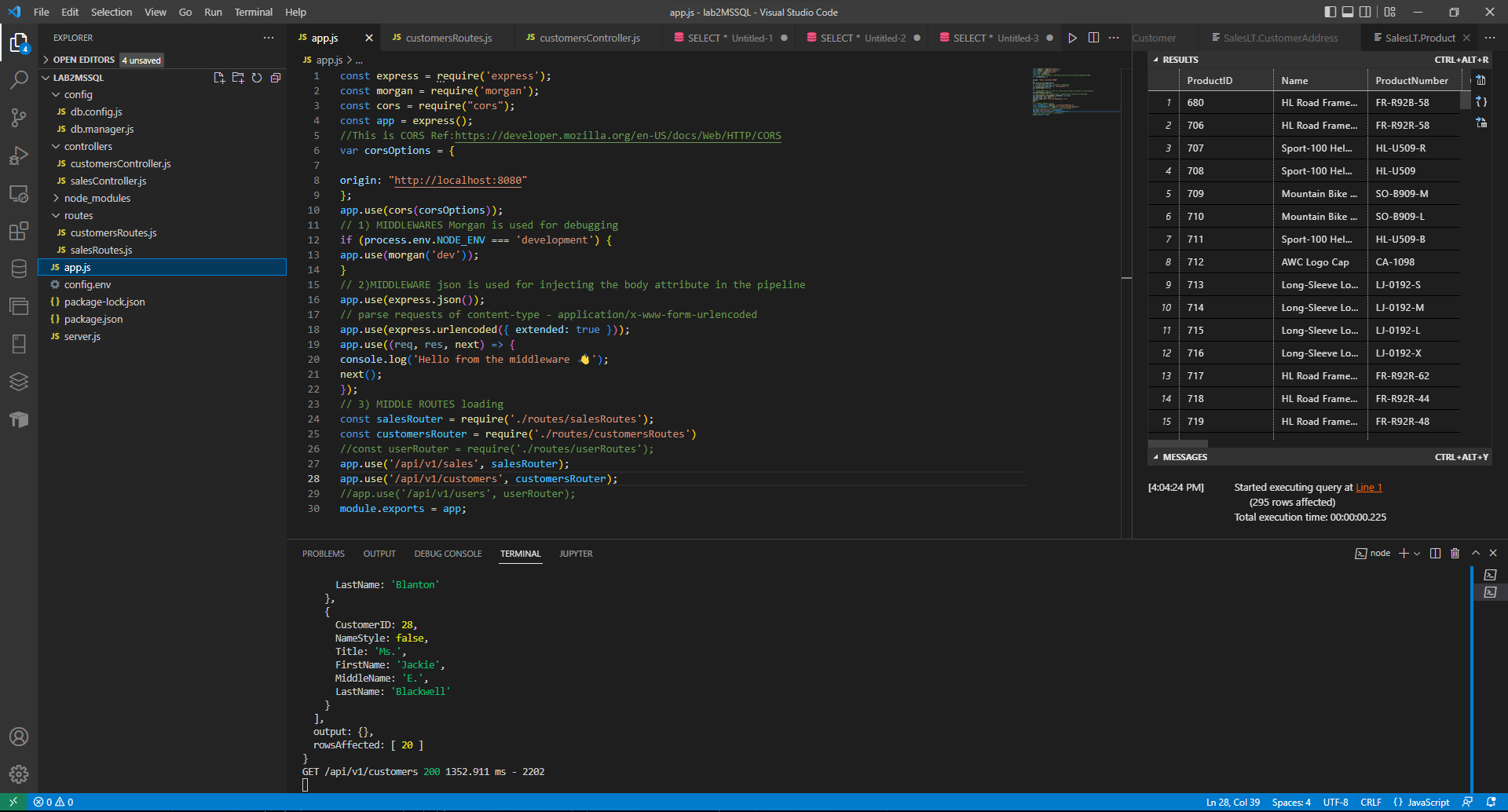
* Connecting to SQL Server:



Proof of installation



* Setup express web server
  + App.js



**Code:**

const express = require('express');

const morgan = require('morgan');

const cors = require("cors");

const app = express();

//This is CORS Ref:https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

var corsOptions = {

origin: "http://localhost:8080"

};

app.use(cors(corsOptions));

// 1) MIDDLEWARES Morgan is used for debugging

if (process.env.NODE\_ENV === 'development') {

app.use(morgan('dev'));

}

// 2)MIDDLEWARE json is used for injecting the body attribute in the pipeline

app.use(express.json());

// parse requests of content-type - application/x-www-form-urlencoded

app.use(express.urlencoded({ extended: true }));

app.use((req, res, next) => {

console.log('Hello from the middleware 👋');

next();

});

// 3) MIDDLE ROUTES loading

const salesRouter = require('./routes/salesRoutes');

const customersRouter = require('./routes/customersRoutes')

//const userRouter = require('./routes/userRoutes');

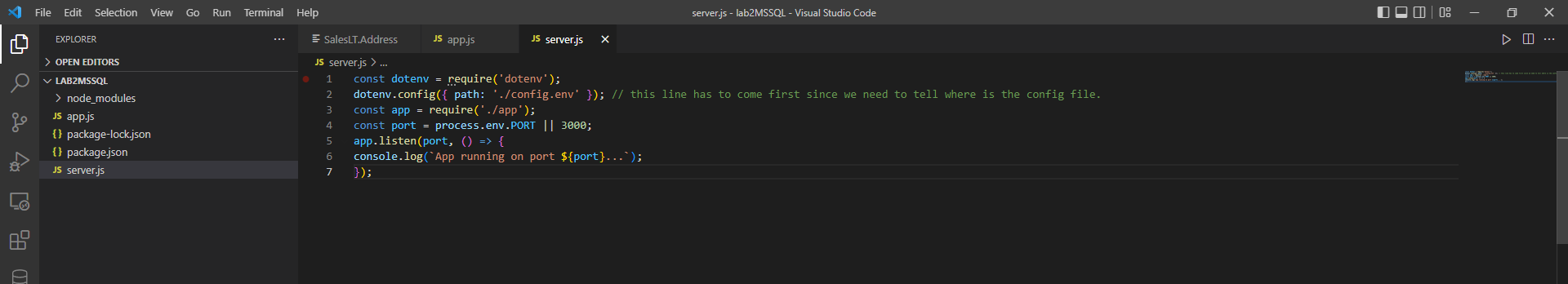
app.use('/api/v1/sales', salesRouter);

app.use('/api/v1/customers', customersRouter);

//app.use('/api/v1/users', userRouter);

module.exports = app;

* Server.js



Code:

const dotenv = require('dotenv');

dotenv.config({ path: './config.env' }); // this line has to come first since we need to tell where is the config file.

const app = require('./app');

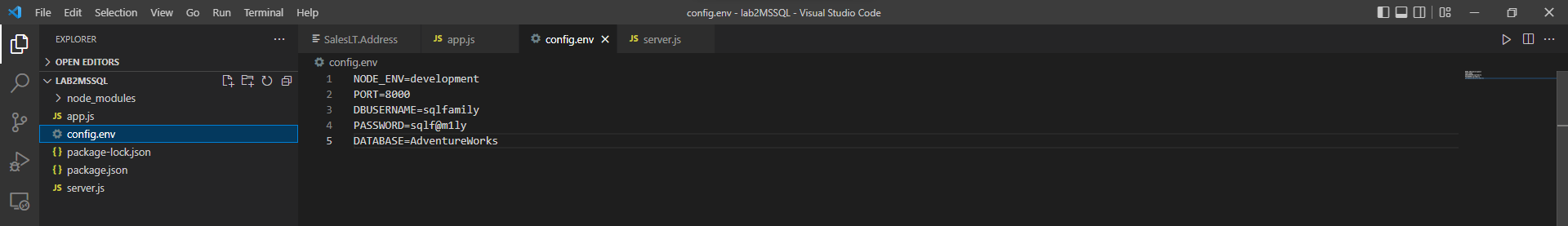
const port = process.env.PORT || 3000;

app.listen(port, () => {

console.log(`App running on port ${port}...`);

});

* Config.env



Code:

NODE\_ENV=development

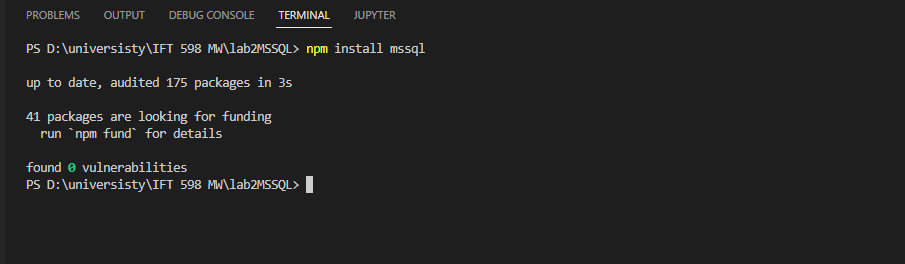
PORT=8000

DBUSERNAME=sqlfamily

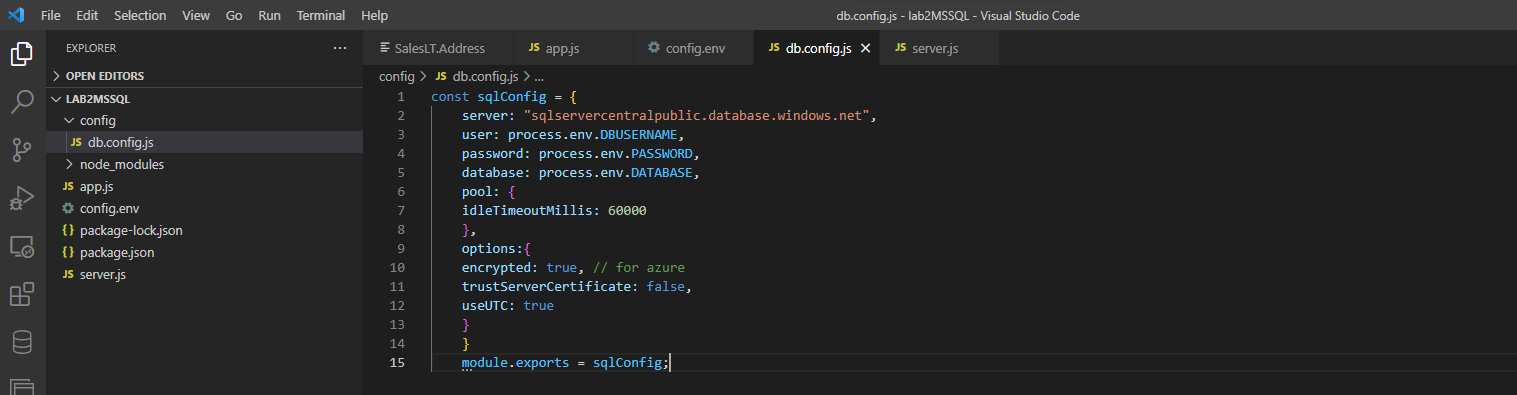
PASSWORD=sqlf@m1ly

DATABASE=AdventureWorks

* Install MSSQL package



* SQL server credentials



Code:

const sqlConfig = {

server: "sqlservercentralpublic.database.windows.net",

user: process.env.DBUSERNAME,

password: process.env.PASSWORD,

database: process.env.DATABASE,

pool: {

idleTimeoutMillis: 60000

},

options:{

encrypted: true, // for azure

trustServerCertificate: false,

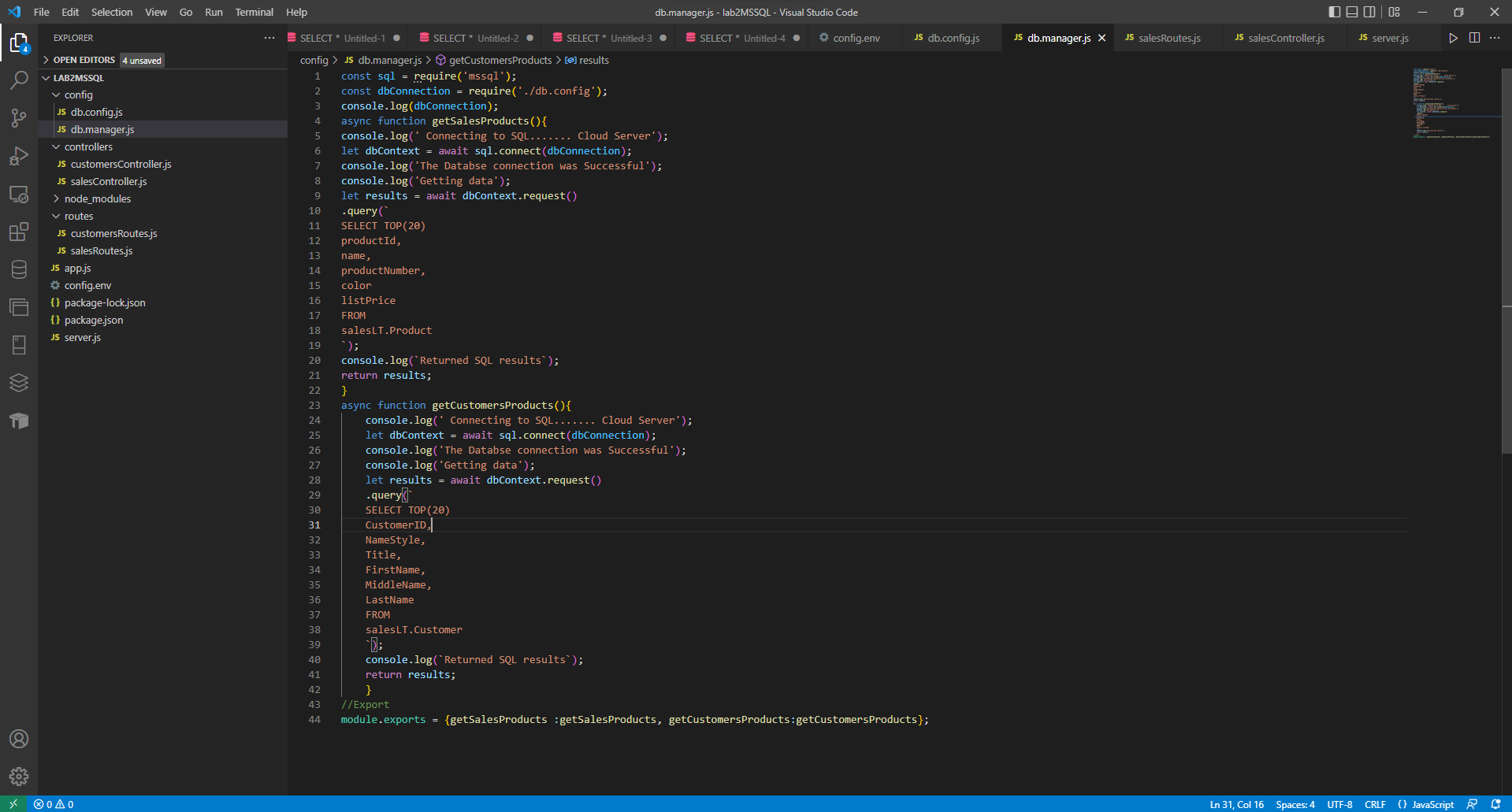
useUTC: true

}

}

module.exports = sqlConfig;

* Db.manager.js



Code:

const sql = require('mssql');

const dbConnection = require('./db.config');

console.log(dbConnection);

async function getSalesProducts(){

console.log(' Connecting to SQL....... Cloud Server');

let dbContext = await sql.connect(dbConnection);

console.log('The Databse connection was Successful');

console.log('Getting data');

let results = await dbContext.request()

.query(`

SELECT TOP(20)

productId,

name,

productNumber,

color

listPrice

FROM

salesLT.Product

`);

console.log(`Returned SQL results`);

return results;

}

async function getCustomersProducts(){

console.log(' Connecting to SQL....... Cloud Server');

let dbContext = await sql.connect(dbConnection);

console.log('The Databse connection was Successful');

console.log('Getting data');

let results = await dbContext.request()

.query(`

SELECT TOP(20)

CustomerID,

NameStyle,

Title,

FirstName,

MiddleName,

LastName

FROM

salesLT.Customer

`);

console.log(`Returned SQL results`);

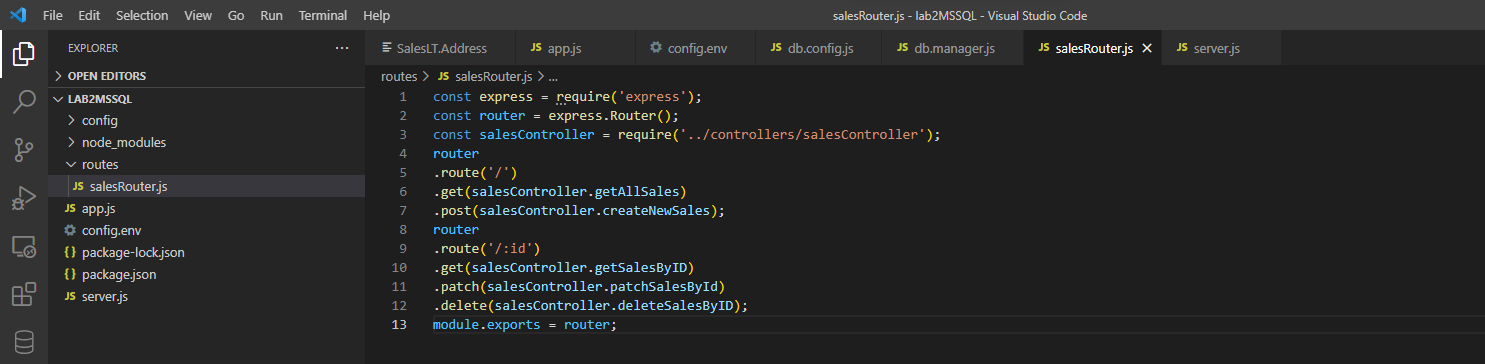
return results;

}

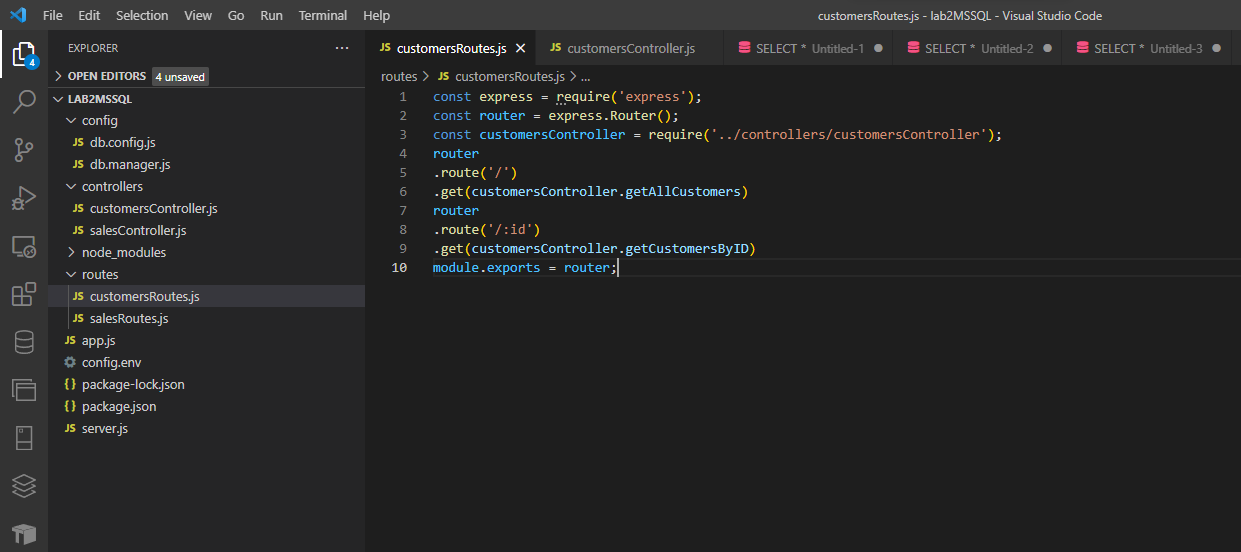
//Export

module.exports = {getSalesProducts :getSalesProducts, getCustomersProducts:getCustomersProducts};

* Creating routes folder
  + Salesroutes.js



* + customersRoutes:



Code:

const express = require('express');

const router = express.Router();

const customersController = require('../controllers/customersController');

router

.route('/')

.get(customersController.getAllCustomers)

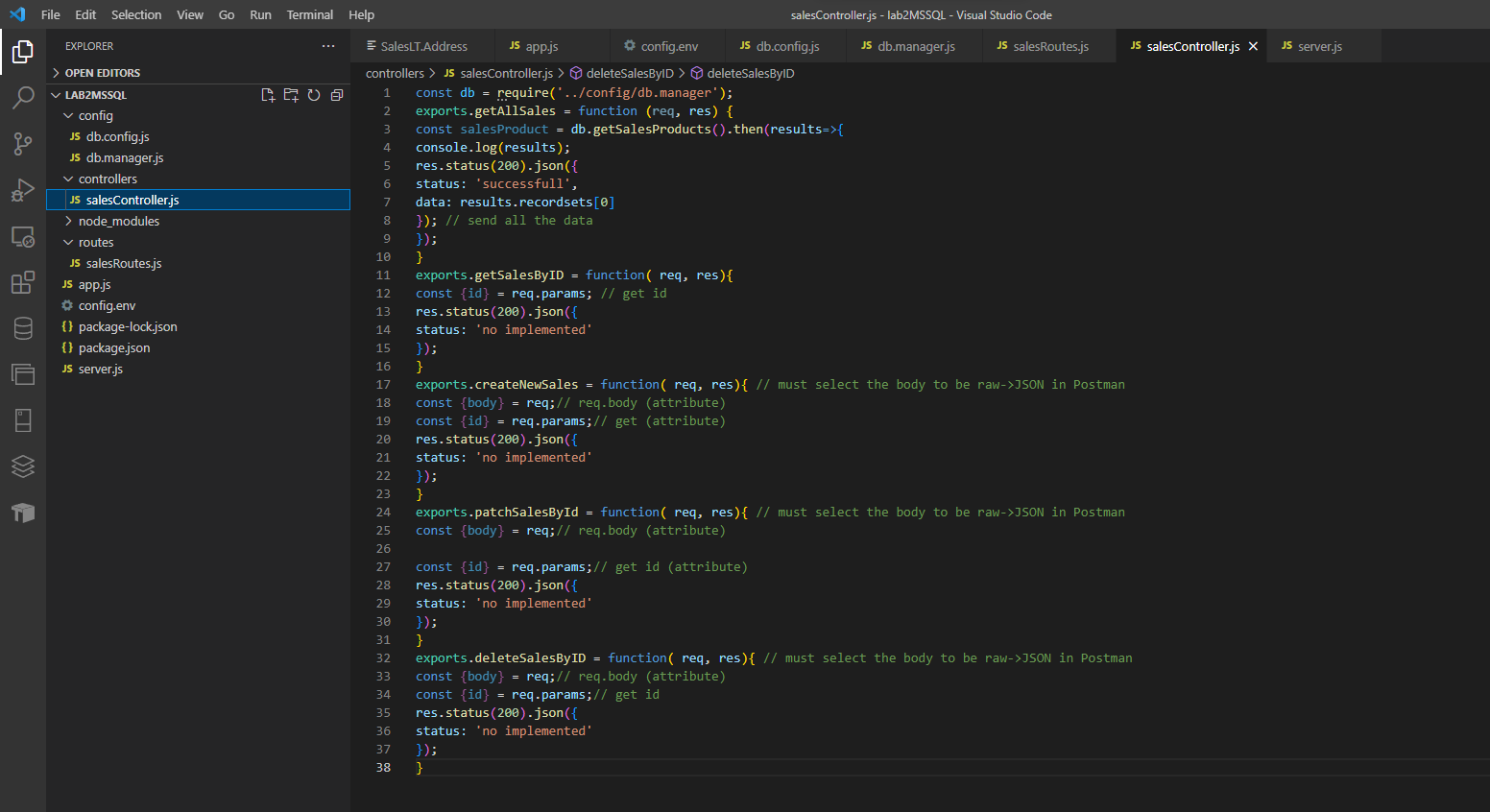
router

.route('/:id')

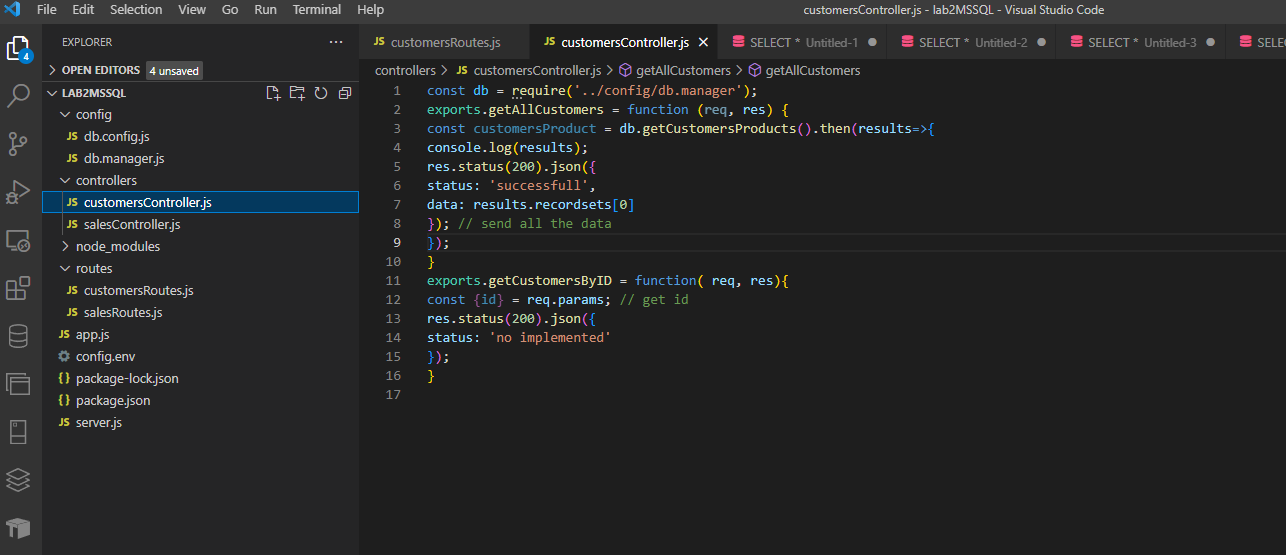
.get(customersController.getCustomersByID)

module.exports = router;

* Creating Controllers folder:
  + salesController.js



* + customersController.js



Code:

const db = require('../config/db.manager');

exports.getAllCustomers = function (req, res) {

const customersProduct = db.getCustomersProducts().then(results=>{

console.log(results);

res.status(200).json({

status: 'successfull',

data: results.recordsets[0]

}); // send all the data

});

}

exports.getCustomersByID = function( req, res){

const {id} = req.params; // get id

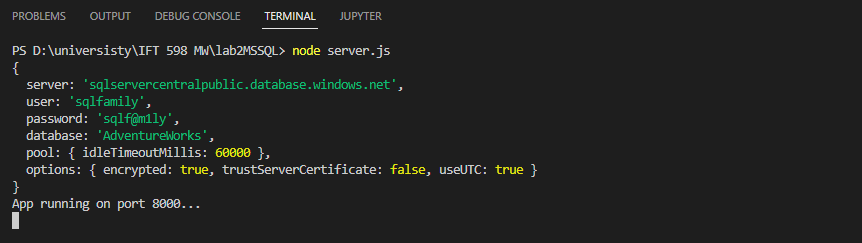
res.status(200).json({

status: 'no implemented'

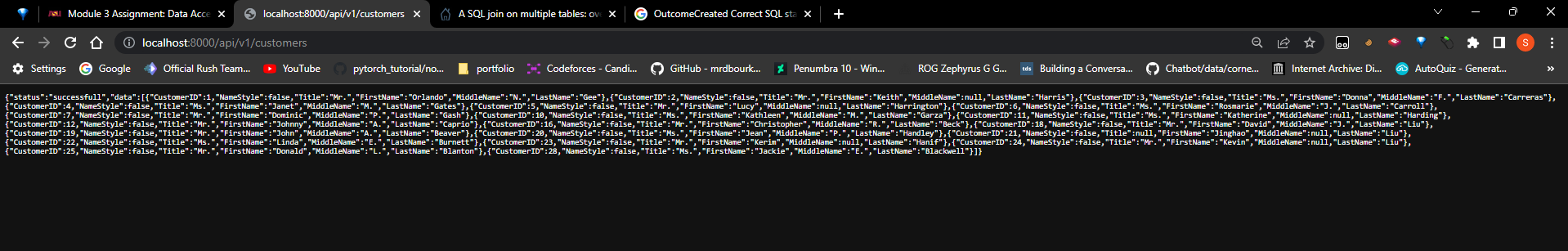
});

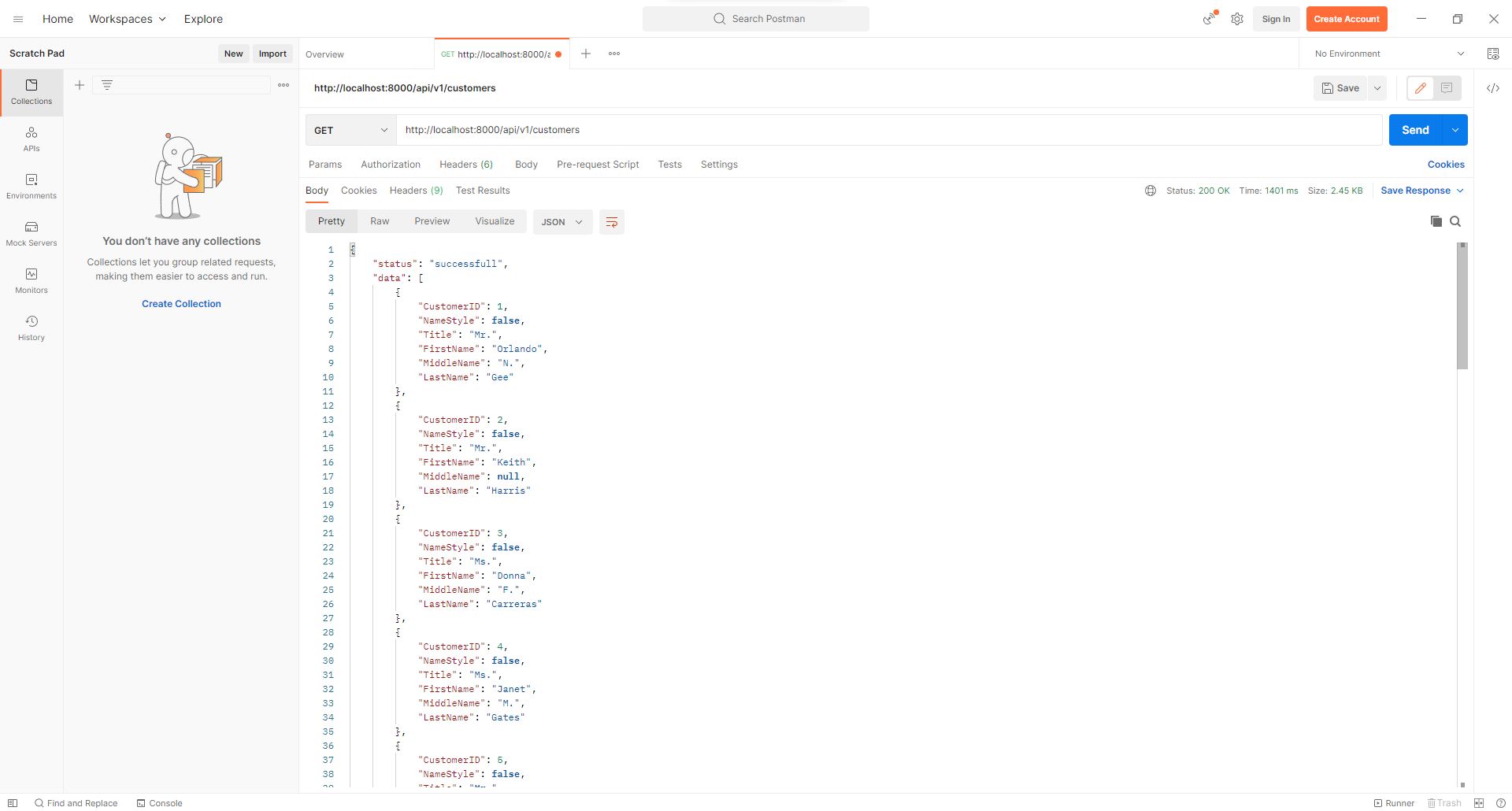
}

* node server.js (Connected to SQL server)



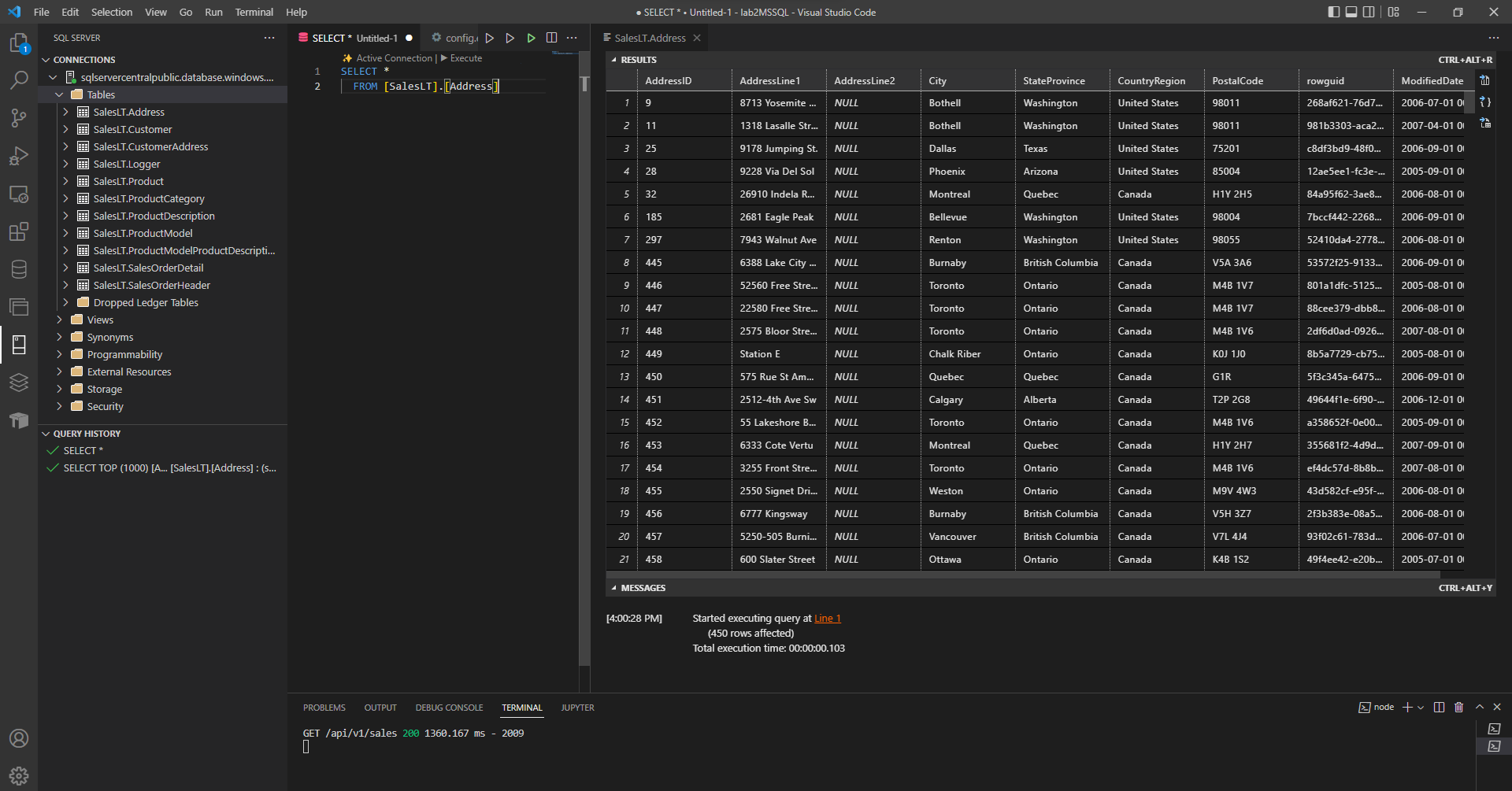
* Getting customers for Route /api/v1/customers





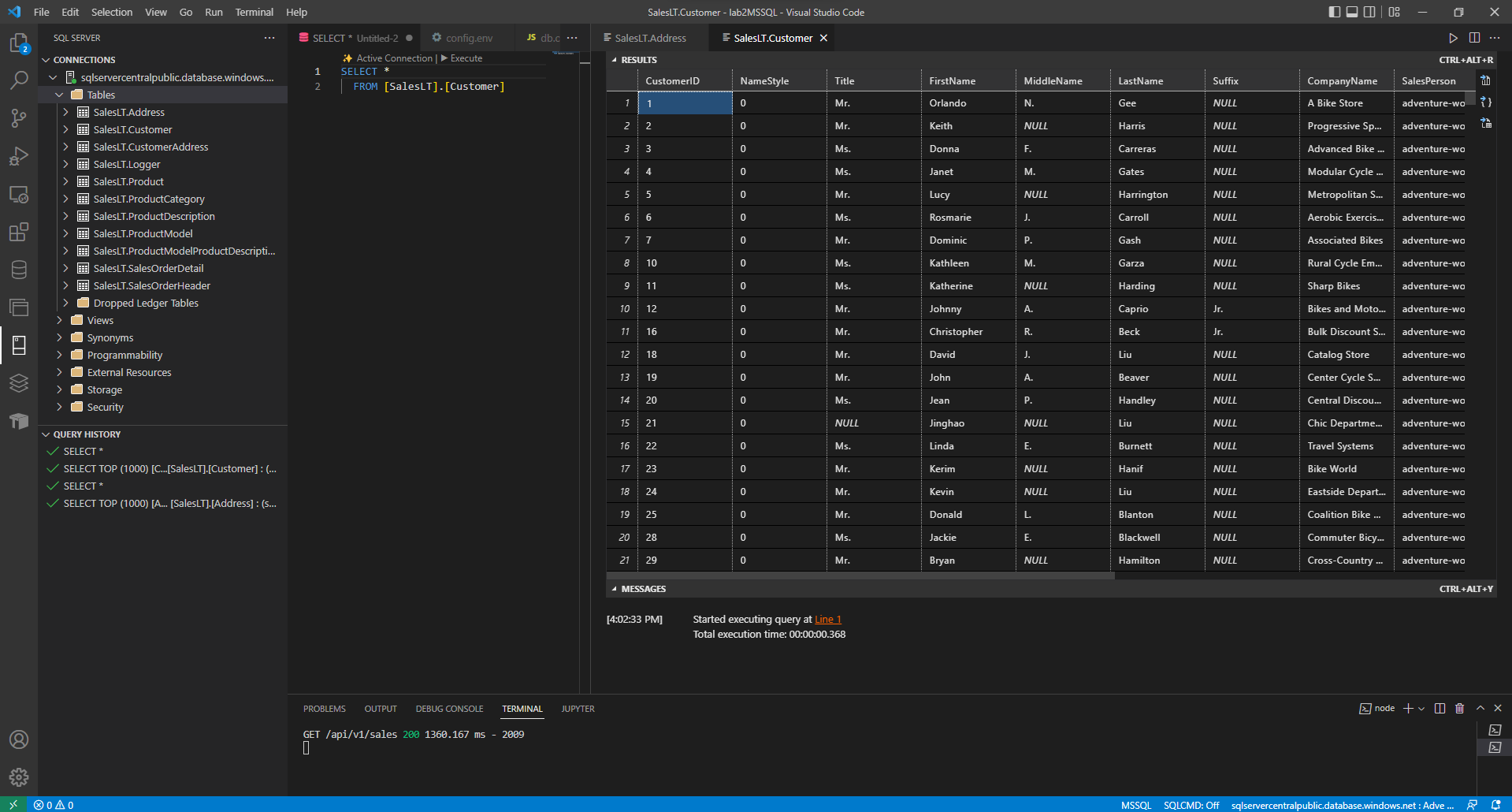
* **SQL SELECT Statement to Query Data in four Tables**
  + SELECT from Address table

SELECT \* FROM [SalesLT].[ Address]



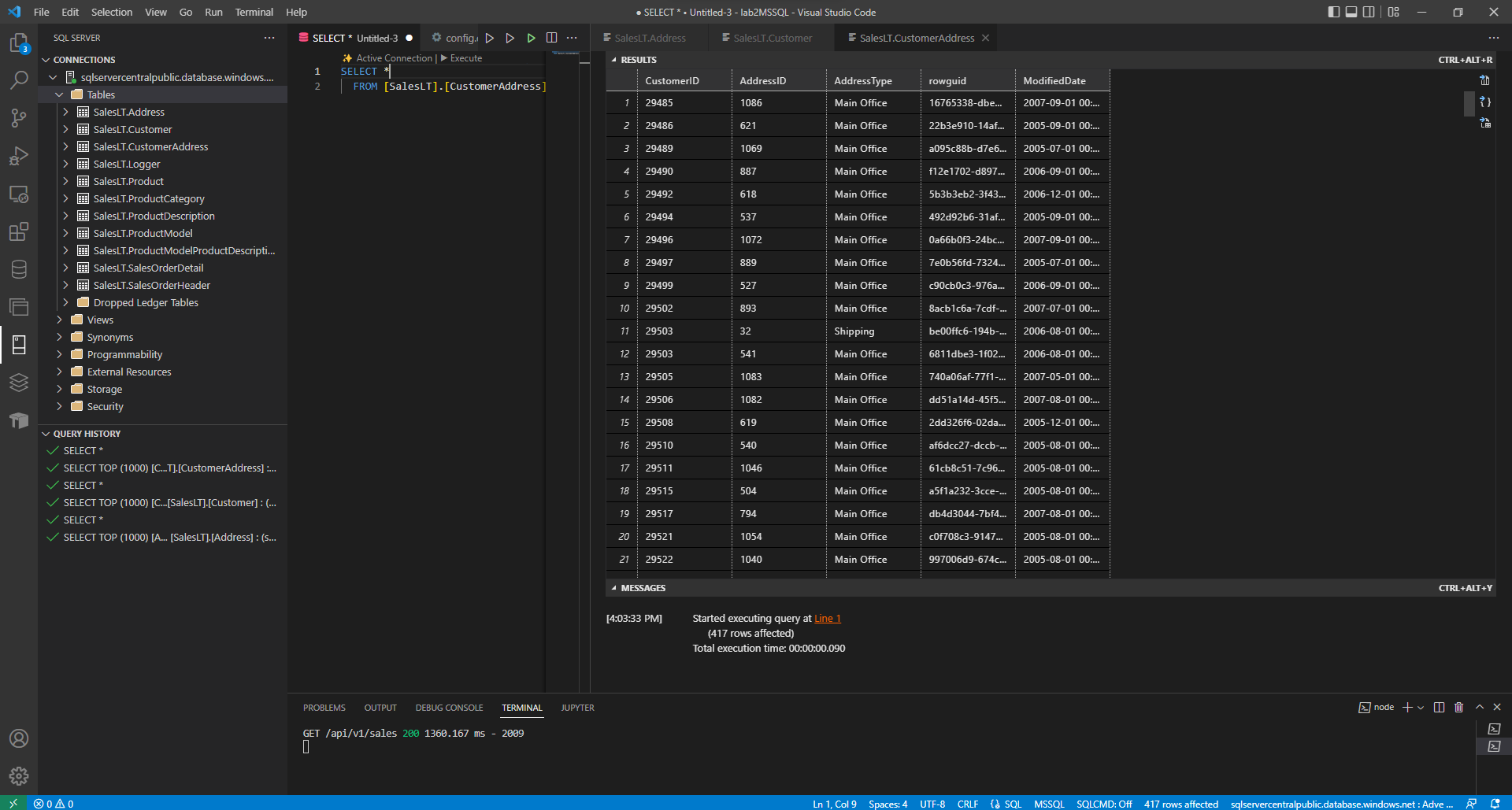
* + SELECT from Customer Table

SELECT \* FROM [SalesLT].[Customer]



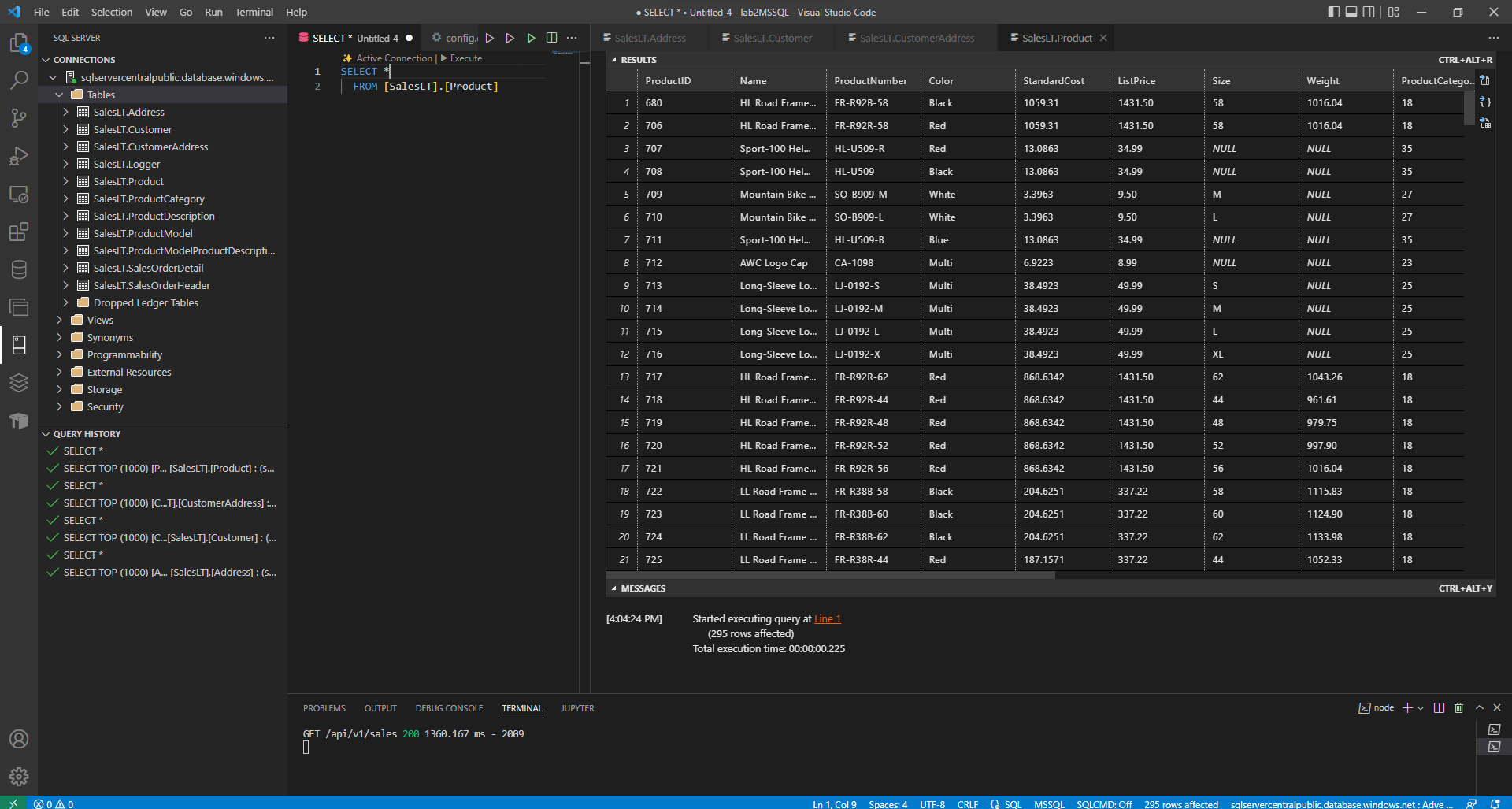
* + SELECT from CustomerAddress Table

SELECT \* FROM [SalesLT].[ CustomerAddress]



* + SELECT from Product Table

SELECT \* FROM [SalesLT].[ Product]



* **Joining two tables:**

SELECT

SalesLT.Customer.FirstName AS full\_name,

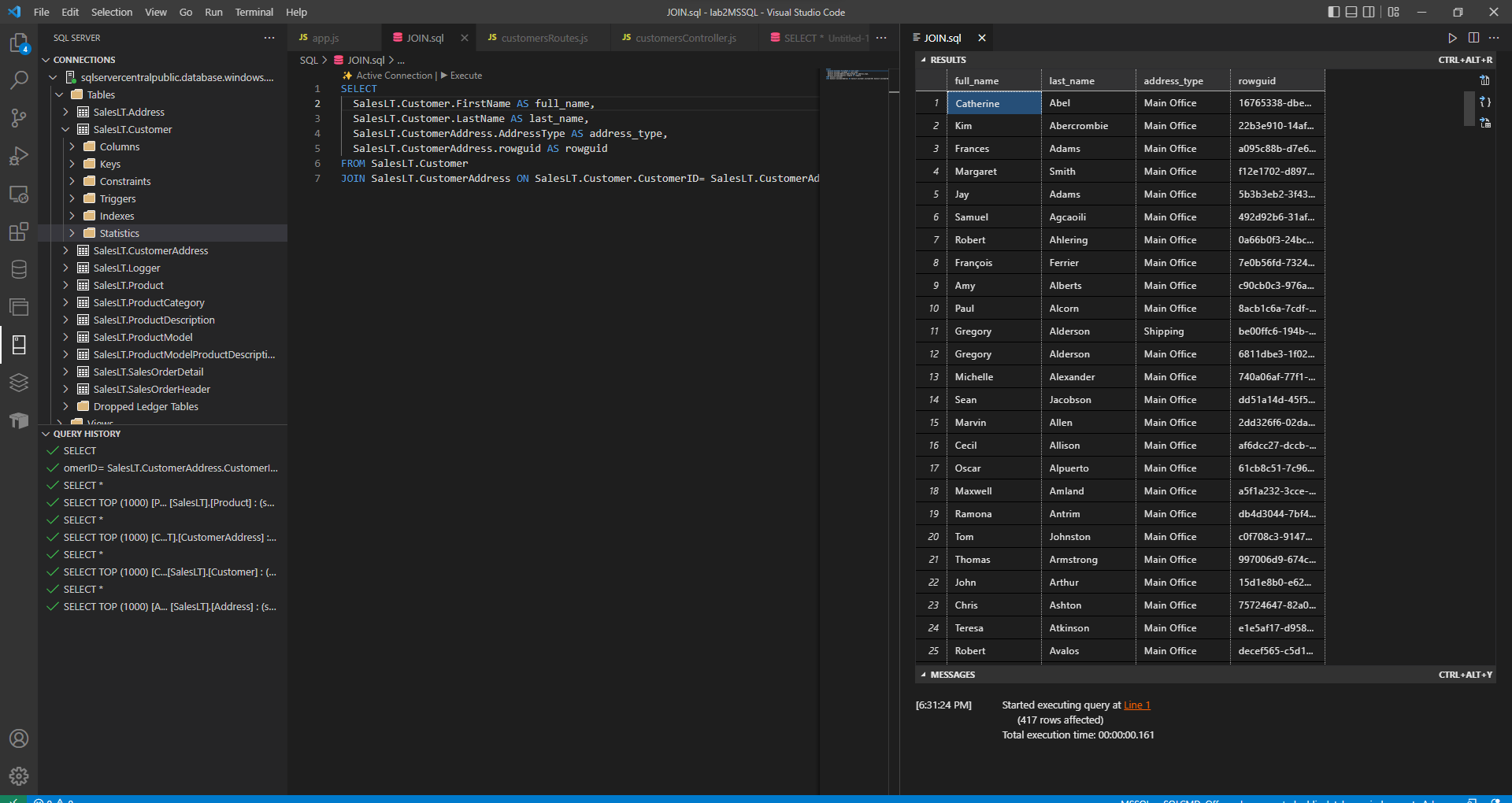
SalesLT.Customer.LastName AS last\_name,

SalesLT.CustomerAddress.AddressType AS address\_type,

SalesLT.CustomerAddress.rowguid AS rowguid

FROM SalesLT.Customer

JOIN SalesLT.CustomerAddress ON SalesLT.Customer.CustomerID= SalesLT.CustomerAddress.CustomerID;



**GITHUB LINK:**

<https://github.com/saimrunaal23/M4-A1>