HA 450 - XS ADVANCED

(NATIVE HANA DEVELOPMENT SPS04)

Ramana Yellapu

IBM | SAP NextGen UX Team

23 May 2020

TOPICS COVERED

OVERVIEW

ARCHITECTURE

TOOLS

CONCEPTS

AUTHORIZATION

TUTORIALS– INITIAL SETUP

TUTORIALS ILLUSTRATIONS -END TO END

REFERENCES

Overview

How it evolved?

1.Load data into SAP Hana- for legacy systems and build Hana web apps [initial target for this landscape]

2.XS engine classic- sp01 sps11 to xs advanced sp04 multi tenant (container based) database

3.To improve performance - using this setup with load balancing we can achieve.

4.Migrate functionality available to implement

5.Cloud based app development hana platform

6.Platform to build for SAP cloud platform/HCP/SCP/Analytics cloud

7.Neo platform to cloud foundry

Why is it Essential?

support extended till 2030 and all services in erp are not yet avaialbe in s4 hana

XS Advanced platform allows to access data for native hana development

Use cases:

1. For Group organization to view all transactions data app for complete overview of all data bases

2. Sap Analytical cloud

3. Sectoral industrial apps- public sector, retail

4. MTA Based (Mutli Target Applications) , below scenario depicts how XS Advanced platform fits in as a HANA System.

Application Development in SAP HANA XS Advanced Model

Two approaches for development

Cloud Foundry - cloud environment

XS Advanced - onpremise

ARCHITECTURE

Architecture Overview

As illustrated in the following diagram, the basic system architecture has a classic 3-tier approach:

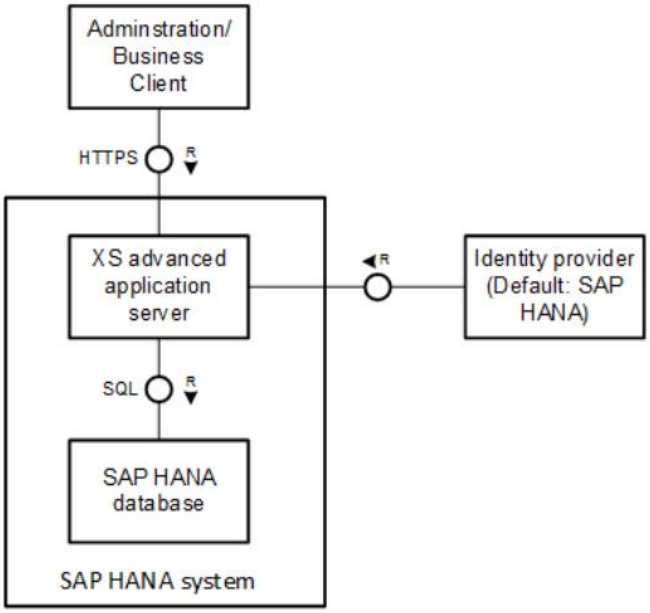


Figure 3: 3-Tier Architecture of SAP HANA with XS Advanced

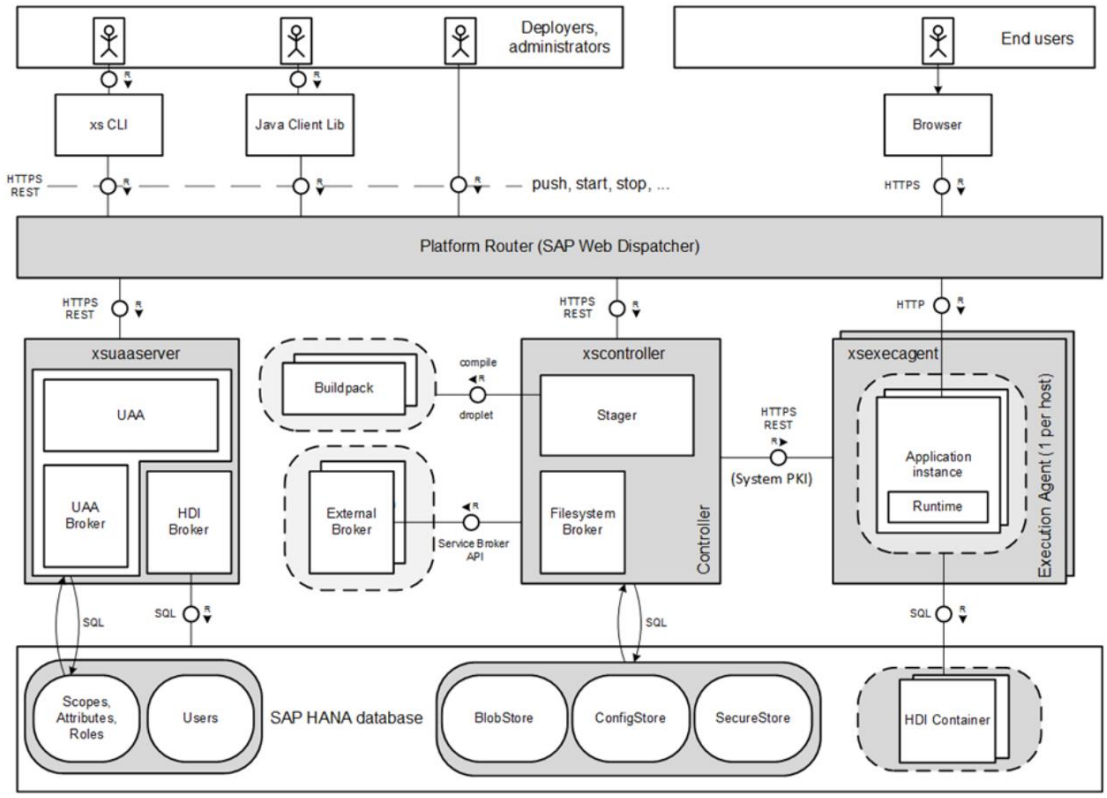


Figure 4: Technical System Landscape of XS Advanced Application Server

Tools

Typical Development landscape in XS Advanced :

Front end - using webide – UI5/HTML5/SAP FIORI

Application server- using Fullstack webide – node.js based [python, java can also be used]

Database layer- using Hana studio - SQL

Evolution of Development tools [All 3 ways are available]

1. Hana Studio - stored in Hana package
2. Hana workbench - sap webide - gets stored in hd container
3. Hana XS Advanced based webide - hana development infrastructure [Recommended]

Can we run on personal laptop?

1. SAP provides SCP platform -cloud foundry(30 days trial) Link
2. To install own environment on personal laptop (16 GB Ram configuration required. Download -> support.spa.com/swdc

You can use **Hana Express edition** if ram > 16 gb on personal laptop – [version supported sp04]

Hana classic can be run for less than 16gb as well [version sp01 only]

CONCEPTS

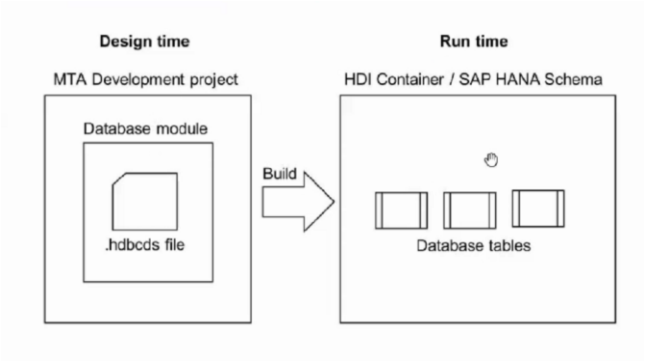
Following concepts are dealt in this topic.

- CDS
- XS Classic vs XS Advanced
- XSODATA
- Node.JS
- HANA Database

CDS:

Two types of CDS are used in SAP Application development.

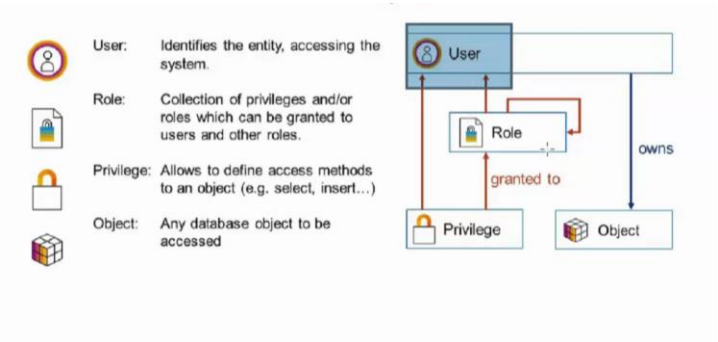
1. ABAP CDS - (ABAP & S/4 HANA- VDM) ,DDL Views.
2. Native CDS – HANA Development . Here we deal with Native CDS, which is used for XS Application development.

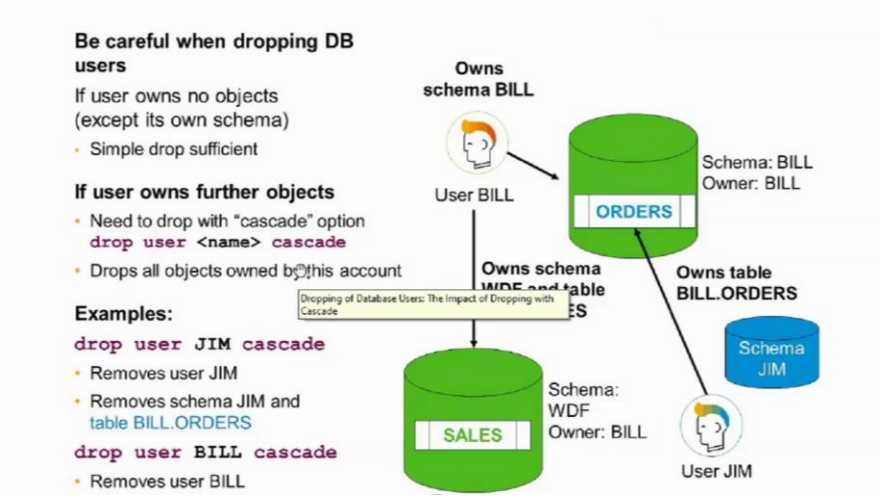


XS Classic and XS Advanced key difference-

XSA primarily has **HDI Container** which provides efficent persistence in handling database authorization based confilicts.

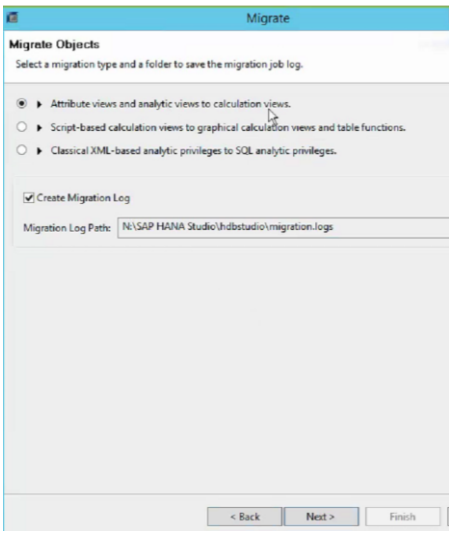
How HDI makes difference? Below illustration provides a usecase to explain the type of confilicts addressed.



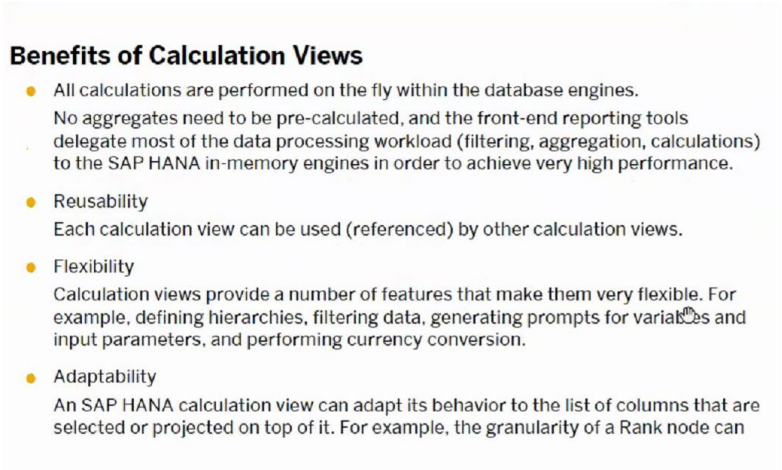



```
Since HANA released:

HANA Studio:
        Attribute VIew          - Join Engine           (Master Data)   1-D
        Analytical View         - OLAP Engine           (Star join / Star schema)       Analytics
                                                        2-D view
        Calculation View        - CV Engine             Multi - Dimensional

1.0 SPS9 onwards,

        SAP optimized the CV engie to do everything using CV


        Calculation VIew        - Sub types

                Dimension               (= Attribute_
                Cube with Start Join    (=Analytical View

With THE SAP HANA 2.0 SPS## versions,
we create calcaulation views in Web IDE
```

- All Hana studio based developments needs to be migrated to Calculation views. In Hand studio option to migrate extension is available to address this.

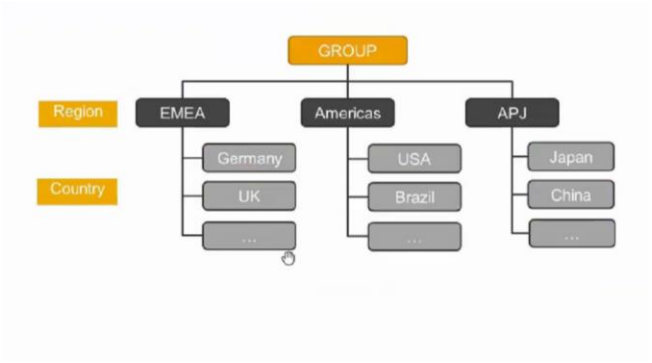


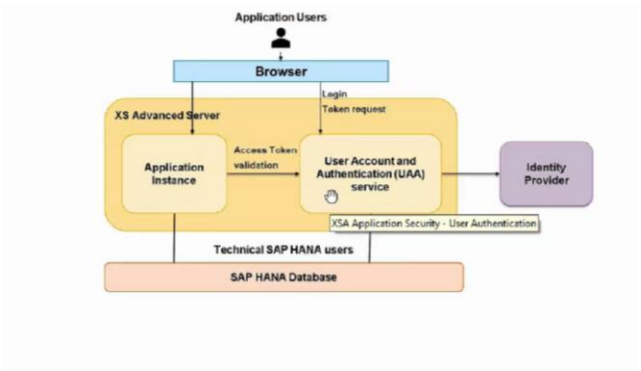
Essentials SP04 onwards all developments are Calculation views based and are available in HDI Container.

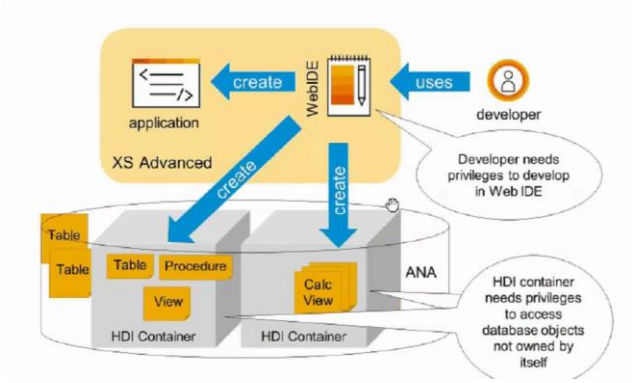


Authorization

Handling Authorization landscape scenarios pictorial representation .







Authorization handling in multi tenant based data handling- Illustration

User	Schema		Project		User Provioning
STUDENT09	STUDENT09		Project_1		
SLTUSER	SLT_SCHEMA	Created	Wants to access data from SLT_SCHEMA		SLT_SCHMA_CROSS_ACCESS
SAPABAp1	SAPABAP1				S_4_Access
			Resources		
			CROSS SCHEMA	Maintain value. 1. SLT ; 2. S4	
			Module	Map the correct pointers	
			Create a new hdbgrants	SELECT, CREATE ANY, etc	
			Create a new synionym (connections- alias) -> table from CROSS Schema		
			Create a new CDS Views / Calculation Views based on Synonymn		

Tutorials Initial Setup

Using the XS Advanced Command Line Interface

1. Use power shell.

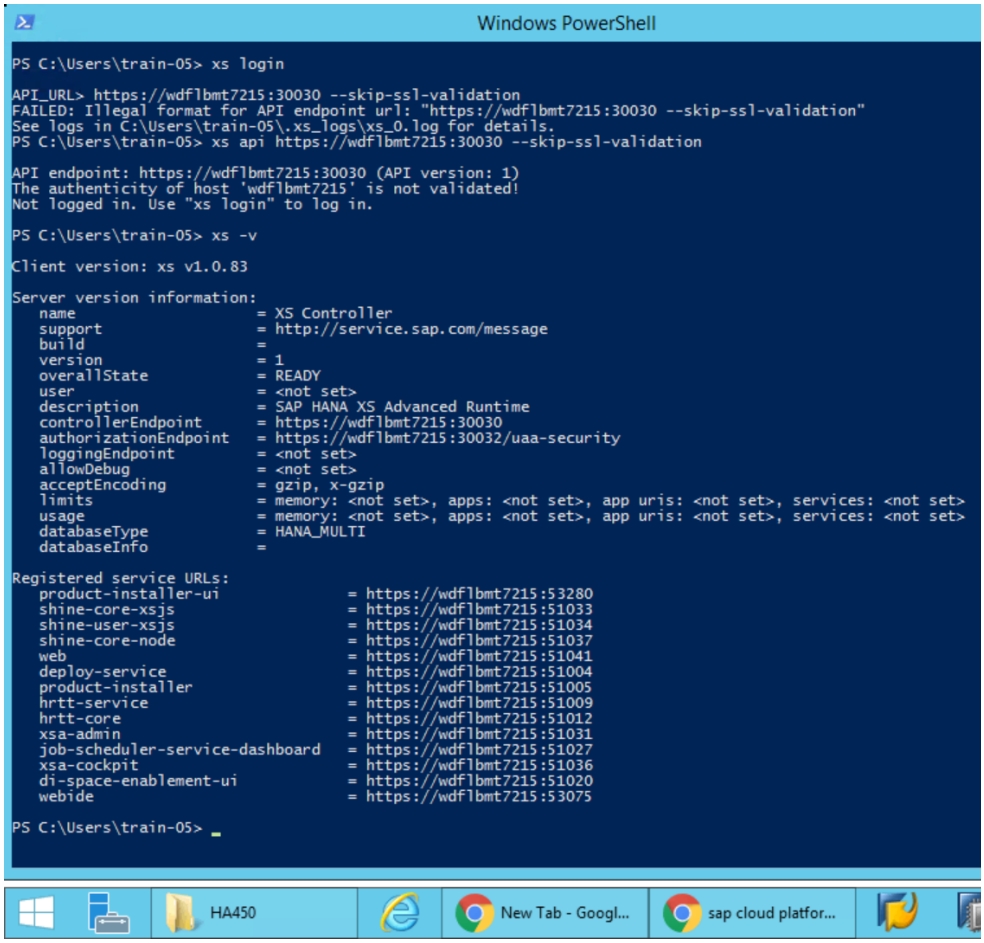
xs login

if it asks for api use below command

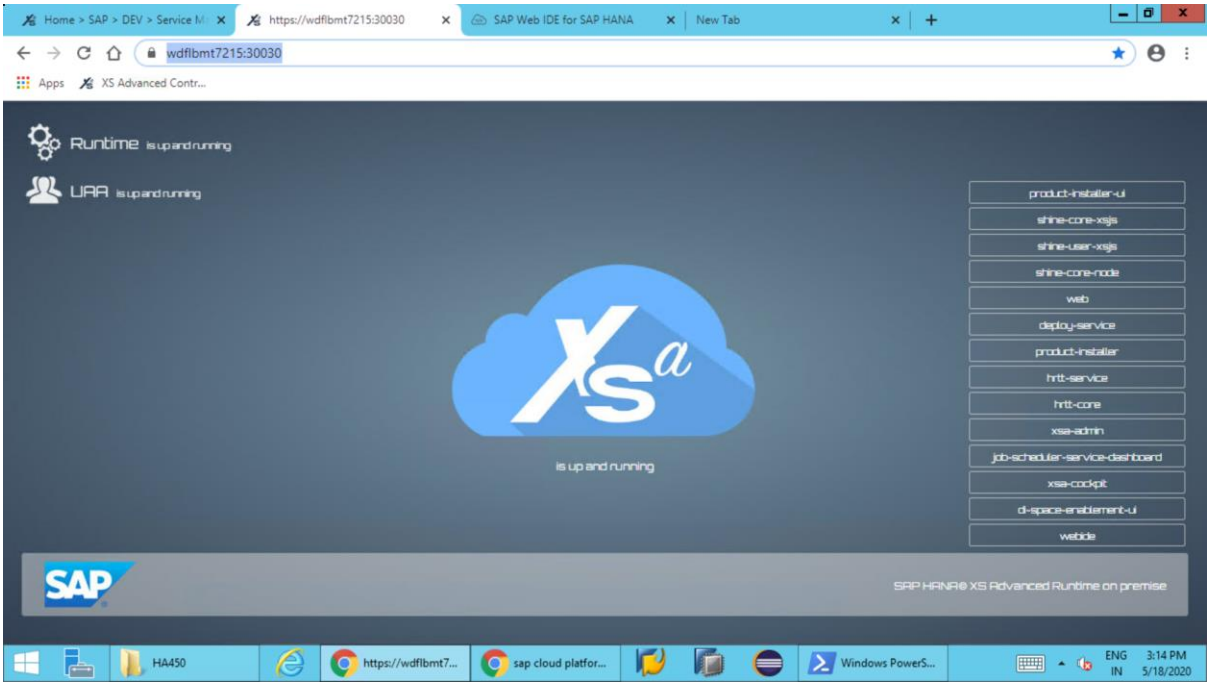
xs api

xs -v

xs v1.0.83



2. Open Chrome browser. Copy the controller end point url as shown below. It allows to access all related tools related to XS Advanced.
   https://wdflbmt7215:30030/

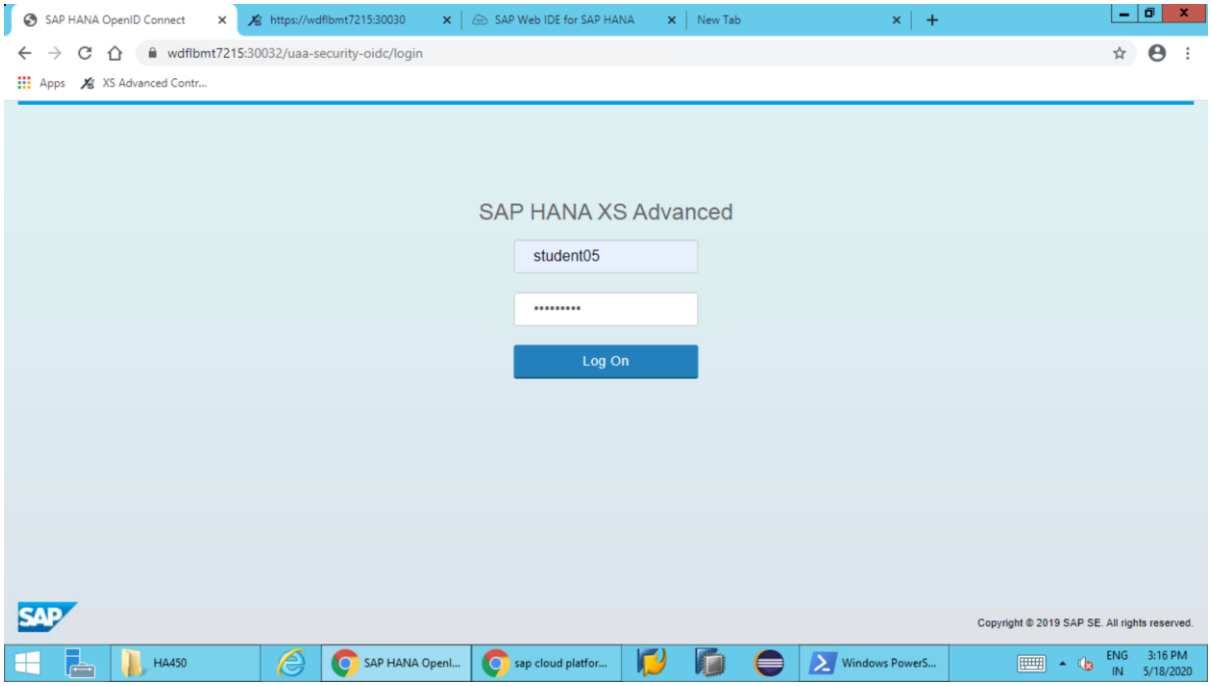


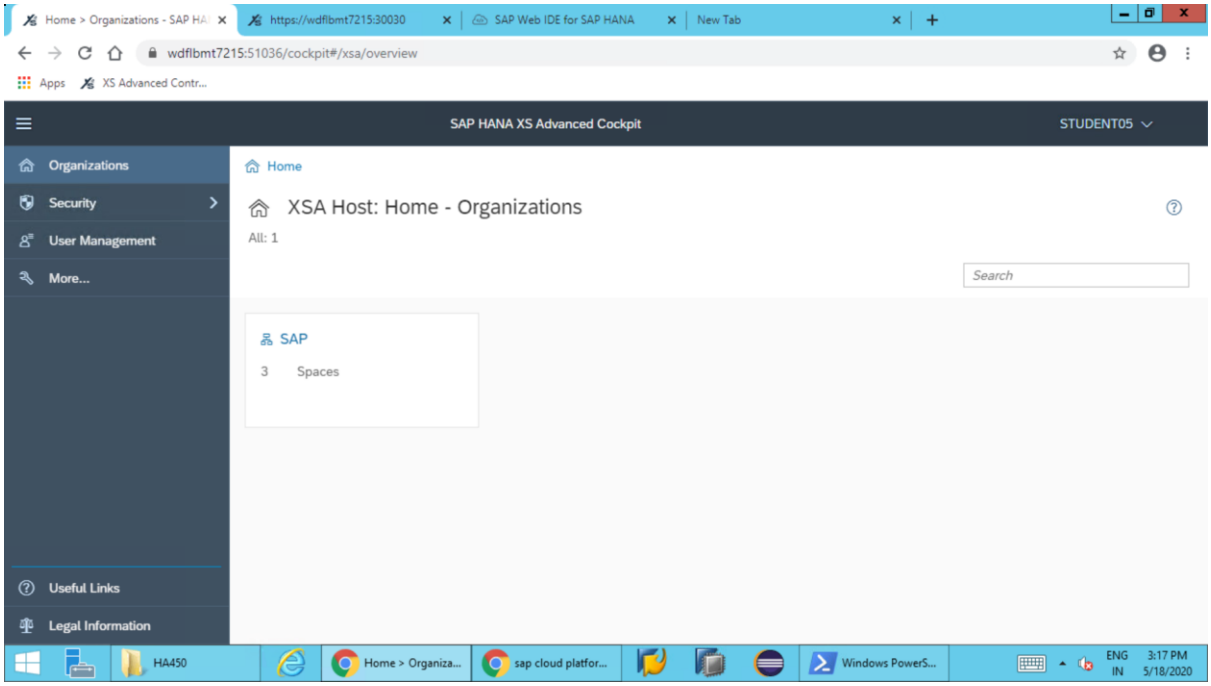
3. Select webide and xsa-cockpit – these are the XS Advanced on premise tools for development and monitoring.

XS Advanced Cockpit https://wdflbmt7215:30032/uaa-security-oidc/login

Useid: student05

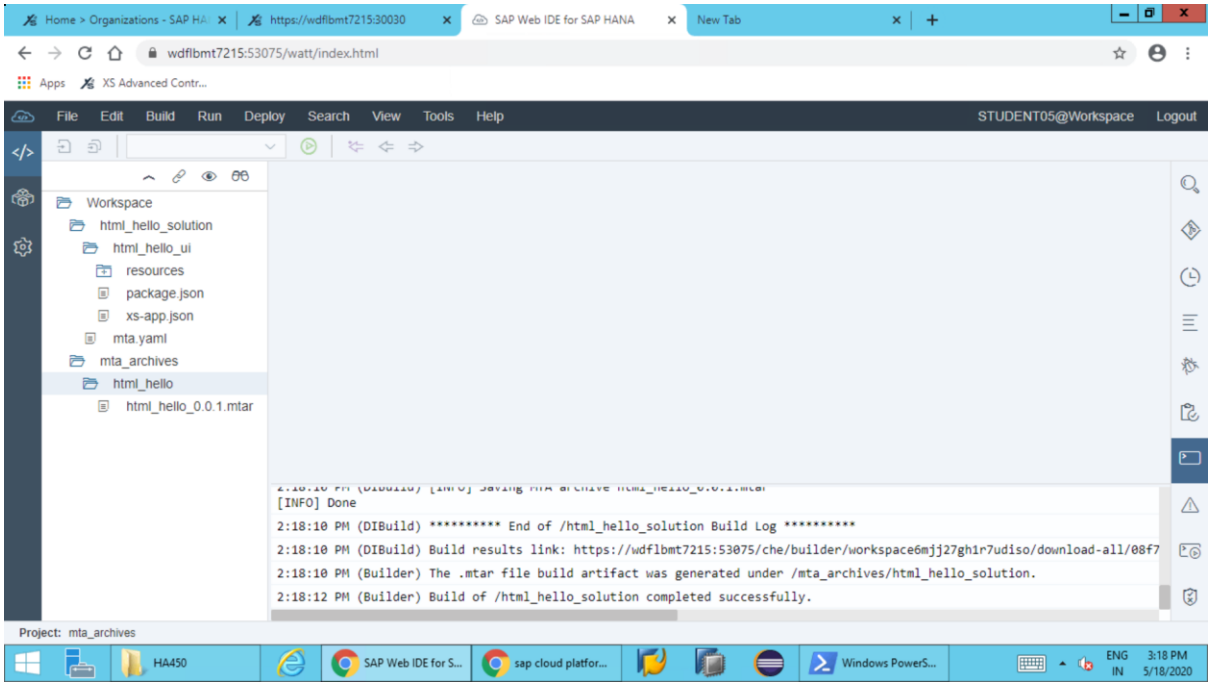
Pwd: Training1





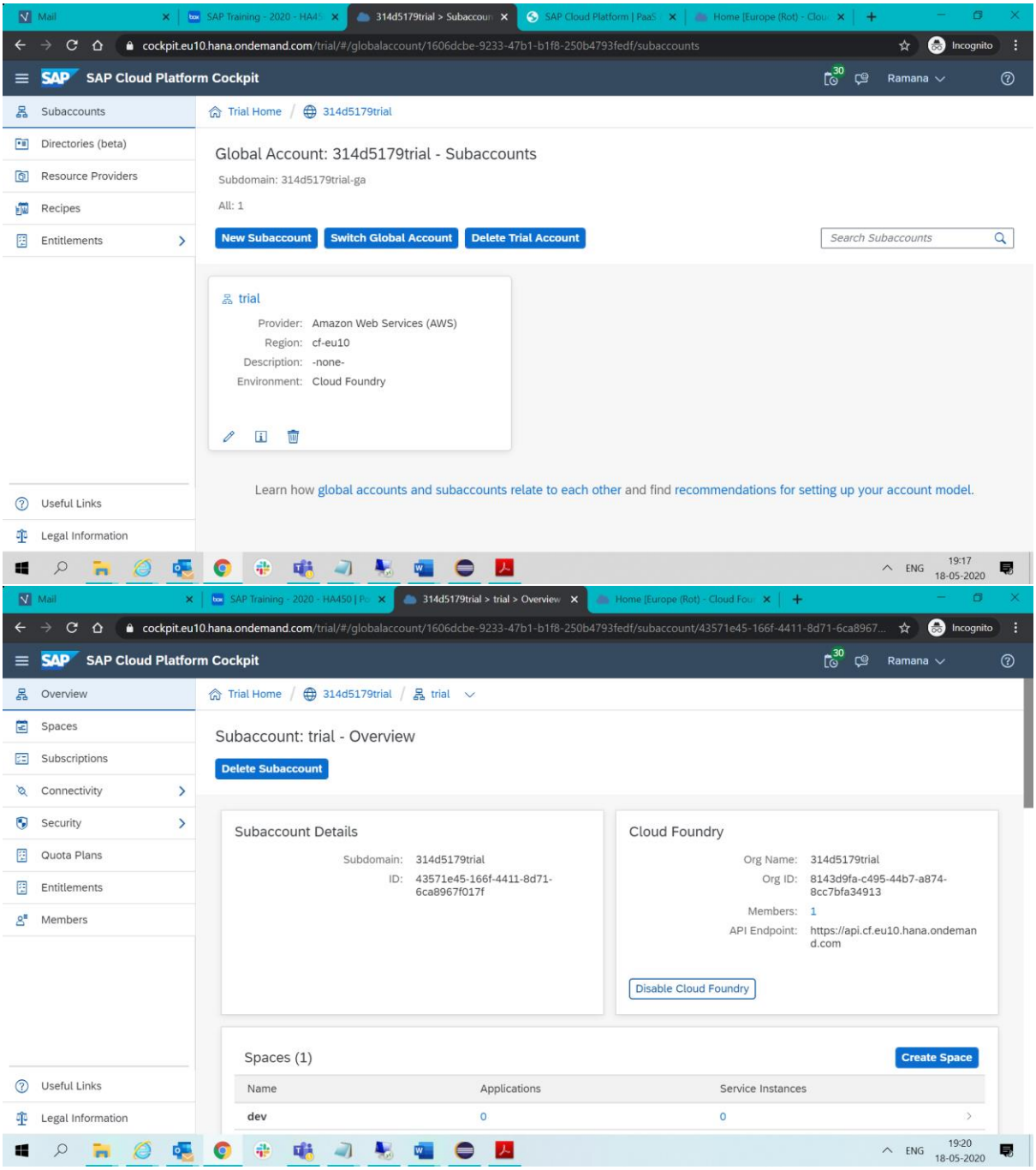
Webide:

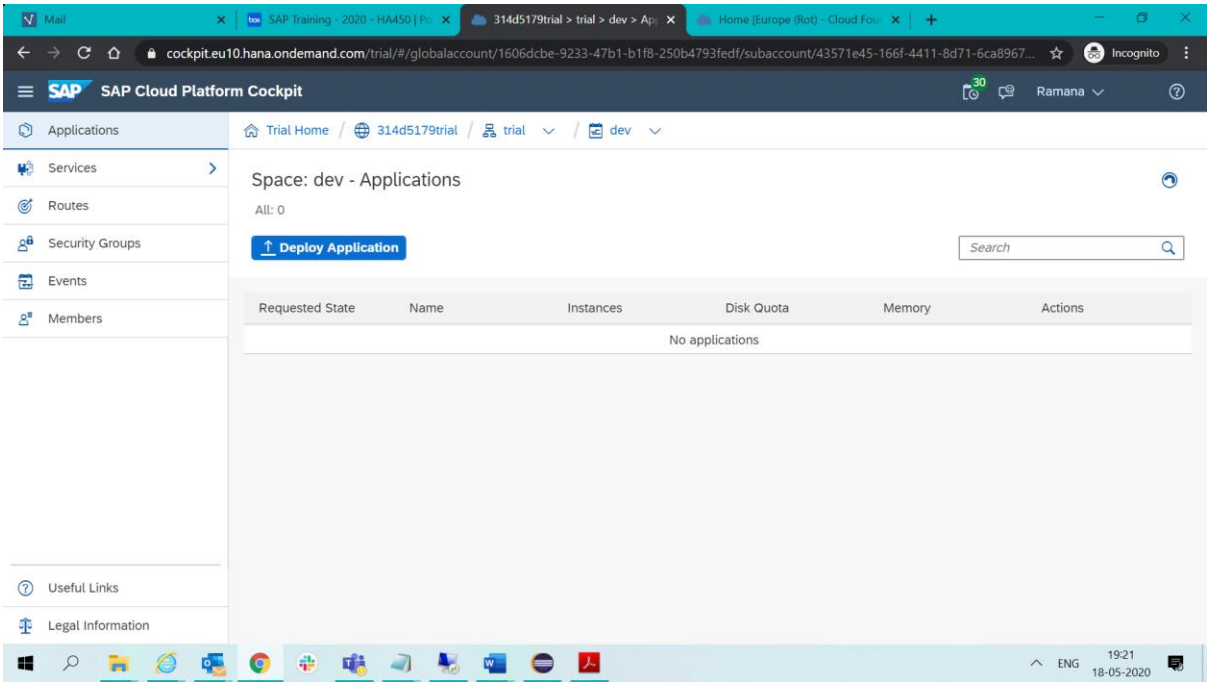
https://wdflbmt7215:53075/watt/index.html



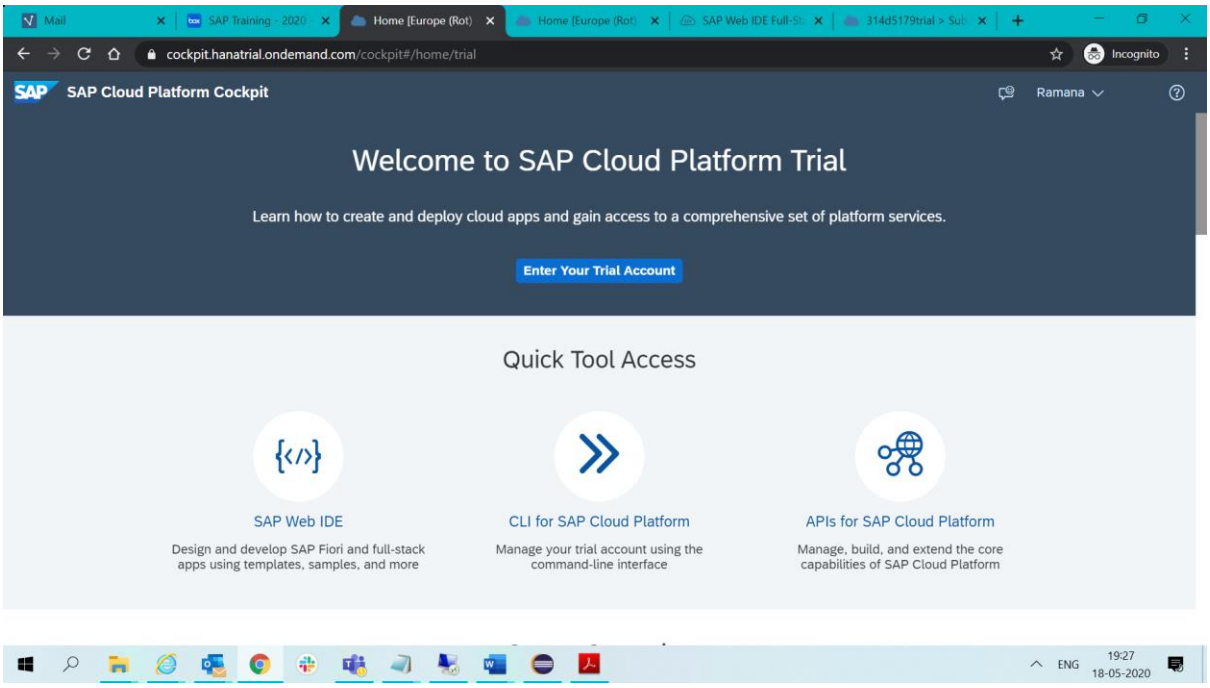
4. Now create a trial account to access cloud foundry trial account. Valid for 30 days. It takes a while to create new account. If already have account and if its expired use new phone number.

   https://cockpit.hanatrial.ondemand.com/cockpit#/home/trial


SAP Cloud Platform Cockpit
Subaccounts
Directories (beta)
Resource Providers
Recipes
Entitlements
Useful Links
Legal Information
Trial Home / 314d5179trial
Global Account: 314d5179trial - Subaccounts
Subdomain: 314d5179trial-ga
All: 1
New Subaccount
Switch Global Account
Delete Trial Account
Search Subaccounts
trial
Provider: Amazon Web Services (AWS)
Region: cf-eu10
Description: -none-
Environment: Cloud Foundry
Learn how global accounts and subaccounts relate to each other and find recommendations for setting up your account model.
SAP Cloud Platform Cockpit
Overview
Spaces
Subscriptions
Connectivity
Security
Quota Plans
Entitlements
Members
Useful Links
Legal Information
Trial Home / 314d5179trial / trial
Subaccount: trial - Overview
Delete Subaccount
Subaccount Details
Subdomain: 314d5179trial
ID: 43571e45-166f-4411-8d71-6ca8967f017f
Cloud Foundry
Org Name: 314d5179trial
Org ID: 8143d9fa-c495-44b7-a874-8cc7bfa34913
Members: 1
API Endpoint: https://api.cf.eu10.hana.ondemand.com
Disable Cloud Foundry
Spaces (1)
Create Space
Name
Applications
Service Instances
dev
0
0



Goto Home screen
https://cockpit.hanatrial.ondemand.com/cockpit#/home/trial



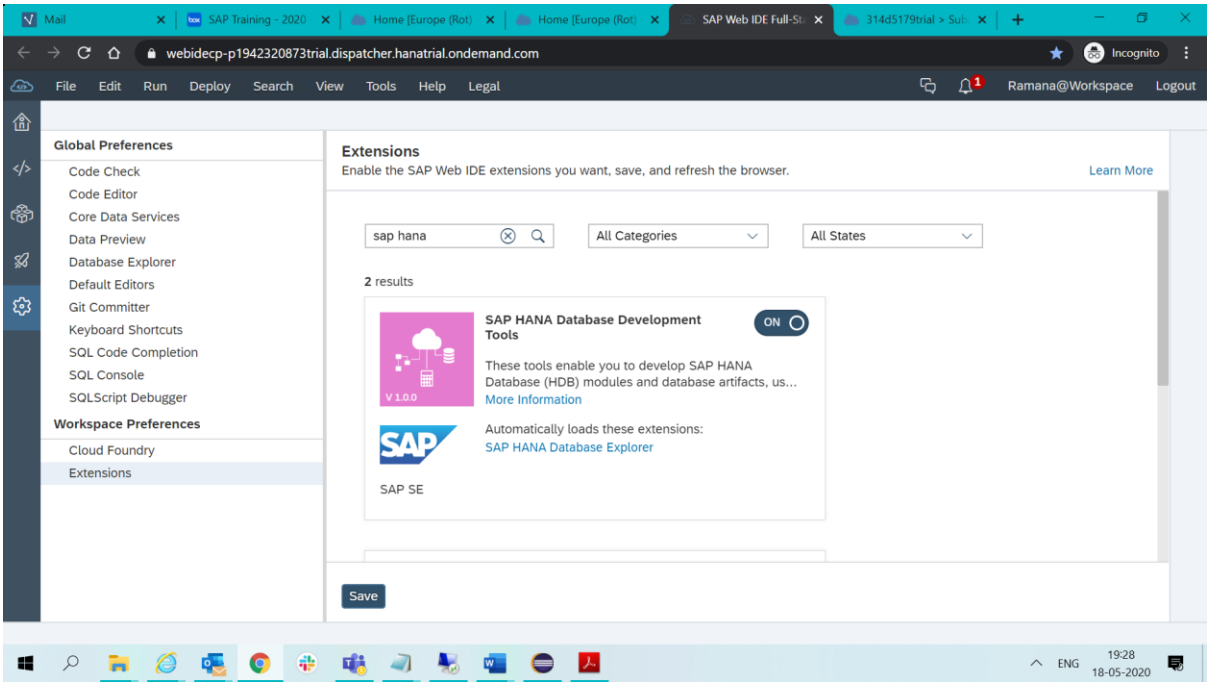
And select webide url and bookmark it.

https://webidecp-p1942320873trial.dispatcher.hanatrial.ondemand.com/

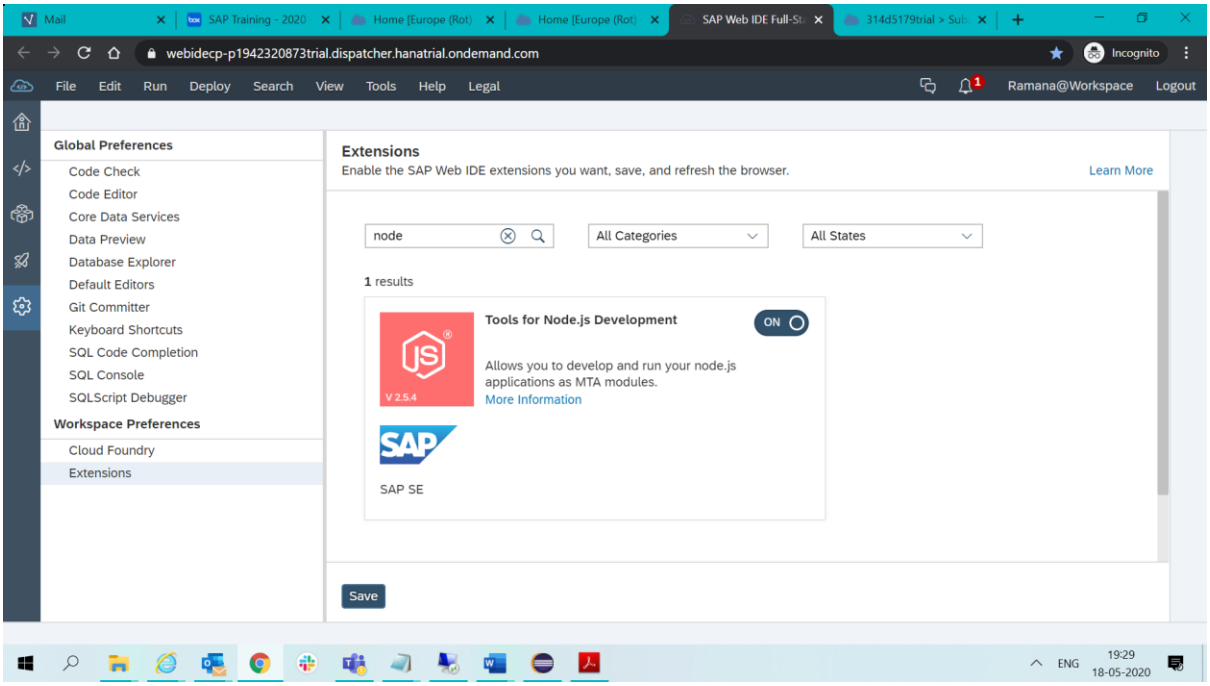
Now goto Settings and Configure below extensions and select turn on

Go to preferences → Extensions.

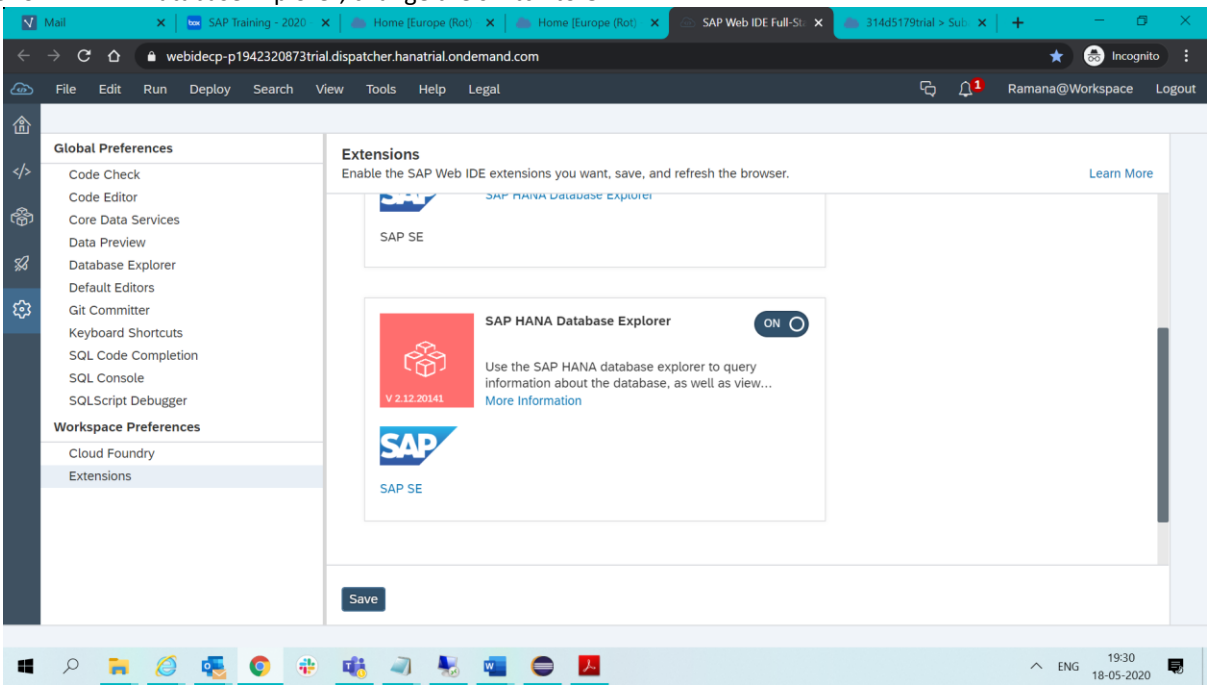
Search for Tools for SAP HANA Database Development, change the switch to On.



Search for Tools for Node.js Development, change the switch to On.



Search for SAP HANA Database Explorer, change the switch to On.



Don’t forget to select Save . After opting for addons.

5. Now you can Use Exercise pdf to follow tutorial instructions to do it yourself. If you wish to have complete overview refer next steps in the tutorials

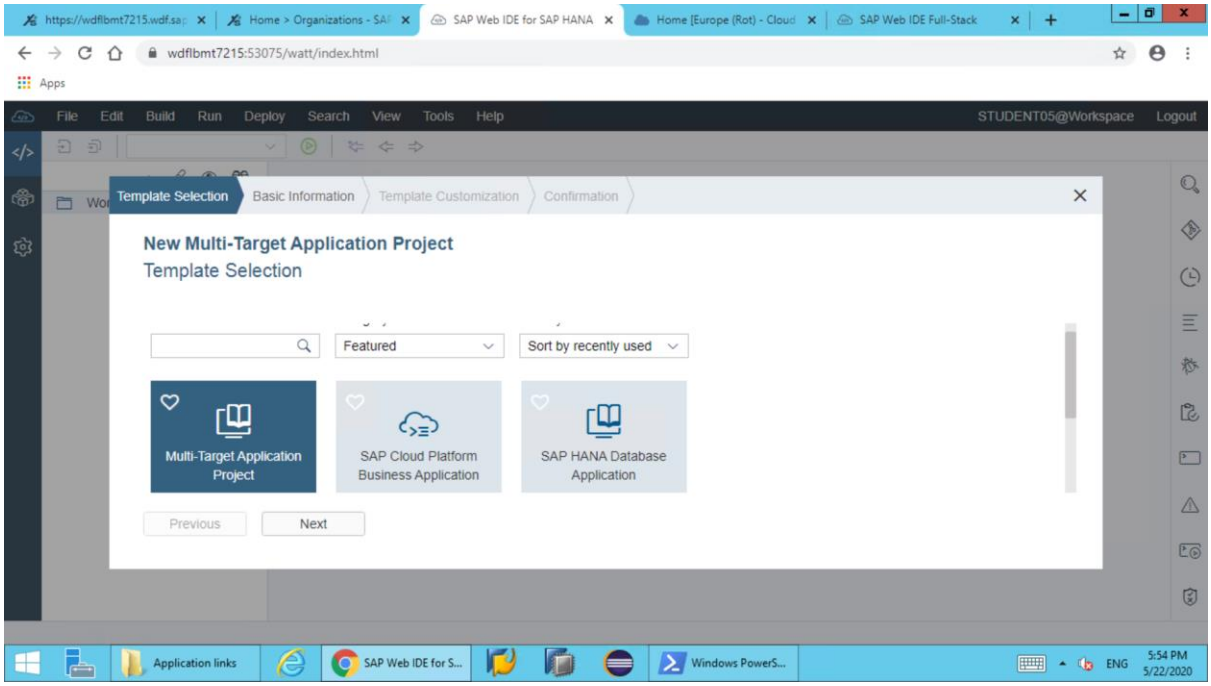
02.HA450 Exercises.pdf

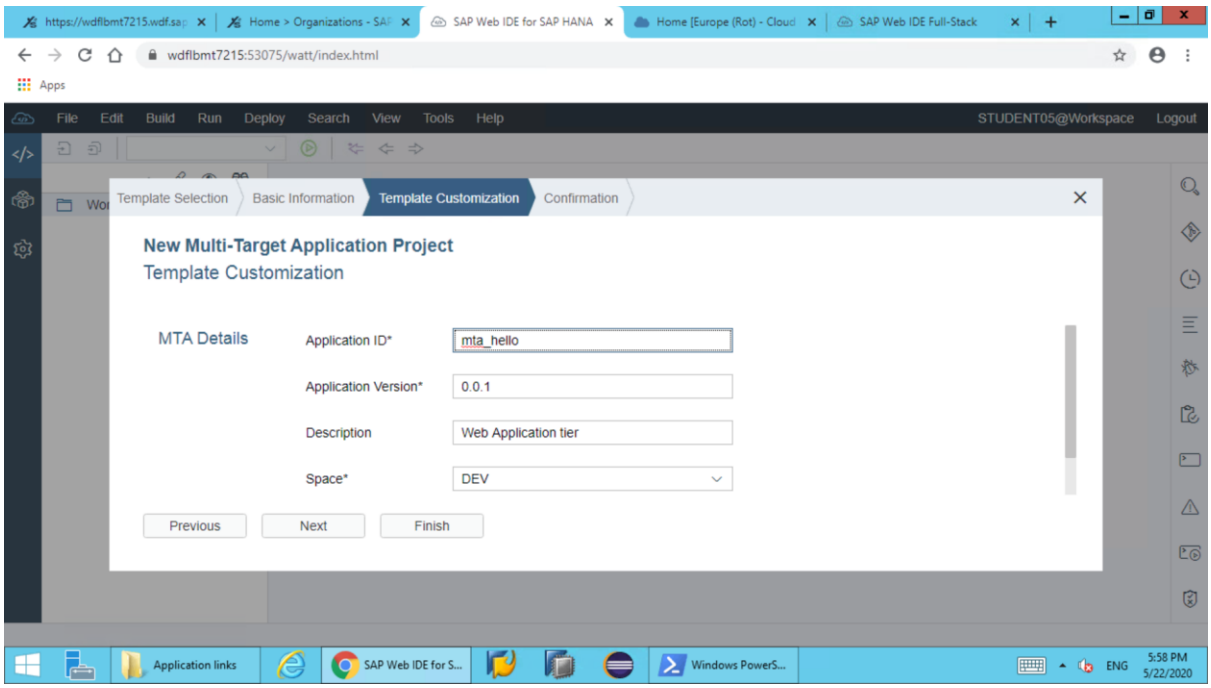
WHAT IS MTA ?

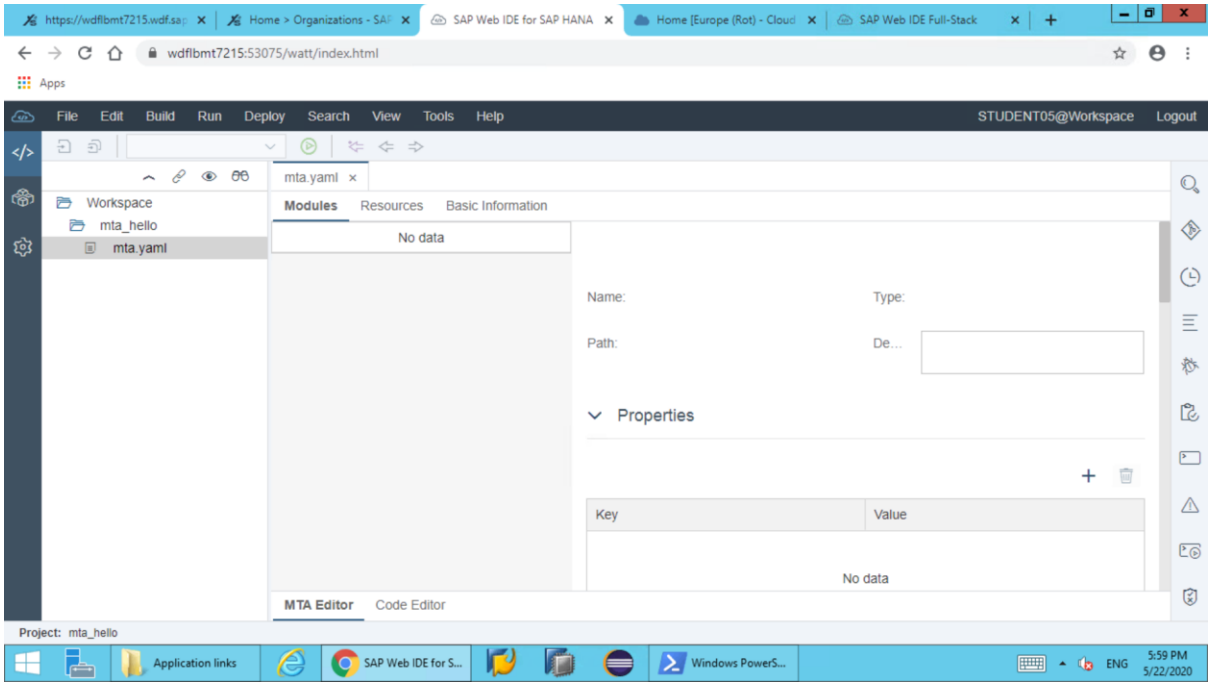
MTA - MULTI TARGET APPLICATION are comprised of multiple software modules representing the data, business logic and UI tiers . These modules are created with different technologies, yet share the same development lifecycle.

Getting Started with MTA

1. Create new MTA project



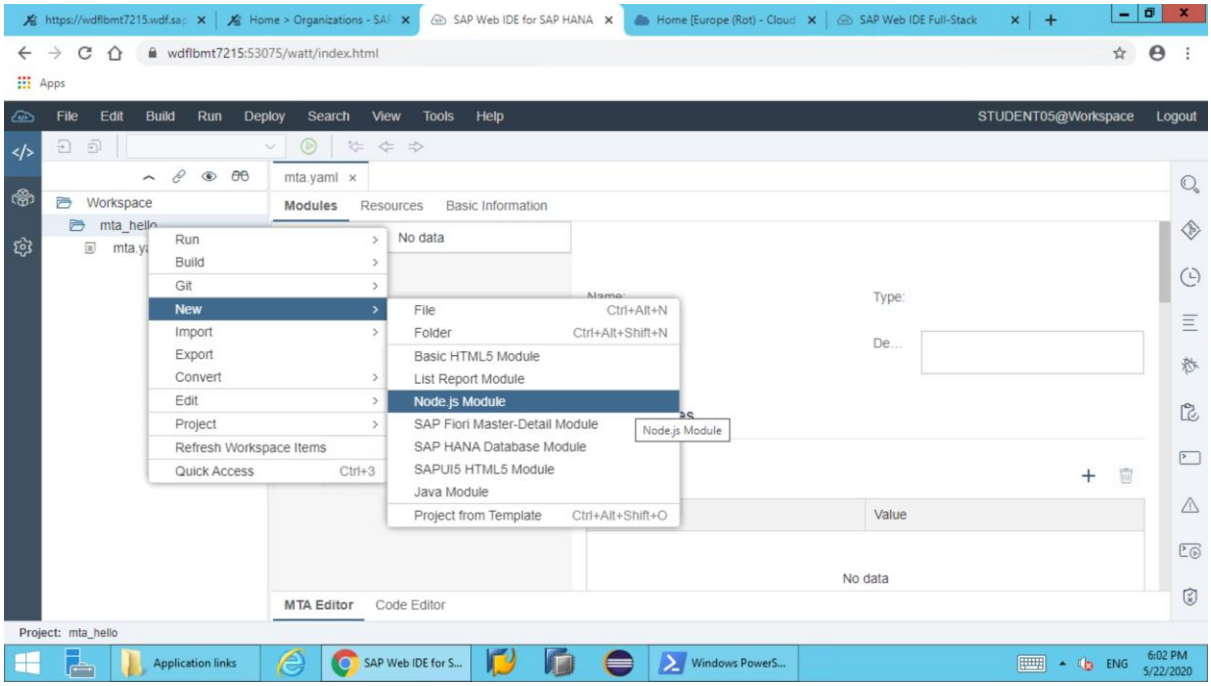


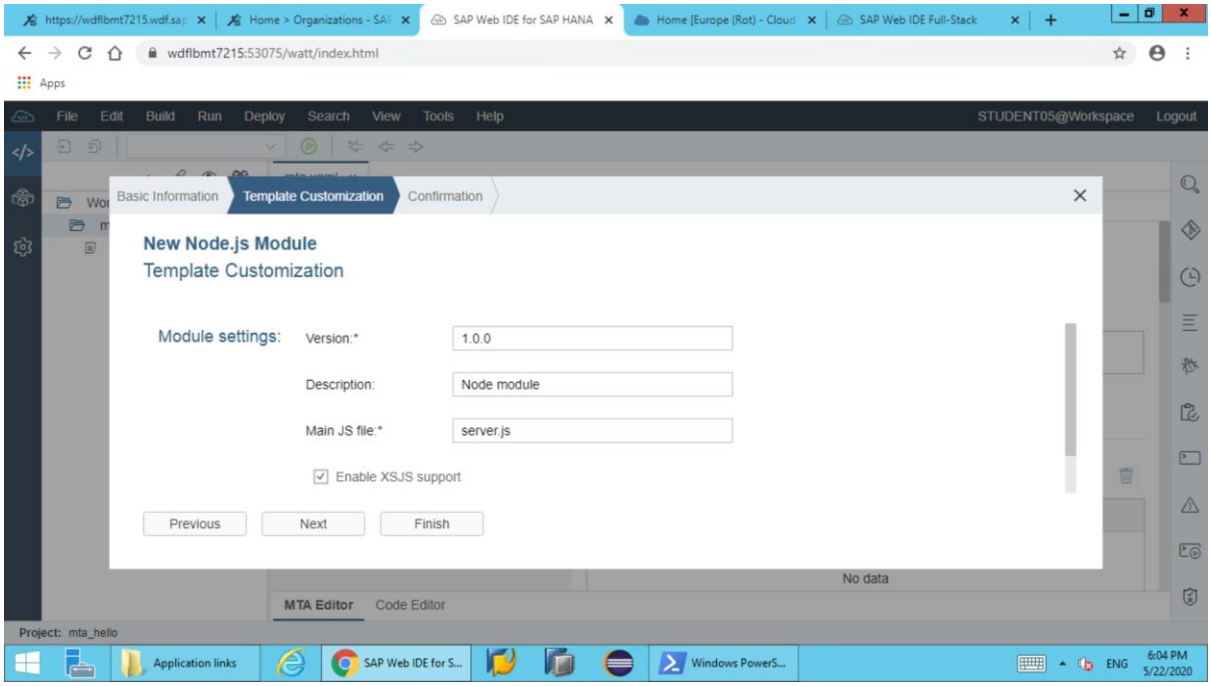


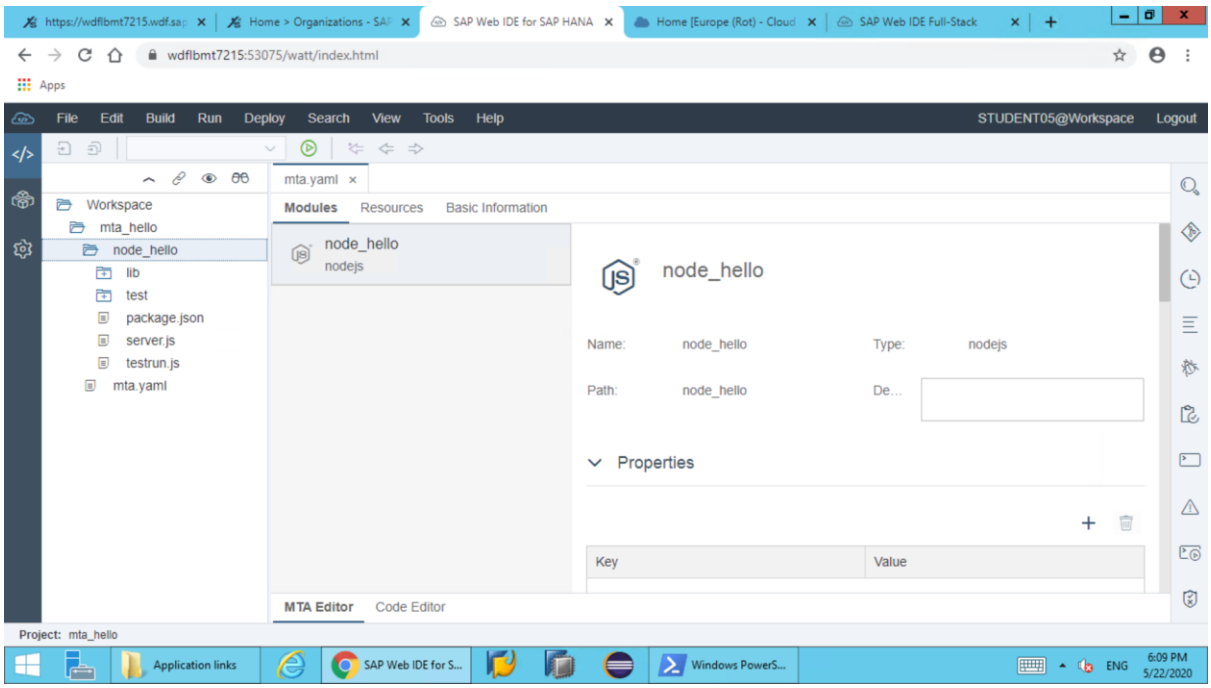
Now we can have multiple modules in MTA based projects.

1. SAP UI5/SAP Fiori/List Report /HTML5/Java module for UI Tier
2. Node.js modules for Application server tier
3. Hana database modules for Database tier

In this section , we are using Node.js module.







Delete all the files under node_hello.

And add new package.json and server.js files.

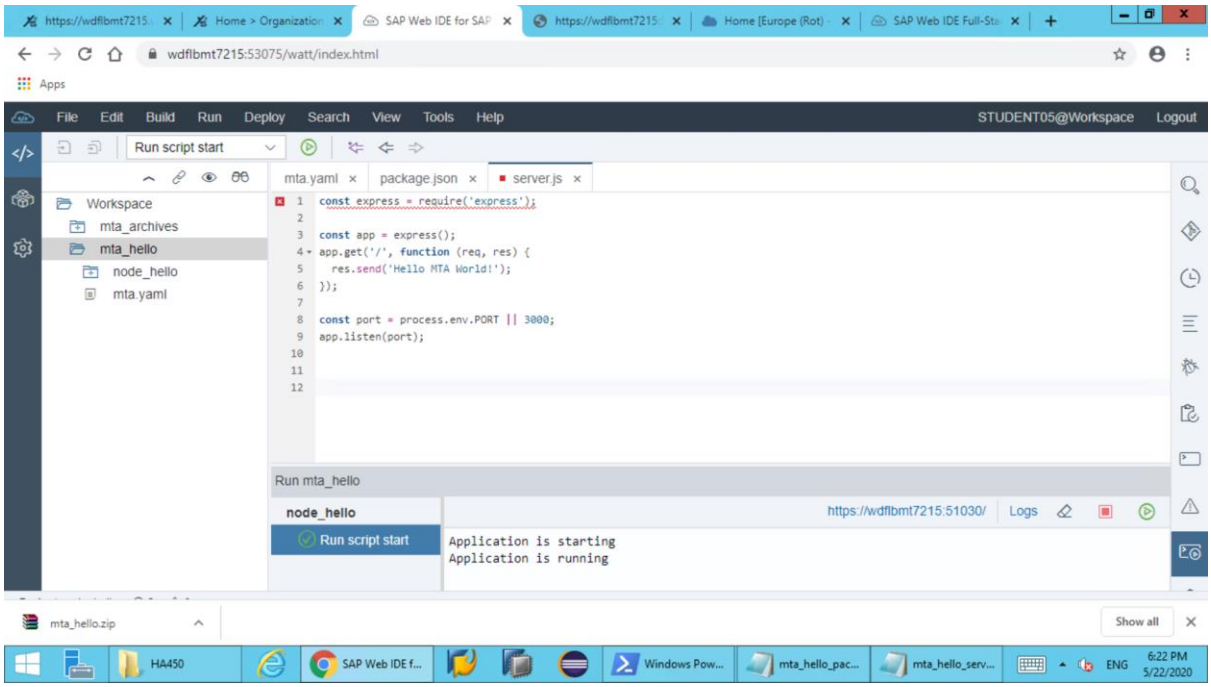
Pakage.json

```
{
        "name": "node_hello",
        "version": "1.0.0",
        "dependencies": {
```

```
        "express": "^4.0.0"
    },
    "scripts": {
        "start": "node server.js"
    },
    "engines": {
        "node": "10.x"
    }
}
```

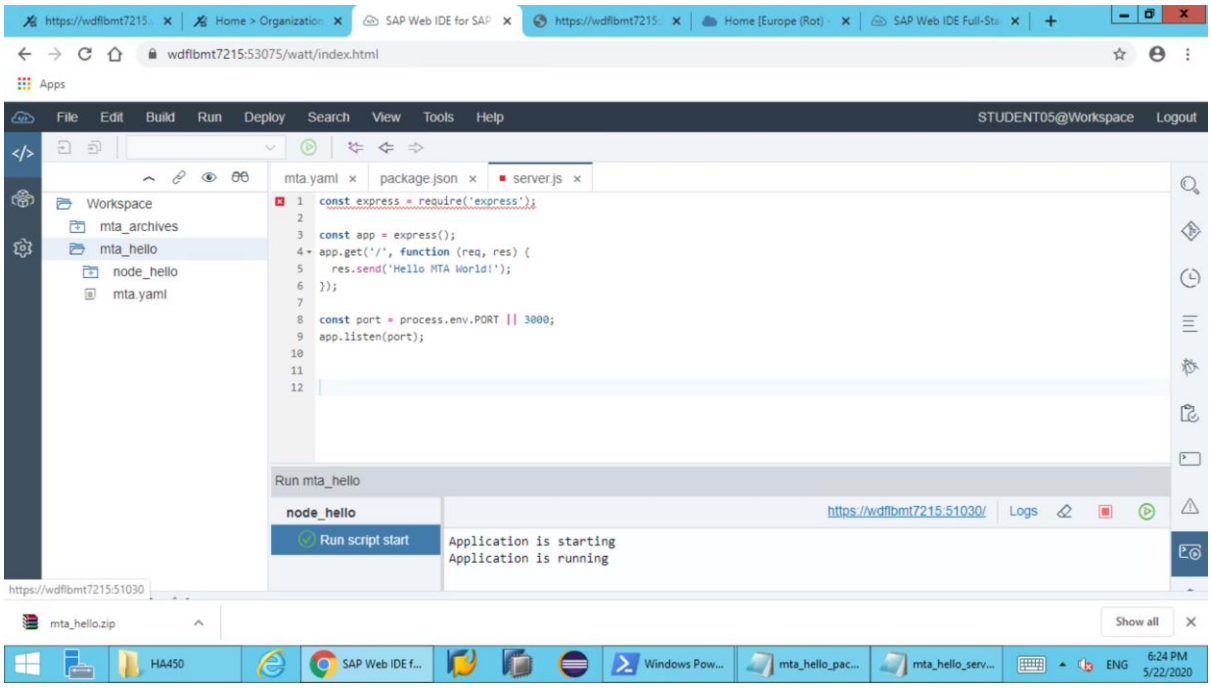
Server.js

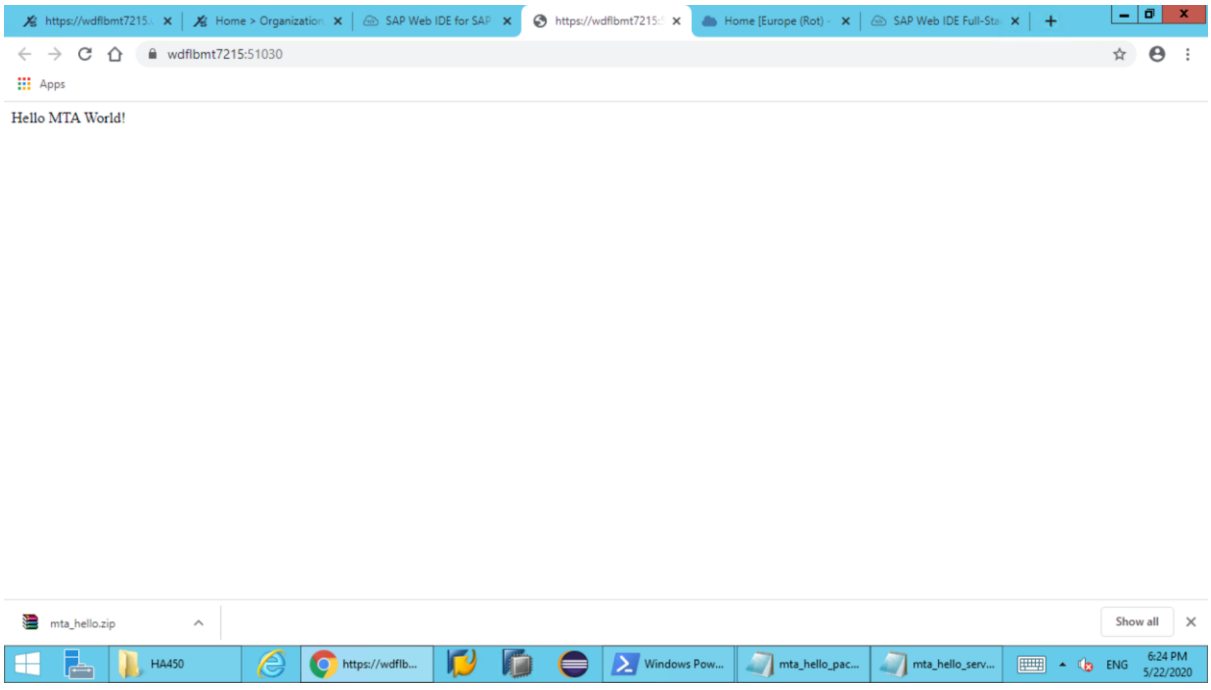
```
const express = require('express');

const app = express();
app.get('/', function (req, res) {
  res.send('Hello MTA World!');
});

const port = process.env.PORT || 3000;
app.listen(port);
```



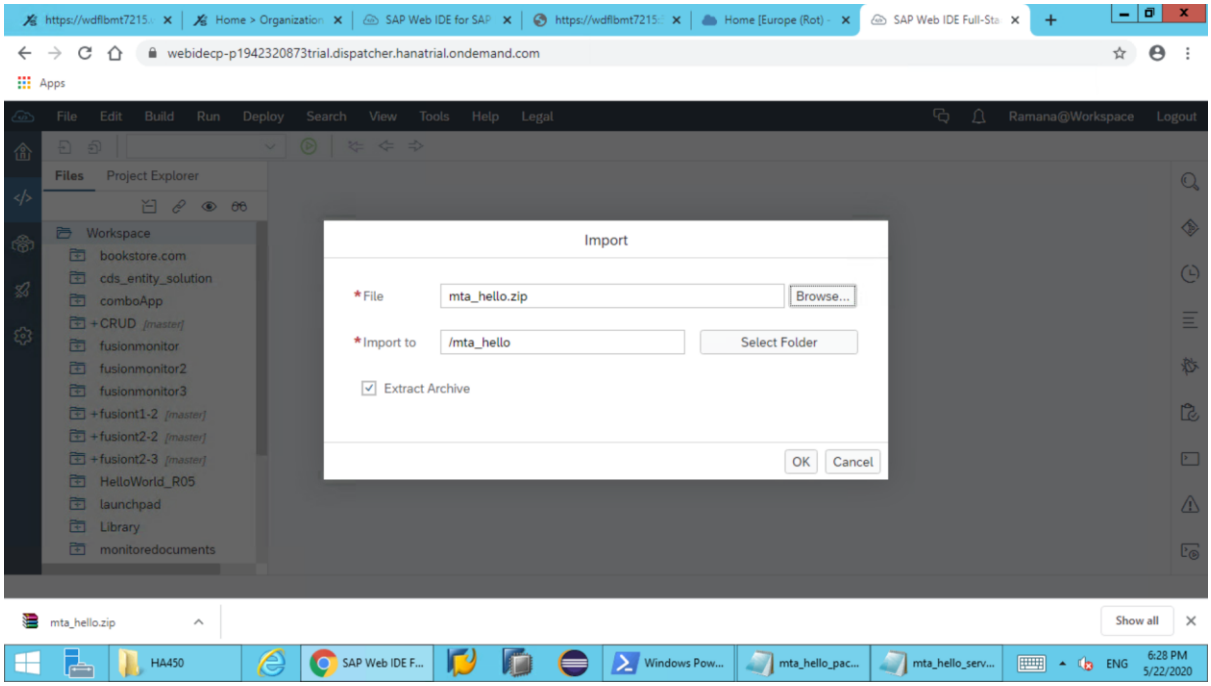
Now build the node_hello folder and mta_hello folders.

And select node_hello and run as node.js application.

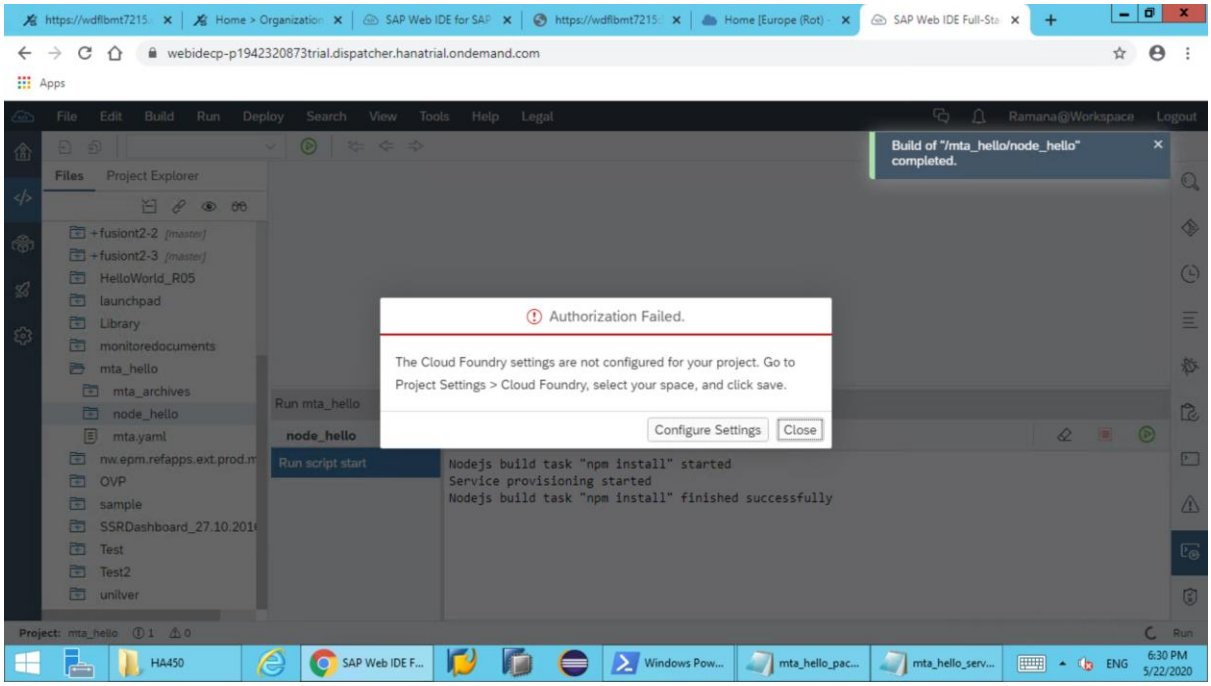


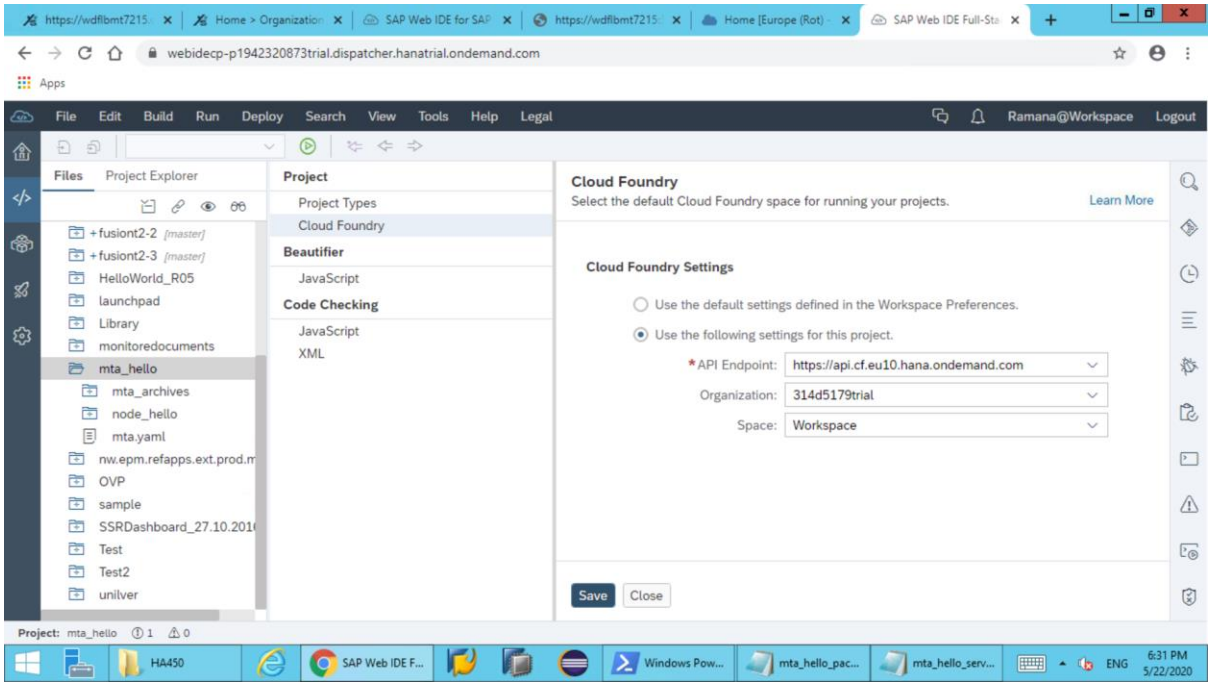


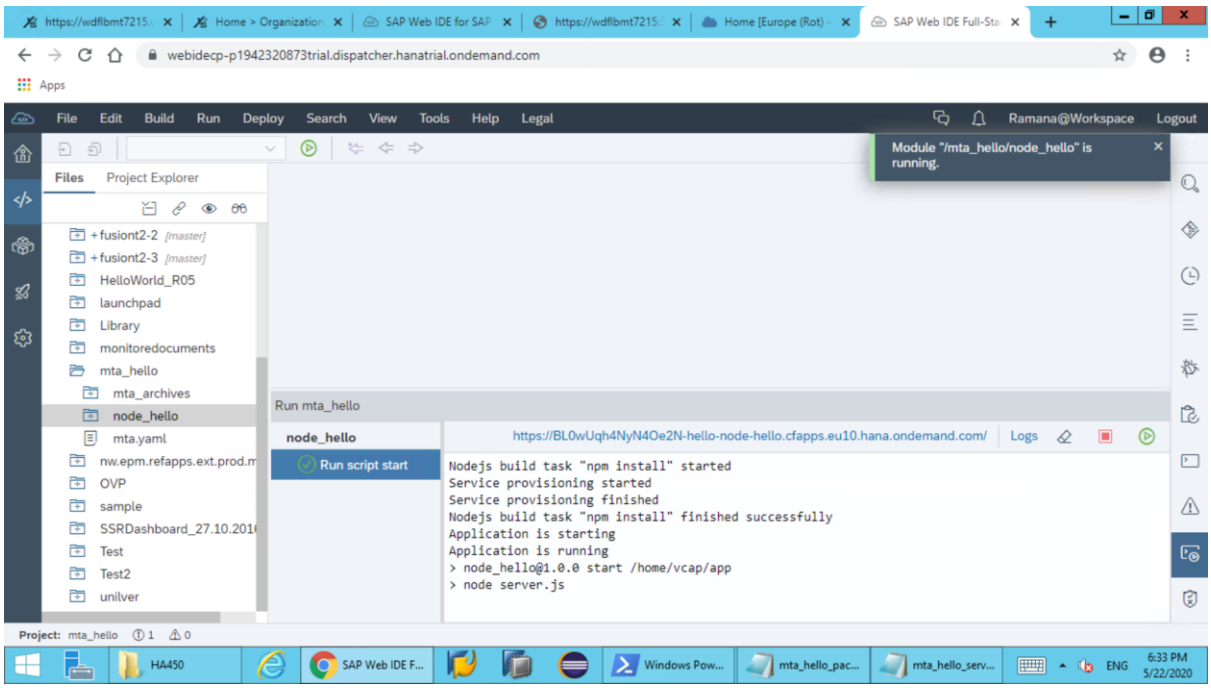
Now importing this project and lets run in SAP Cloud platform trial account. Open webide and import created node_hello from onpremise system

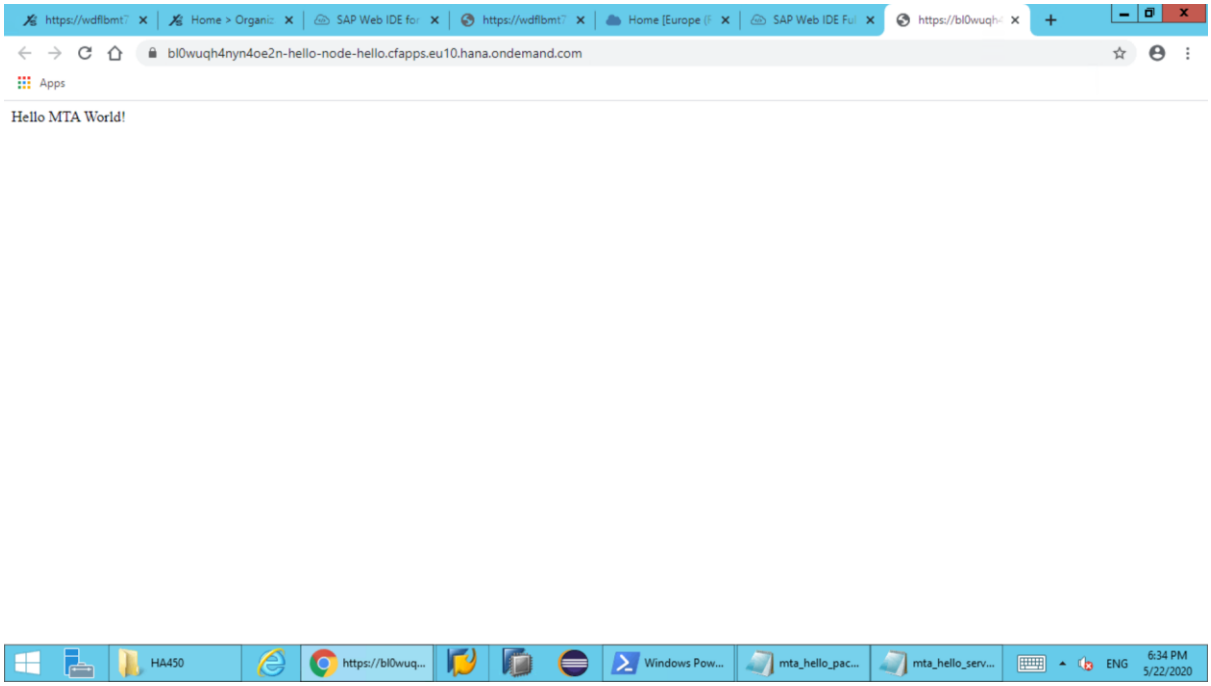


Build the mta_hello foler and then node_hello folder.

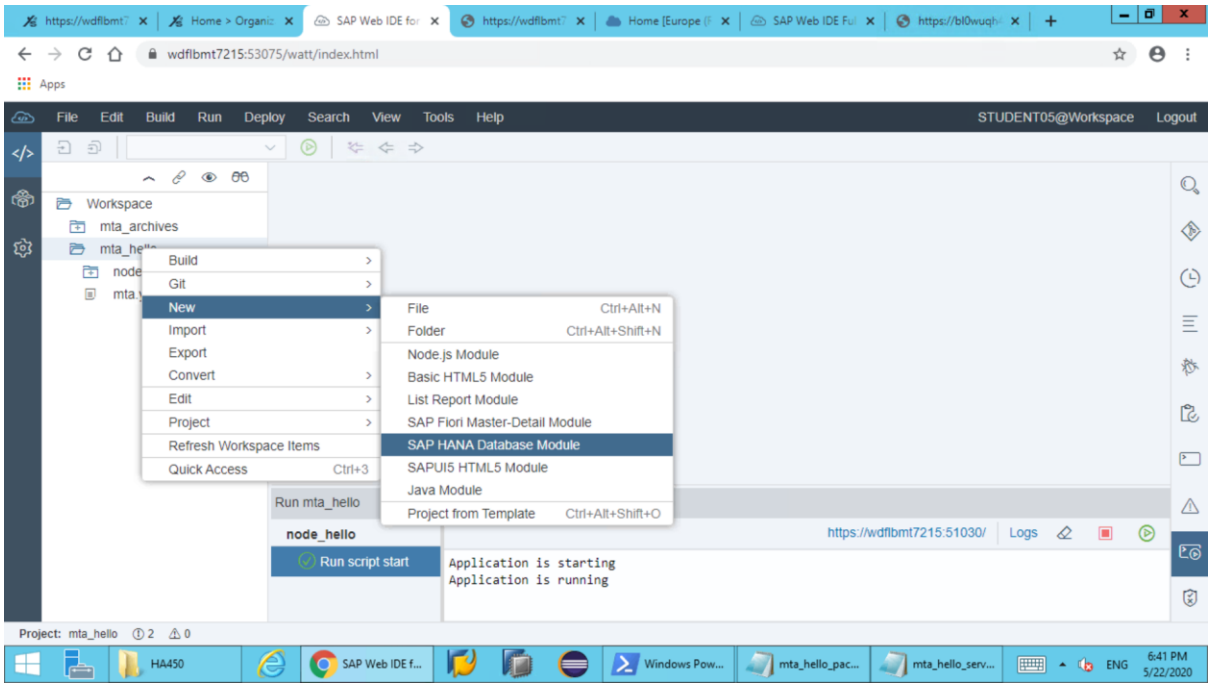


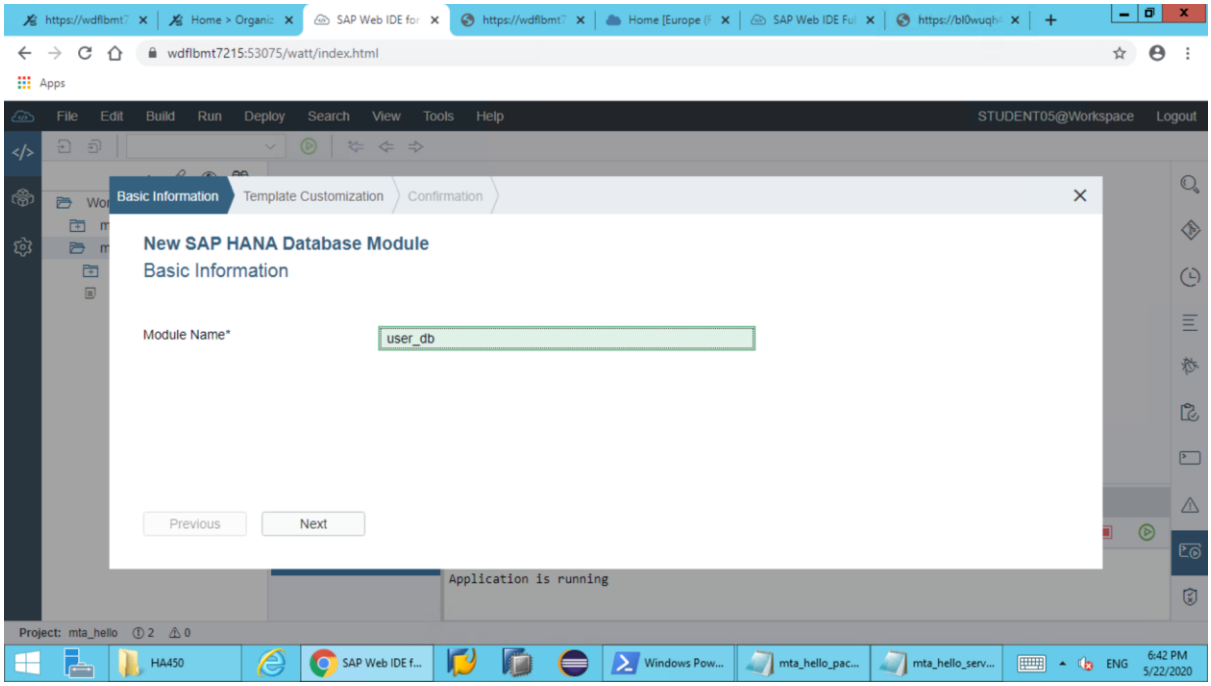

webidecp-p1942320873trial.dispatcher.hanatrial.ondemand.com
File Edit Build Run Deploy Search View Tools Help Legal
Ramana@Workspace Logout
Files Project Explorer
Project
Project Types
Cloud Foundry
Beautifier
JavaScript
Code Checking
JavaScript
XML
Cloud Foundry
Select the default Cloud Foundry space for running your projects.
Learn More
Cloud Foundry Settings
Use the default settings defined in the Workspace Preferences.
Use the following settings for this project.
*API Endpoint: https://api.cf.eu10.hana.ondemand.com
Organization: 314d5179trial
Space: Workspace
Save
Close
Project: mta_hello


webidecp-p1942320873trial.dispatcher.hanatrial.ondemand.com
File Edit Build Run Deploy Search View Tools Help Legal
Ramana@Workspace Logout
Module "/mta_hello/node_hello" is running.
Files Project Explorer
Run mta_hello
node_hello
Run script start
https://BL0wUqh4NyN4Oe2N-hello-node-hello.cfapps.eu10.hana.ondemand.com/
Logs
Nodejs build task "npm install" started
Service provisioning started
Service provisioning finished
Nodejs build task "npm install" finished successfully
Application is starting
Application is running
> node_hello@1.0.0 start /home/vcap/app
> node server.js
Project: mta_hello

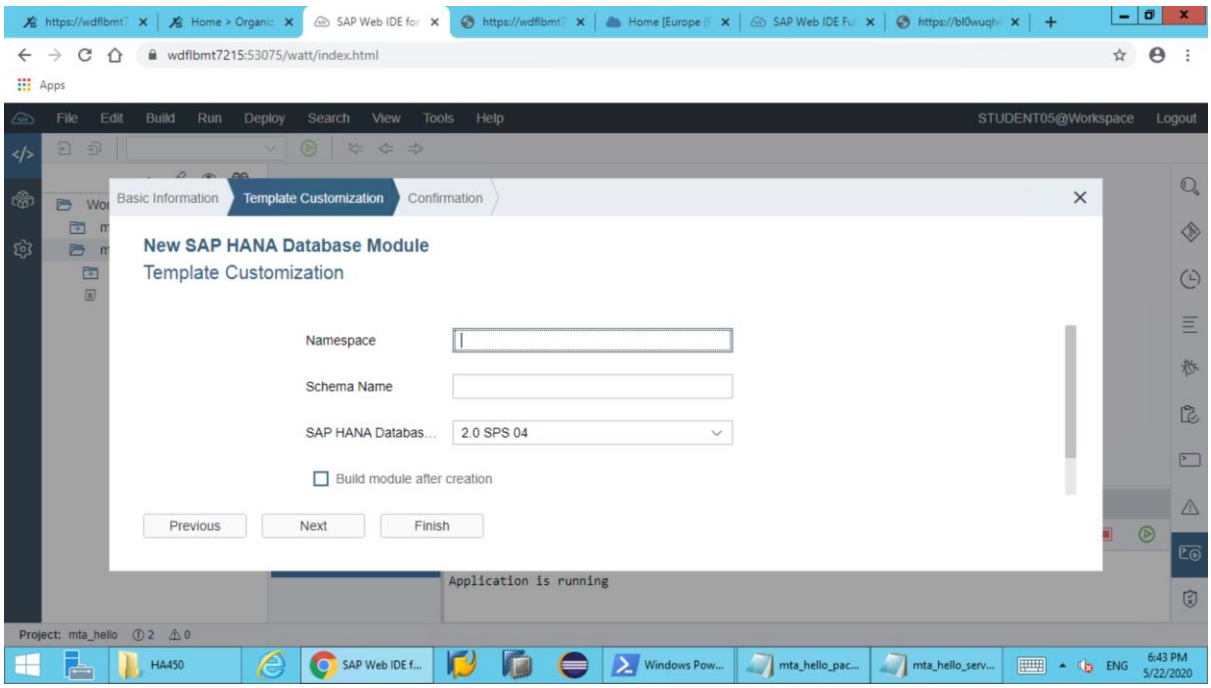


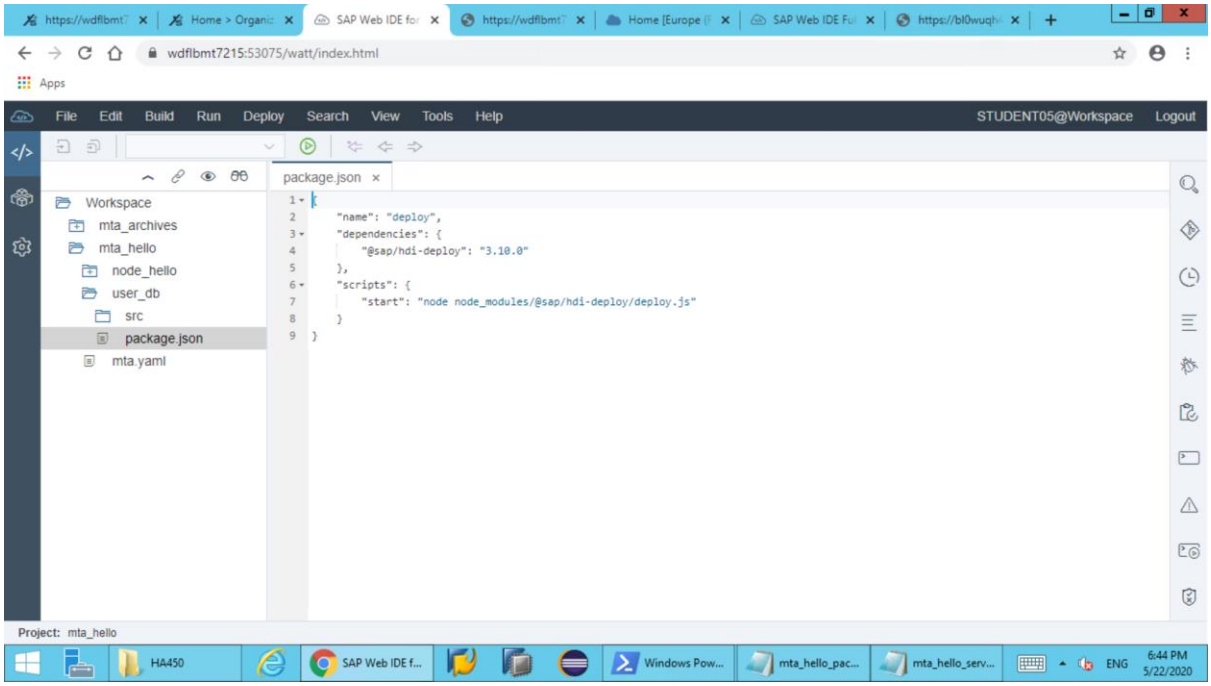
Now lets add Hana Database module using CDS .



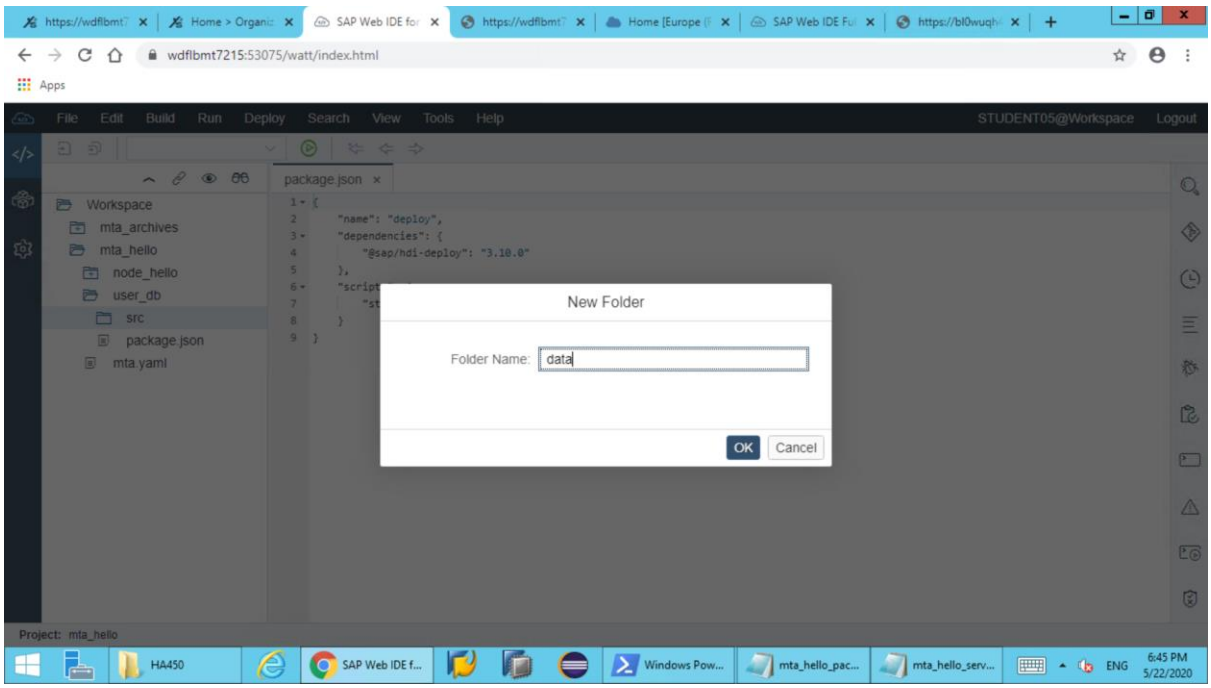


Let namespace be blank and proceed

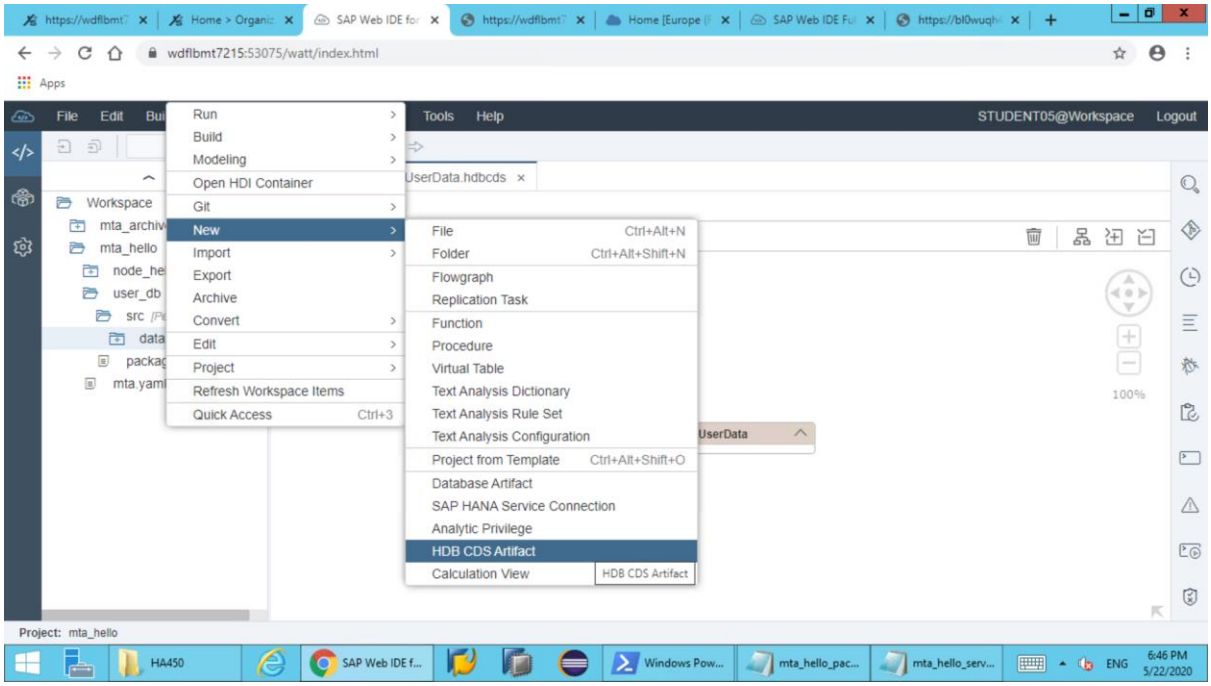




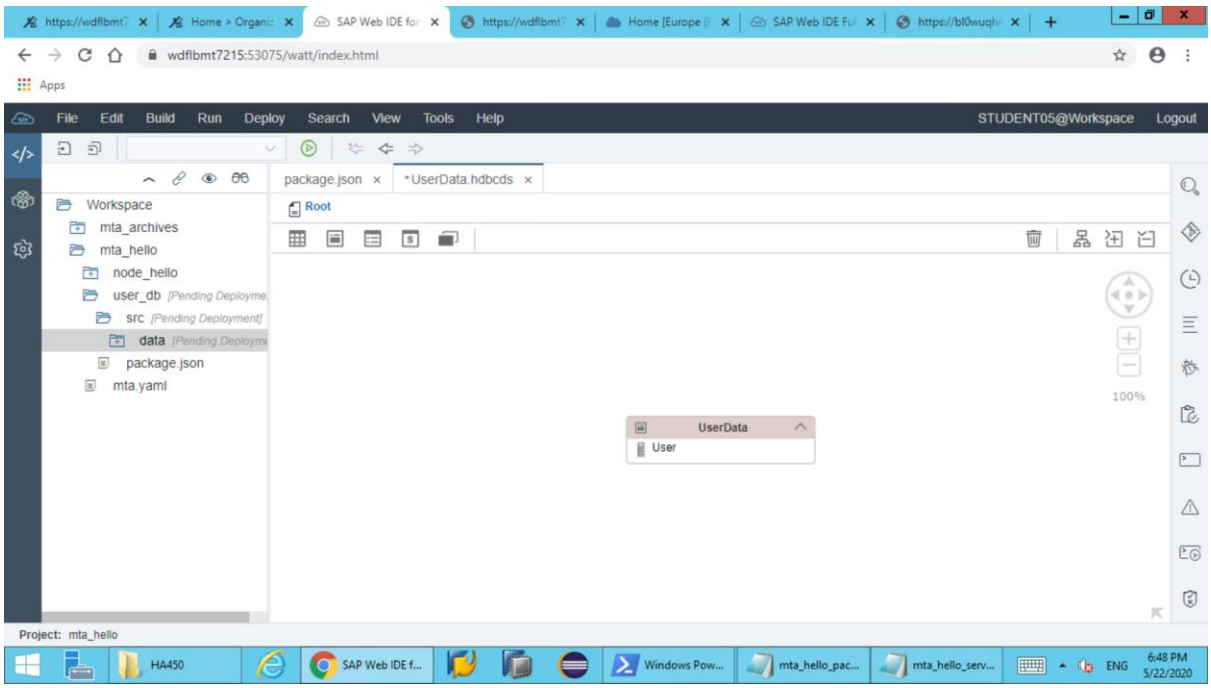
Create a new subfolder data ->under src



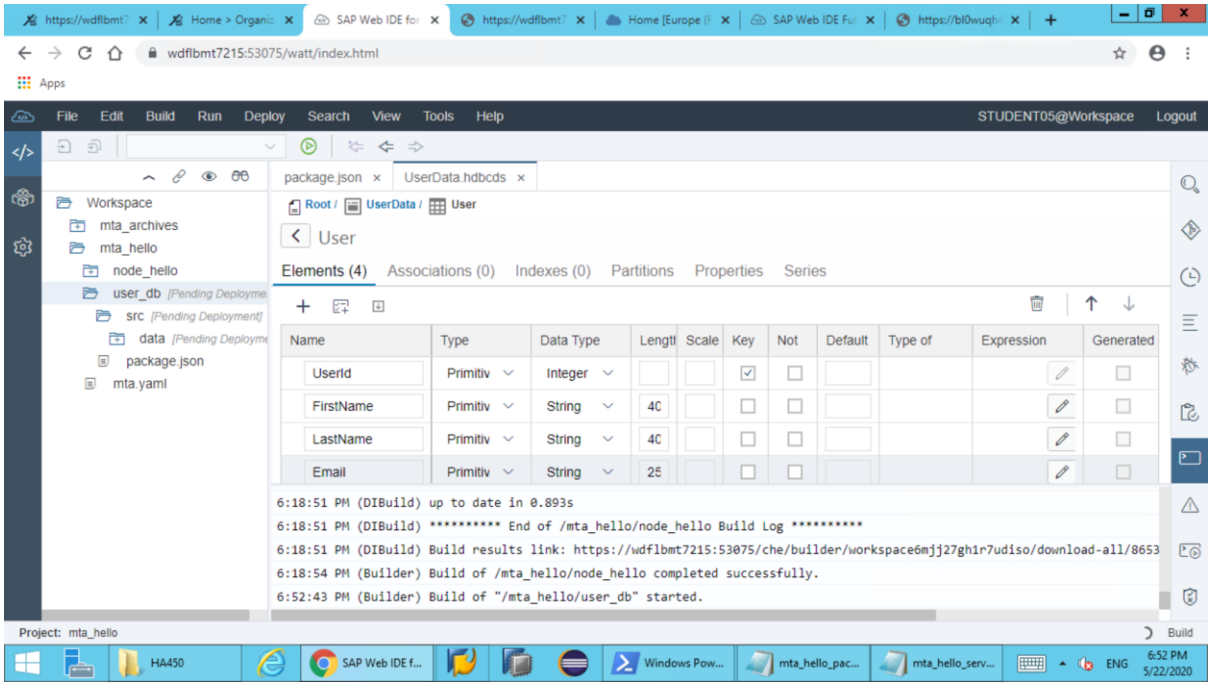
Create new file UserData.hdbcds



Now double click on UserData

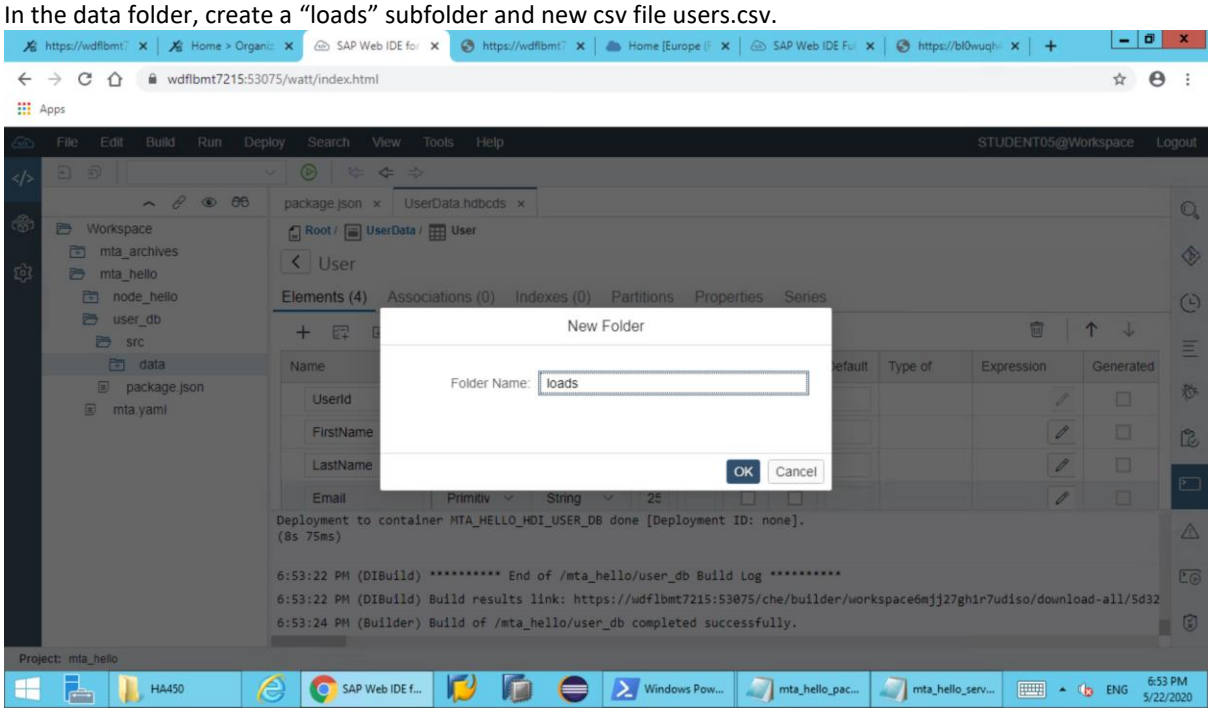


Now create a entity named User and double click and add properties.



Now build the user_db module.

In the data folder, create a “loads” subfolder and new csv file users.csv.

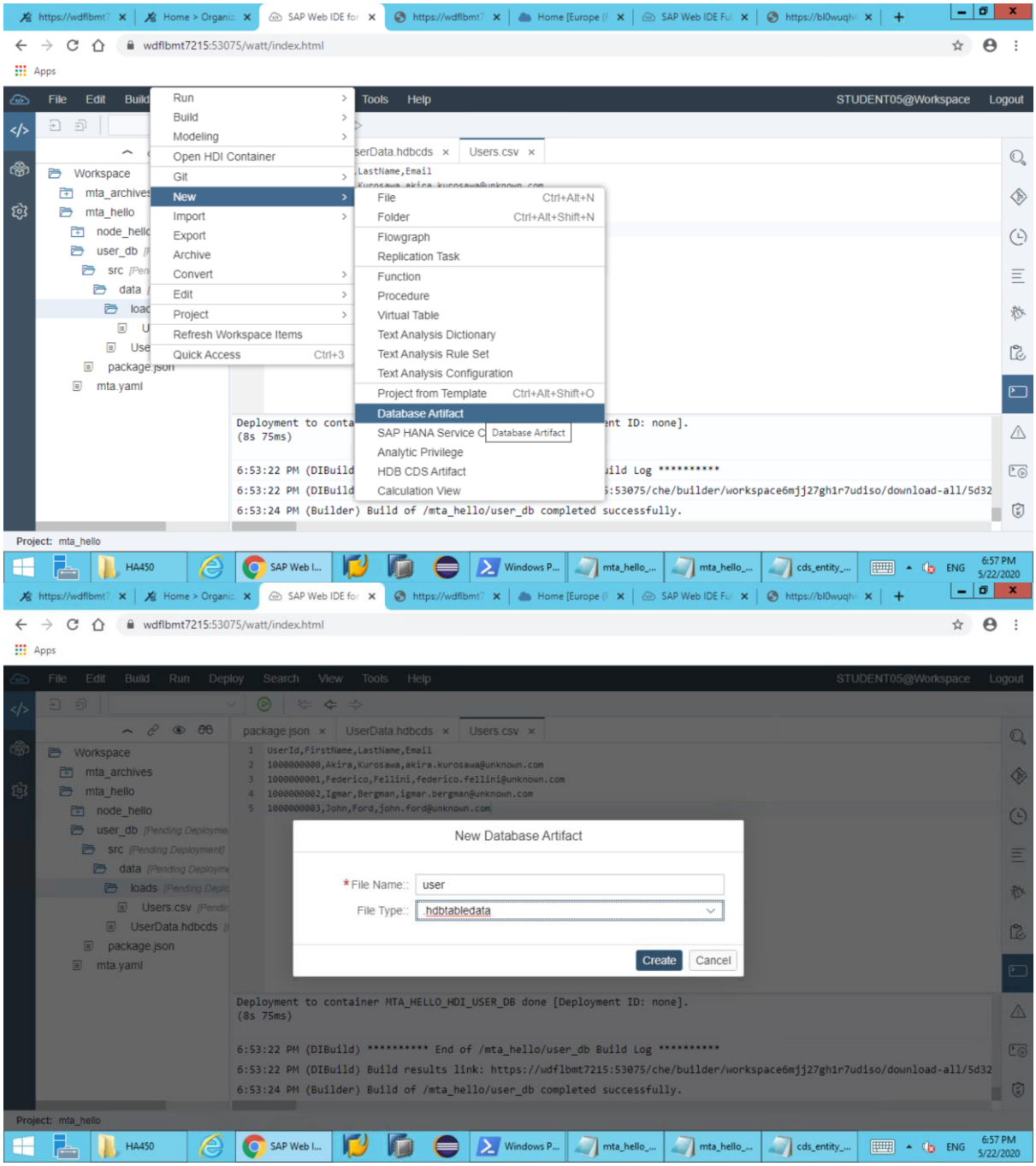


Users.csv

UserId,FirstName,LastName,Email
1000000000,Akira,Kurosawa,akira.kurosawa@unknown.com
1000000001,Federico,Fellini,federico.fellini@unknown.com
1000000002,Igmar,Bergman,igmar.bergman@unknown.com

1000000003,John,Ford,john.ford@unknown.com

In the loads folder, create a table data file named User.hdbtabledata to load data of the users.csv file into the UserData.User table.



User.hdbtabledata

```
{
  "format_version": 1,
  "imports": [
    {
      "target_table": "UserData.User",
      "source_data": {
```

```
            "data_type": "CSV",
            "file_name": "users.csv",
            "has_header": true
          }
      }
  ]
}
```

Now build the user_db folder.
Once build is successful. Right click folder and select open HDI Container.

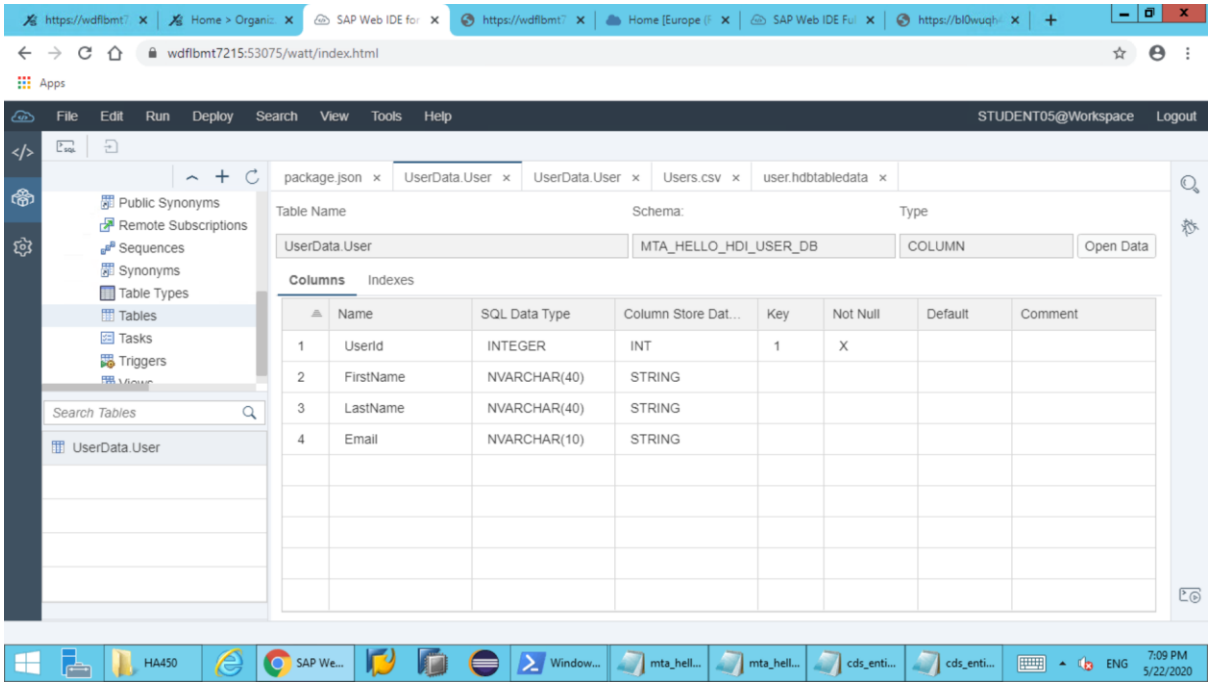
Display the created database table and its content.

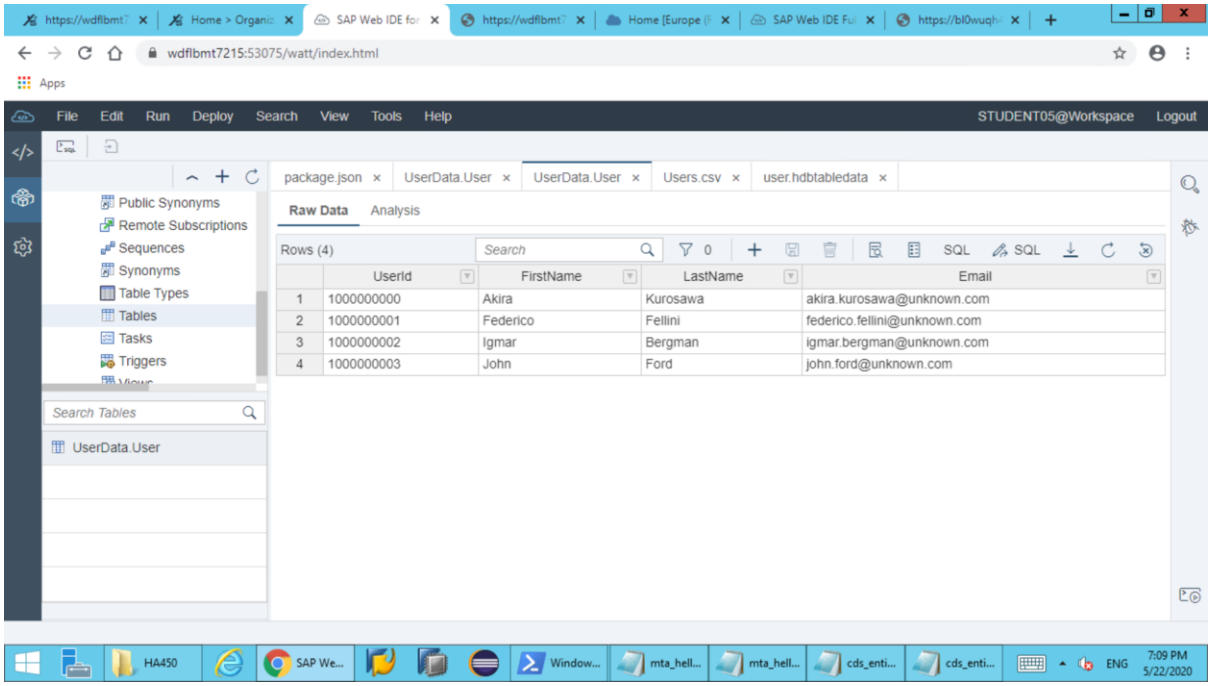
Right-click on the SAP HANA database module (user_db) folder and choose Open HDI Container.

Select the Tables subfolder.

Right-click the UserData.User table and choose Open Data.

You should be able to see the data from the users.csv file uploaded to the table.

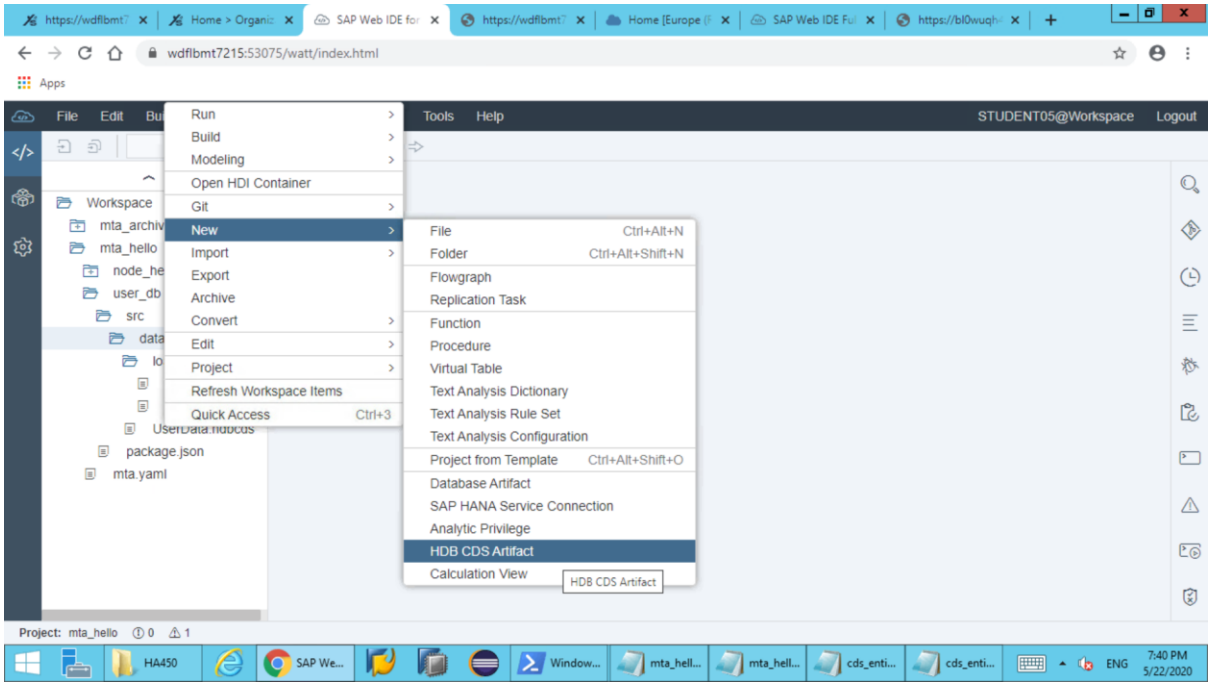


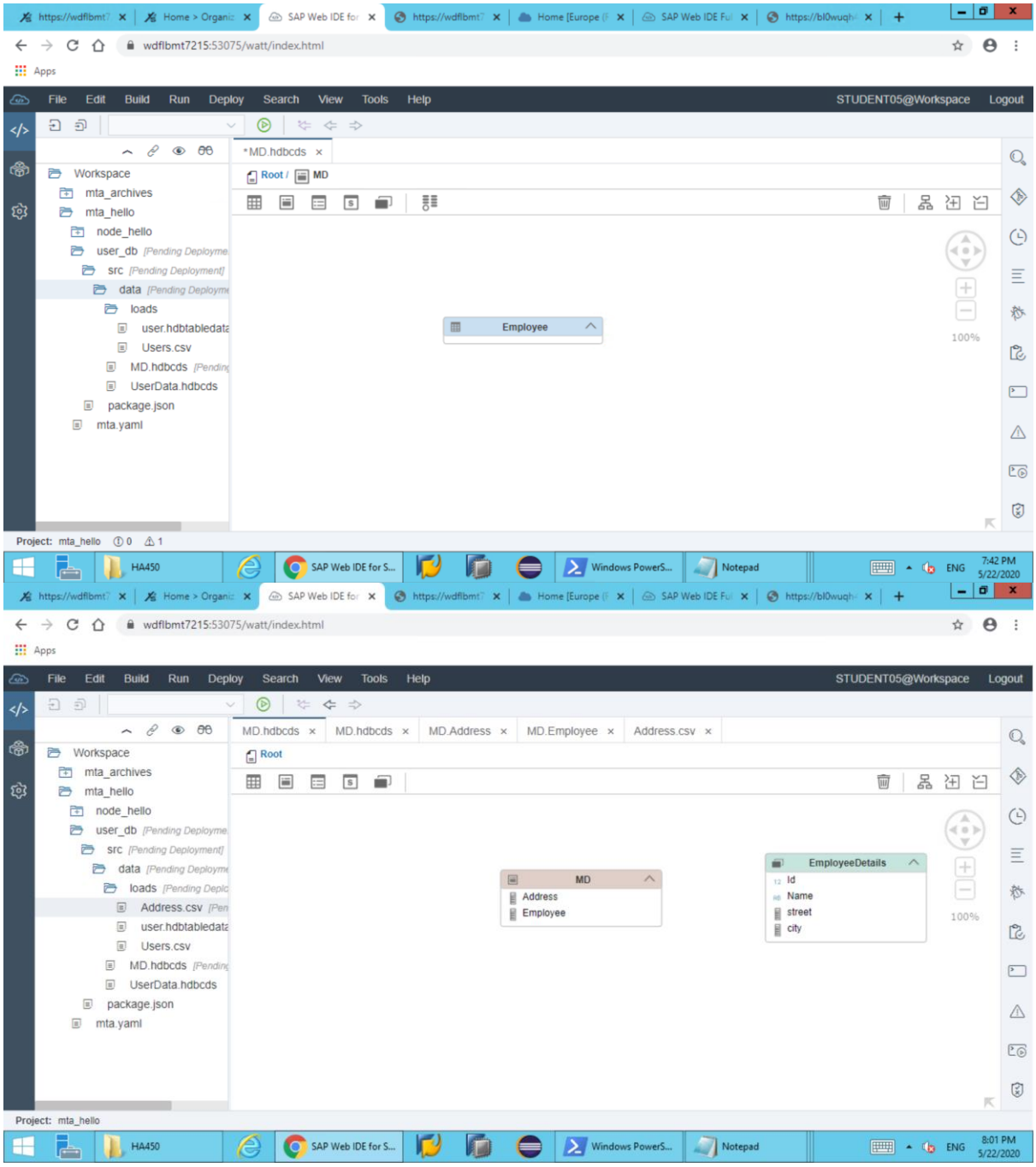


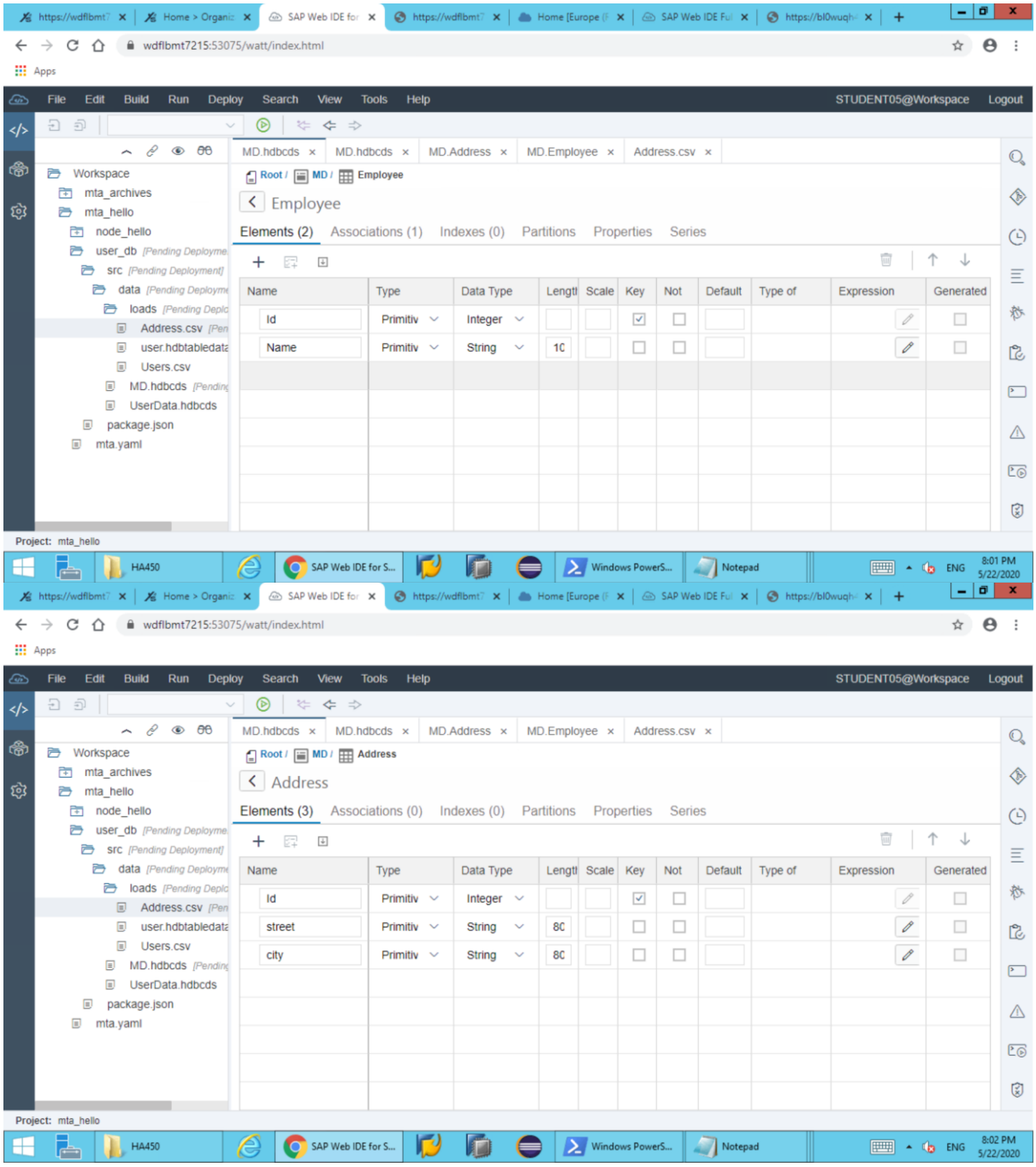
Note: all files name are case sensitive. If any incorrect info, build fails. Check the file names used in the hdbtabledata file .

Now we could successfully, using cds created a database at runtime using hdbcds and used csv mock data and loaded data into the database table using hdbtabledata file.

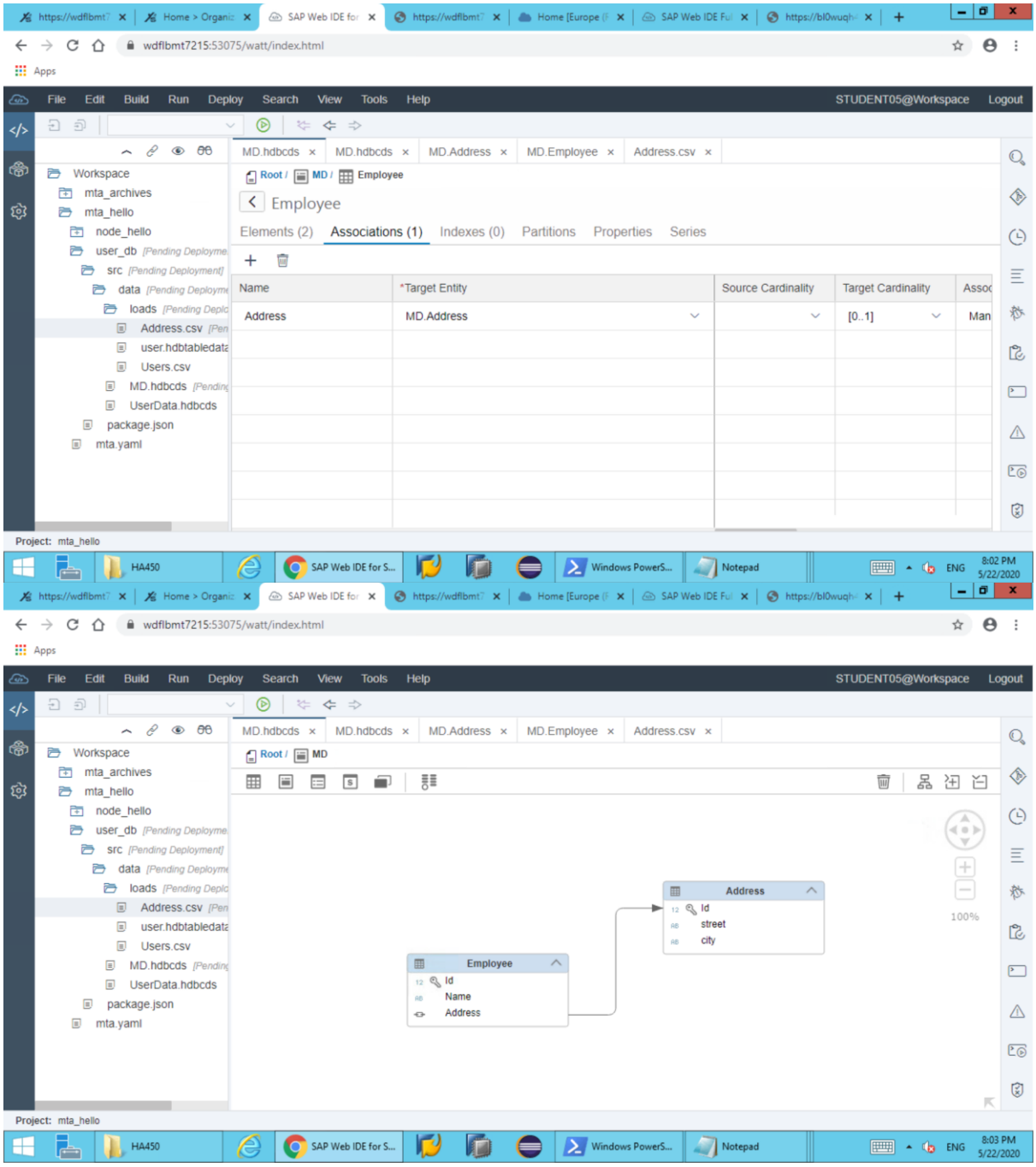
Lets create a Associations and a corresponding views using CDS




wdflbmt7215:53075/watt/index.html
File Edit Build Run Deploy Search View Tools Help
STUDENT05@Workspace Logout
*MD.hdbcds
Root / MD
Workspace
mta_archives
mta_hello
node_hello
user_db [Pending Deployme
src [Pending Deployment]
data [Pending Deployme
loads
user.hdbtabledata
Users.csv
MD.hdbcds [Pending
UserData.hdbcds
package.json
mta.yaml
Employee
100%
Project: mta_hello
MD.hdbcds
MD.Address
MD.Employee
Address.csv
Root
Address.csv [Pen
MD
Address
Employee
EmployeeDetails
Id
Name
street
city



Create Association



Now Build the user_db folder. For the created structure.

Lets add csv files to load into employee and Address.

Address.csv

id,street,city

1,Blade Avenue,Los Angeles

3,Termin Street,New Orleans

2,Alien Square,Huston

Employee.csv

id,name,address.id

1,Rick Deckard,1

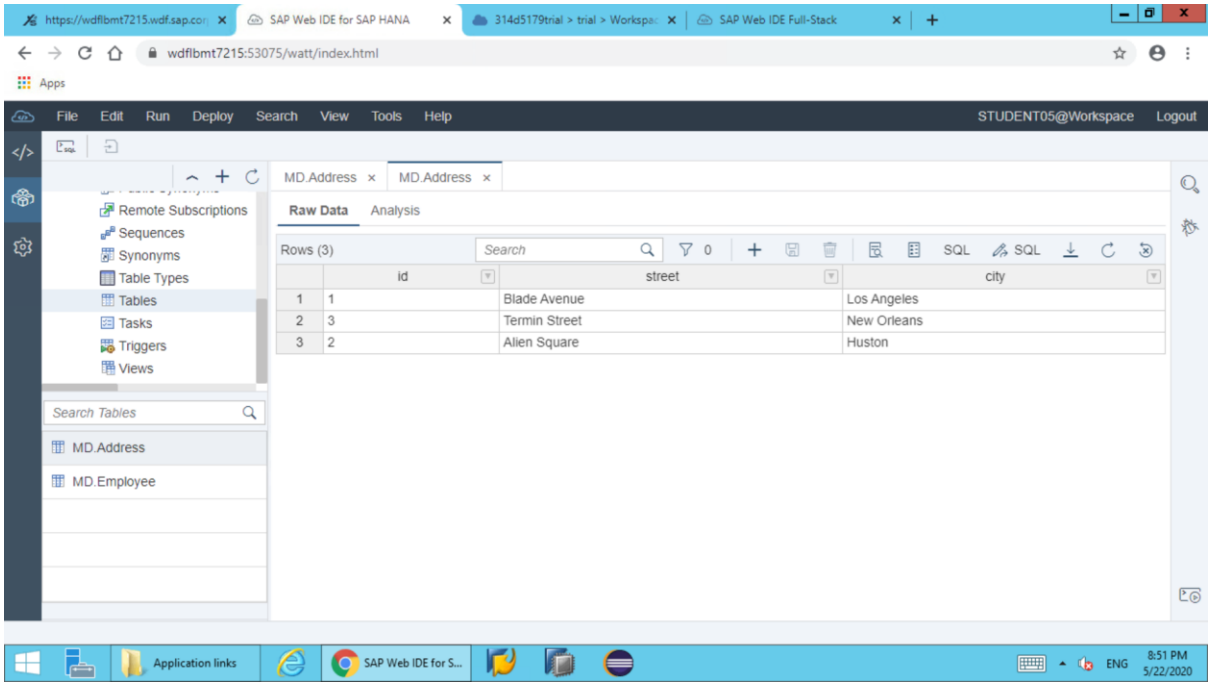
2,Sarah Connor,3

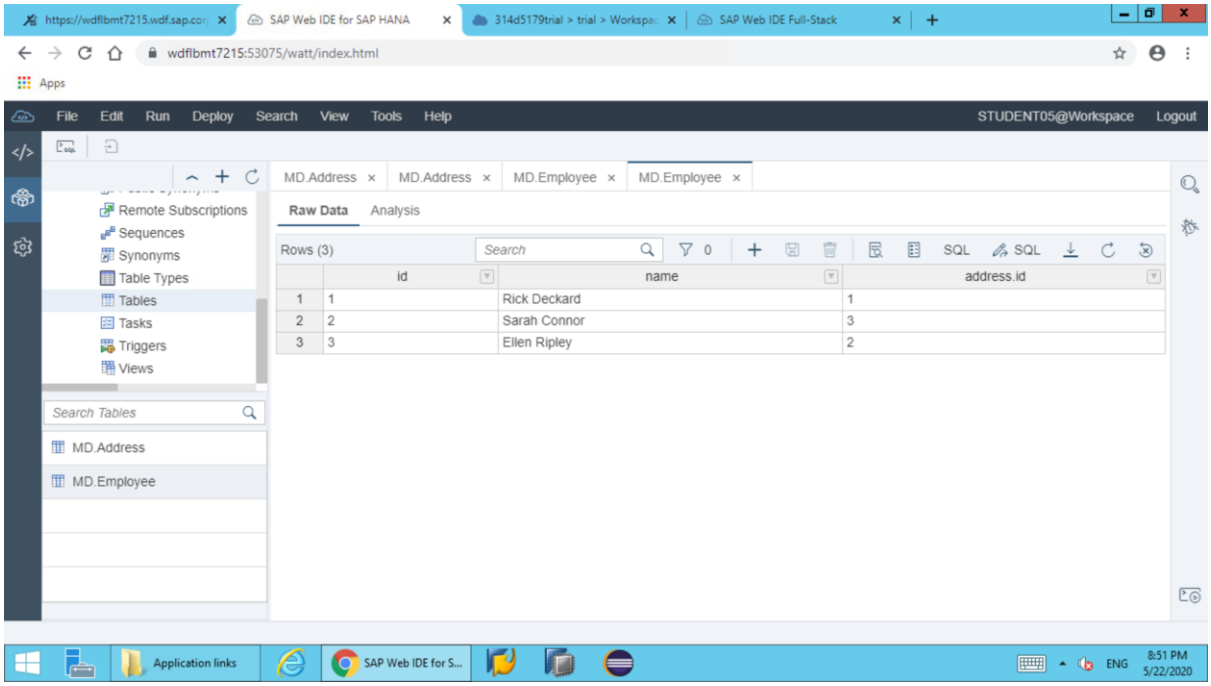
3,Ellen Ripley,2

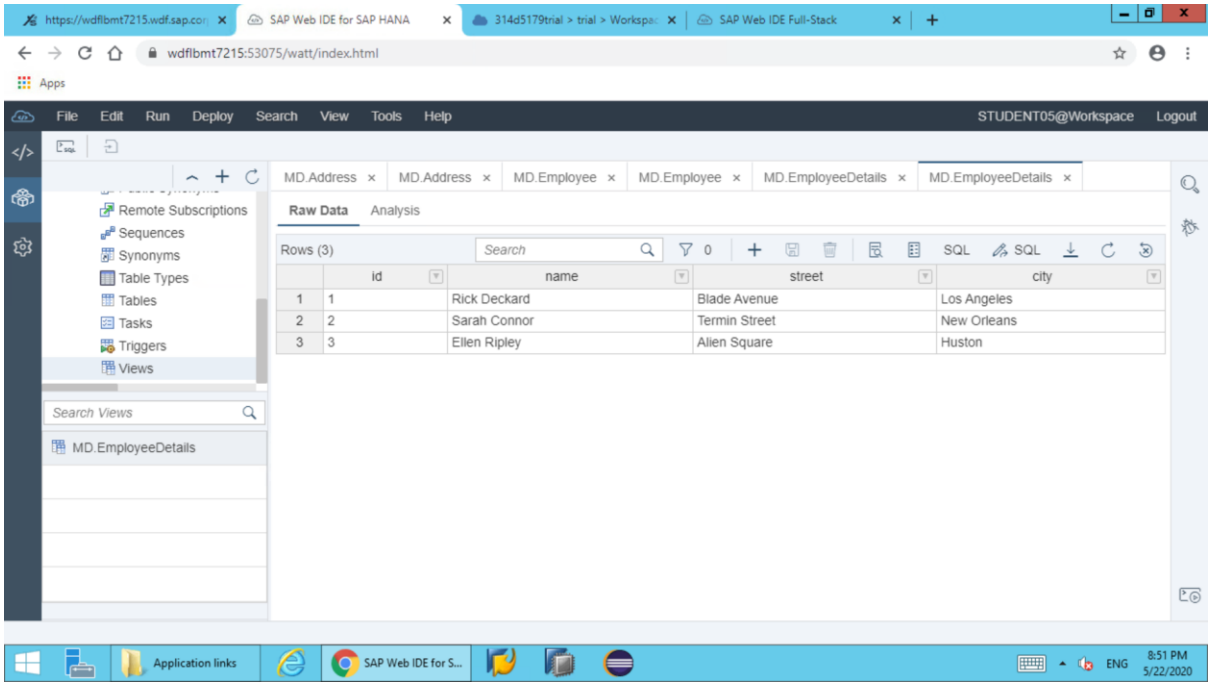
MD.hdbtabledata

```
{
   "format_version": 1,
   "imports": [
      {
         "target_table": "MD.Employee",
         "source_data": {
            "data_type": "CSV",
            "file_name": "employees.csv",
            "has_header": true
         }
      },
                     {
         "target_table": "MD.Address",
         "source_data": {
            "data_type": "CSV",
            "file_name": "addresses.csv",
            "has_header": true
         }
      }
   ]
}
```

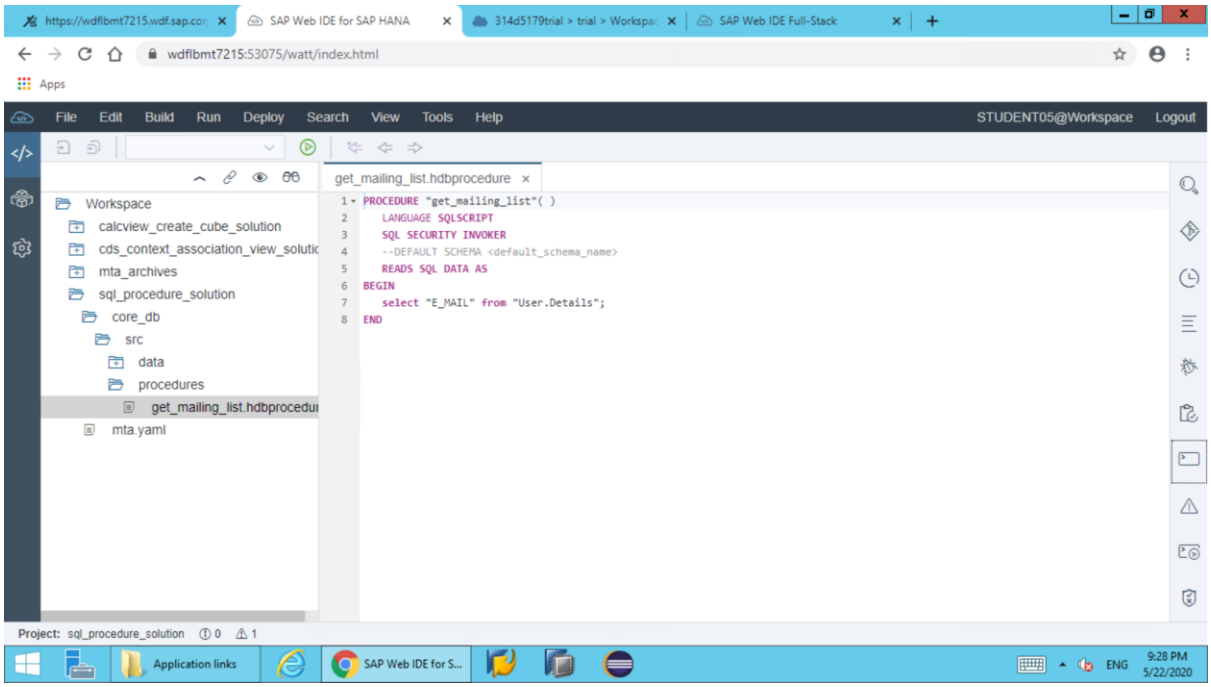
Note: In the SAP Cloud Platform you won't find the graphical CDS editor for .hdbcds files. The code editor must be used instead.

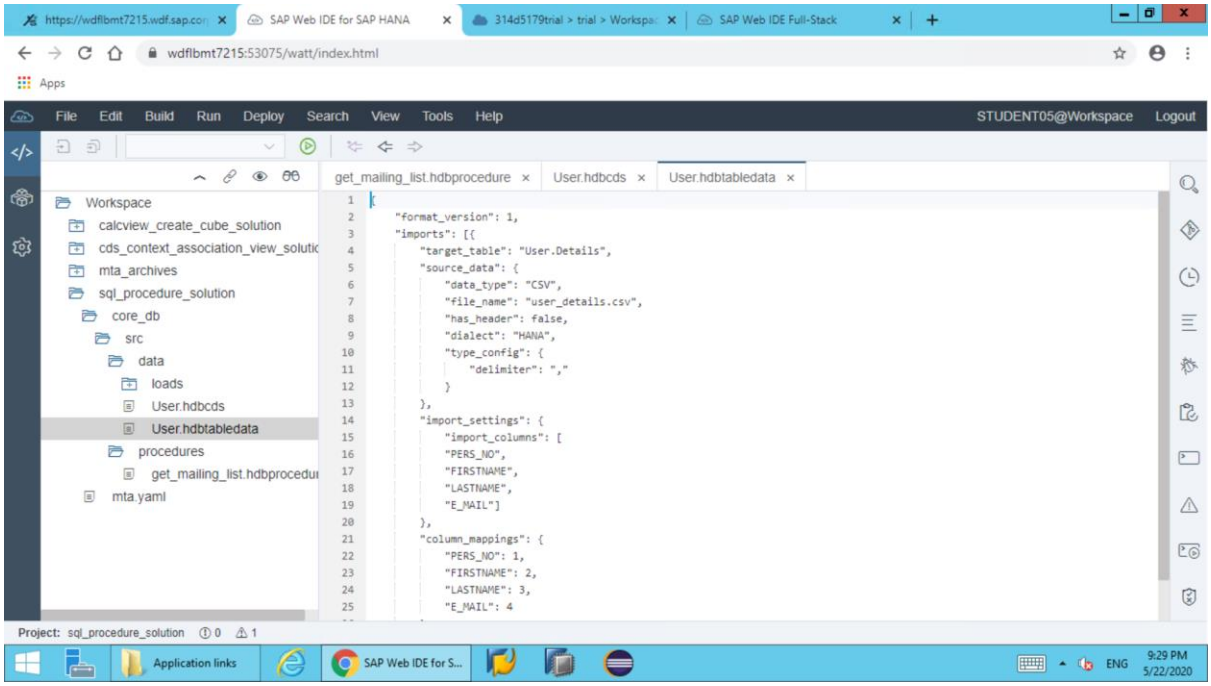


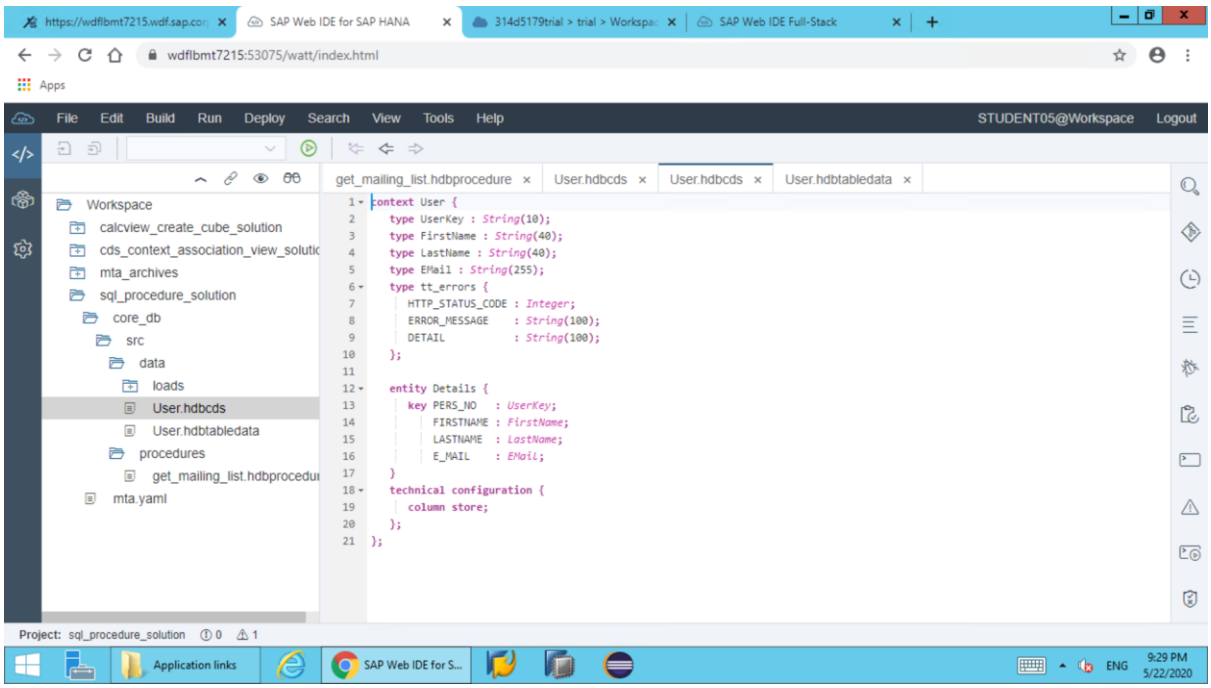


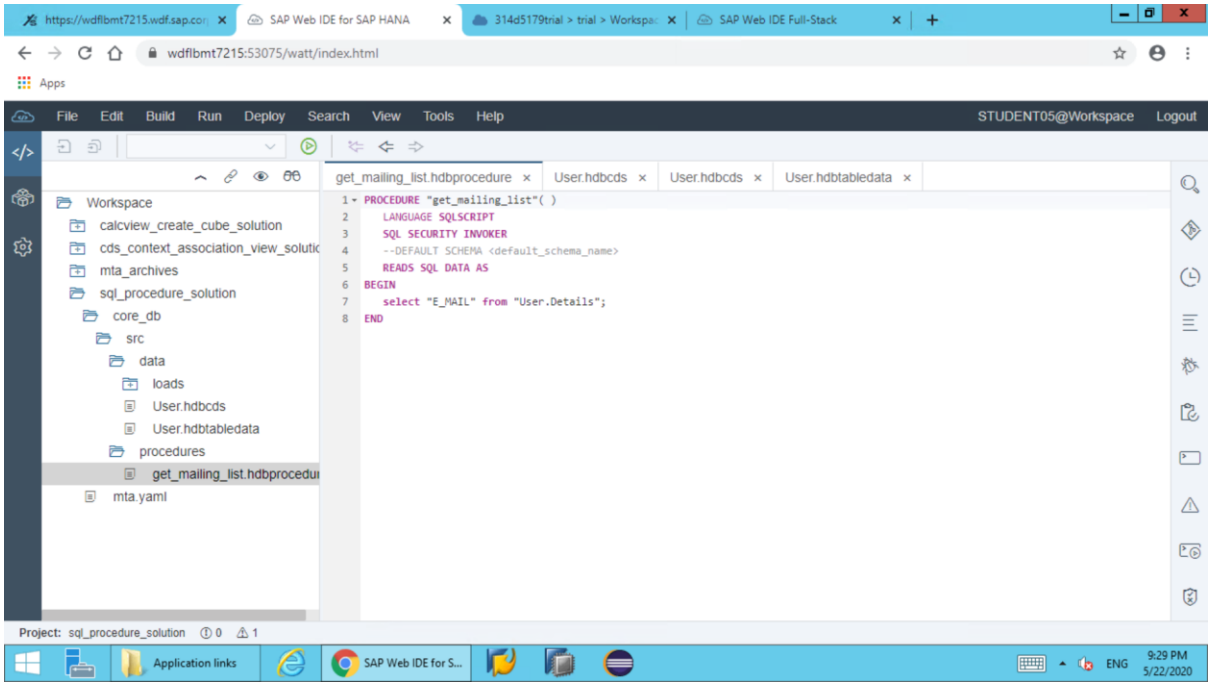


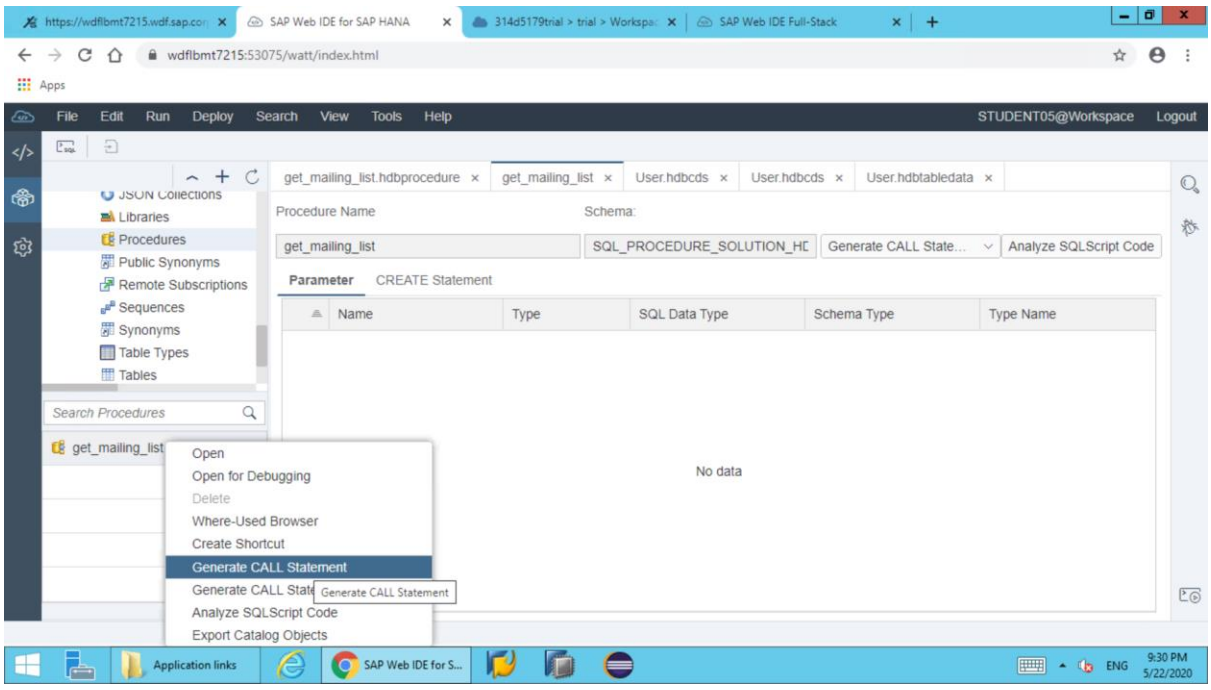
Create a simple procedure that will return a mailing list that can be used to send out news or notifications to users.

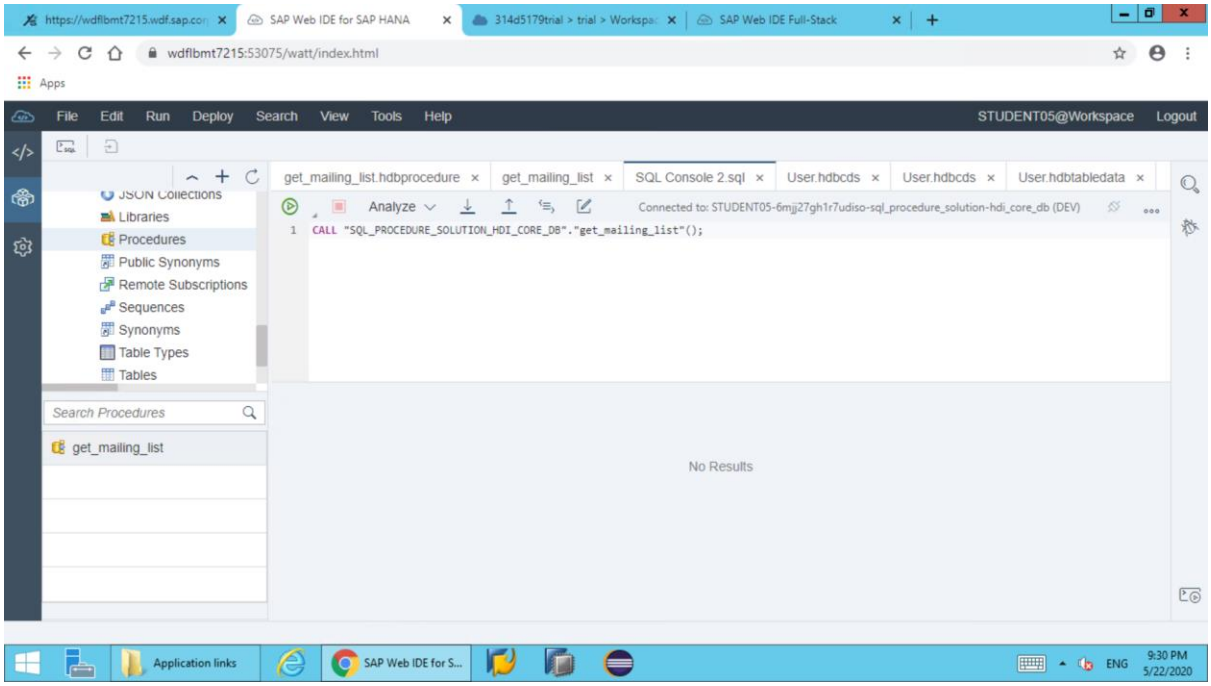


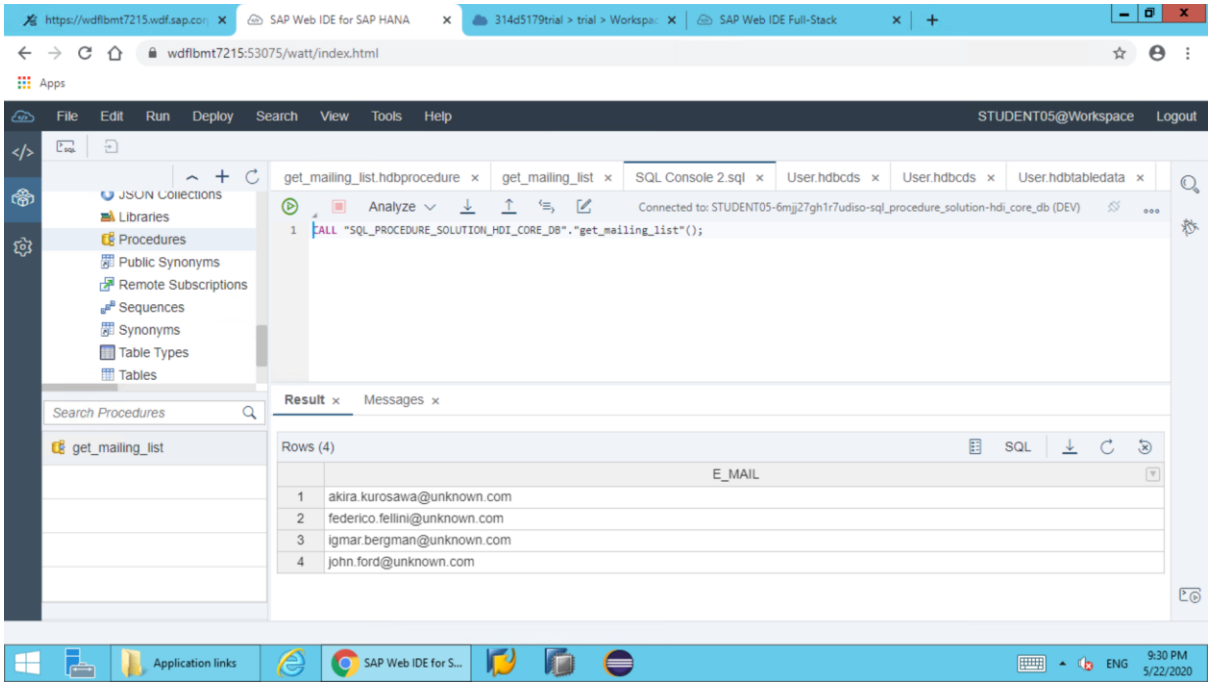

SAP Web IDE for SAP HANA
wdflbmt7215:53075/watt/index.html
File Edit Build Run Deploy Search View Tools Help
STUDENT05@Workspace Logout
Workspace
calcview_create_cube_solution
cds_context_association_view_solutic
mta_archives
sql_procedure_solution
core_db
src
data
loads
User.hdbcds
User.hdbtabledata
procedures
get_mailing_list.hdbprocedu
mta.yaml
get_mailing_list.hdbprocedure
User.hdbcds
User.hdbtabledata
{
"format_version": 1,
"imports": [{
"target_table": "User.Details",
"source_data": {
"data_type": "CSV",
"file_name": "user_details.csv",
"has_header": false,
"dialect": "HANA",
"type_config": {
"delimiter": ","
}
},
"import_settings": {
"import_columns": [
"PERS_NO",
"FIRSTNAME",
"LASTNAME",
"E_MAIL"]
},
"column_mappings": {
"PERS_NO": 1,
"FIRSTNAME": 2,
"LASTNAME": 3,
"E_MAIL": 4
Project: sql_procedure_solution
Application links
SAP Web IDE for S...
ENG
9:29 PM
5/22/2020


SAP Web IDE for SAP HANA
wdflbmt7215:53075/watt/index.html
File Edit Build Run Deploy Search View Tools Help
STUDENT05@Workspace Logout
Workspace
calcview_create_cube_solution
cds_context_association_view_solutic
mta_archives
sql_procedure_solution
core_db
src
data
loads
User.hdbcds
User.hdbtabledata
procedures
get_mailing_list.hdbprocedu
mta.yaml
get_mailing_list.hdbprocedure
User.hdbcds
User.hdbcds
User.hdbtabledata
context User {
type UserKey : String(10);
type FirstName : String(40);
type LastName : String(40);
type EMail : String(255);
type tt_errors {
HTTP_STATUS_CODE : Integer;
ERROR_MESSAGE : String(100);
DETAIL : String(100);
};
entity Details {
key PERS_NO : UserKey;
FIRSTNAME : FirstName;
LASTNAME : LastName;
E_MAIL : EMail;
}
technical configuration {
column store;
};
};
Project: sql_procedure_solution
Application links
SAP Web IDE for S...
ENG
9:29 PM
5/22/2020


SAP Web IDE for SAP HANA
wdflbmt7215:53075/watt/index.html
File Edit Build Run Deploy Search View Tools Help
STUDENT05@Workspace Logout
Workspace
calcview_create_cube_solution
cds_context_association_view_solutic
mta_archives
sql_procedure_solution
core_db
src
data
loads
User.hdbcds
User.hdbtabledata
procedures
get_mailing_list.hdbprocedu
mta.yaml
get_mailing_list.hdbprocedure
User.hdbcds
User.hdbtabledata
1 PROCEDURE "get_mailing_list"( )
2 LANGUAGE SQLSCRIPT
3 SQL SECURITY INVOKER
4 --DEFAULT SCHEMA <default_schema_name>
5 READS SQL DATA AS
6 BEGIN
7 select "E_MAIL" from "User.Details";
8 END
Project: sql_procedure_solution
Application links
SAP Web IDE for S...
ENG
9:29 PM
5/22/2020


SAP Web IDE for SAP HANA
wdflbmt7215:53075/watt/index.html
File Edit Run Deploy Search View Tools Help
STUDENT05@Workspace Logout
get_mailing_list.hdbprocedure
get_mailing_list
User.hdbcds
User.hdbtabledata
JSON Collections
Libraries
Procedures
Public Synonyms
Remote Subscriptions
Sequences
Synonyms
Table Types
Tables
Search Procedures
get_mailing_list
Procedure Name
Schema:
get_mailing_list
SQL_PROCEDURE_SOLUTION_HC
Generate CALL State...
Analyze SQLScript Code
Parameter
CREATE Statement
Name
Type
SQL Data Type
Schema Type
Type Name
No data
Open
Open for Debugging
Delete
Where-Used Browser
Create Shortcut
Generate CALL Statement
Generate CALL State
Generate CALL Statement
Analyze SQLScript Code
Export Catalog Objects
Application links
SAP Web IDE for S...
ENG
9:30 PM
5/22/2020

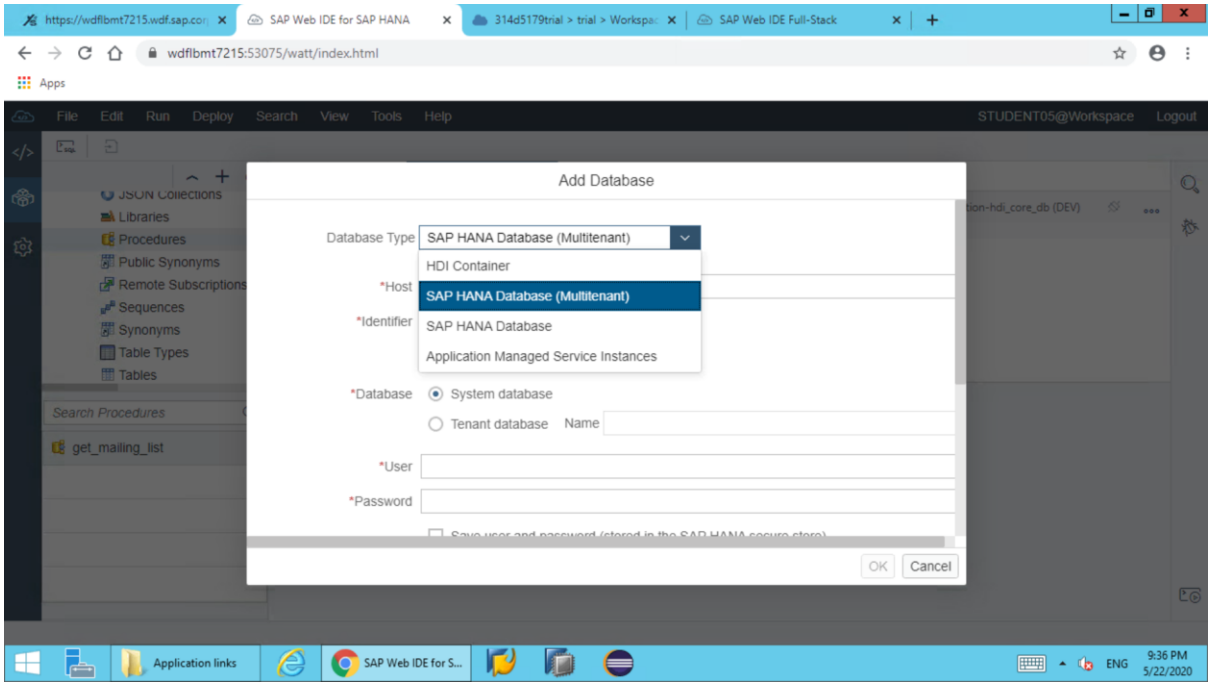


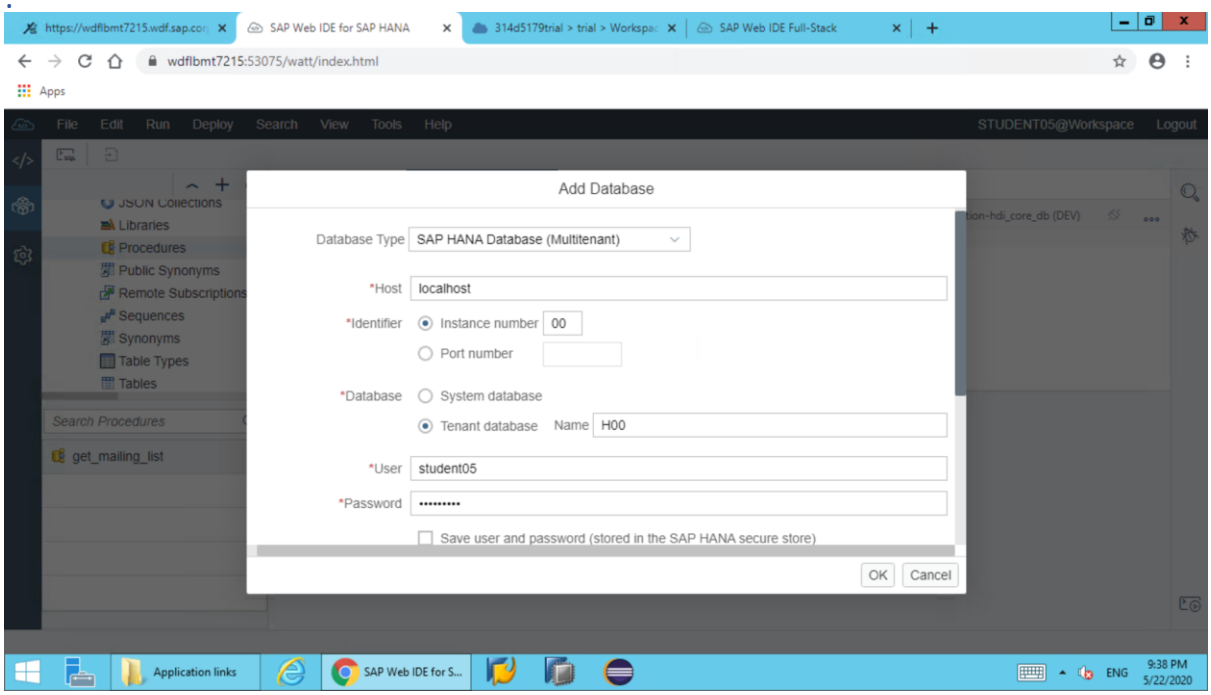


Now successfully the mailing list is obtained. This can be used to send alerts to the subscribers of news and notifications.

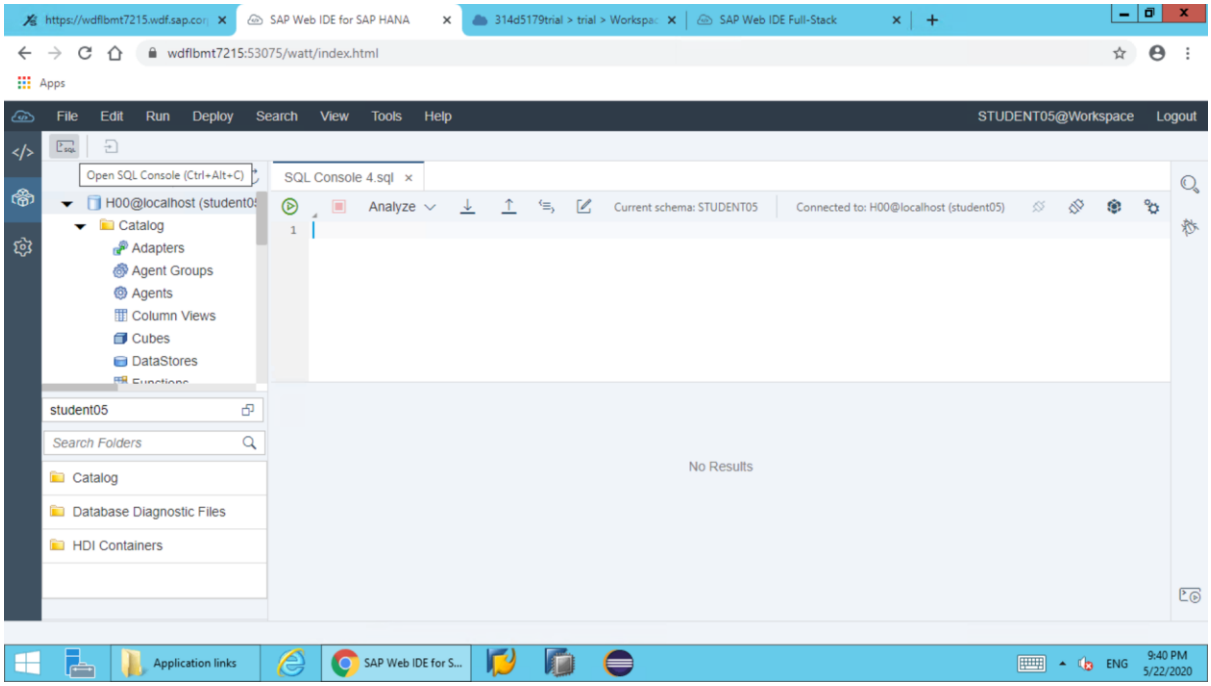
Accessing multiple remote databases / non sap databases data and displaying in Application

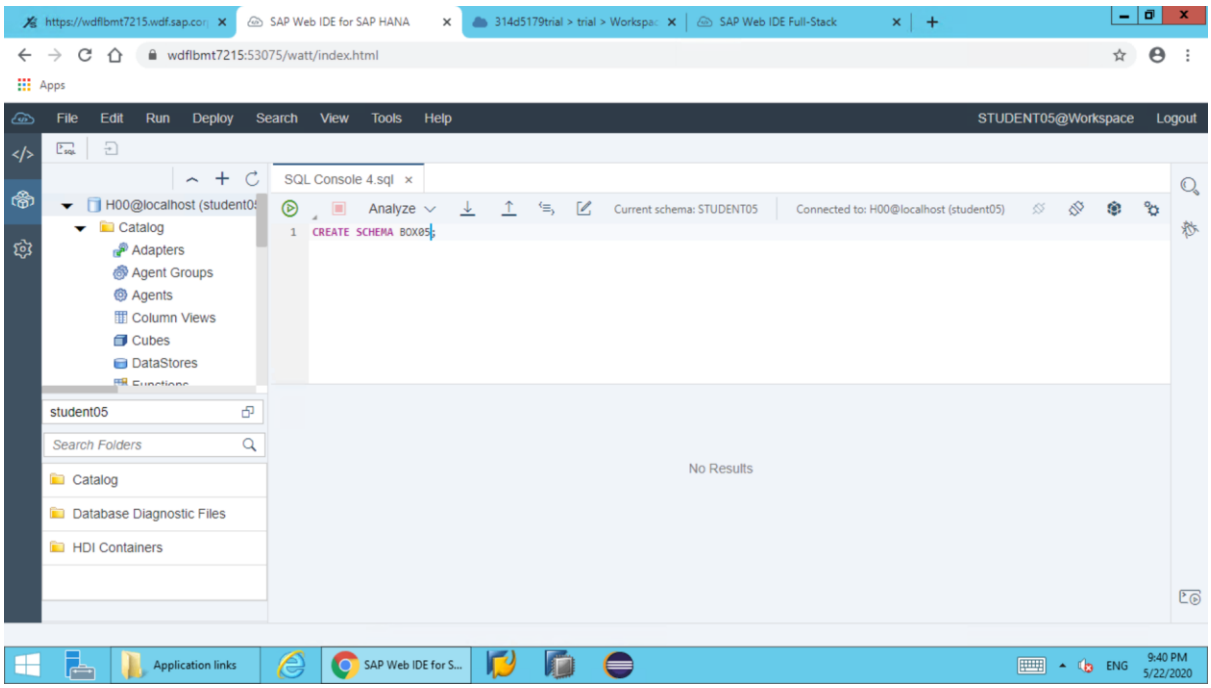
In Database explorer, add the remote database by selecting multitenant





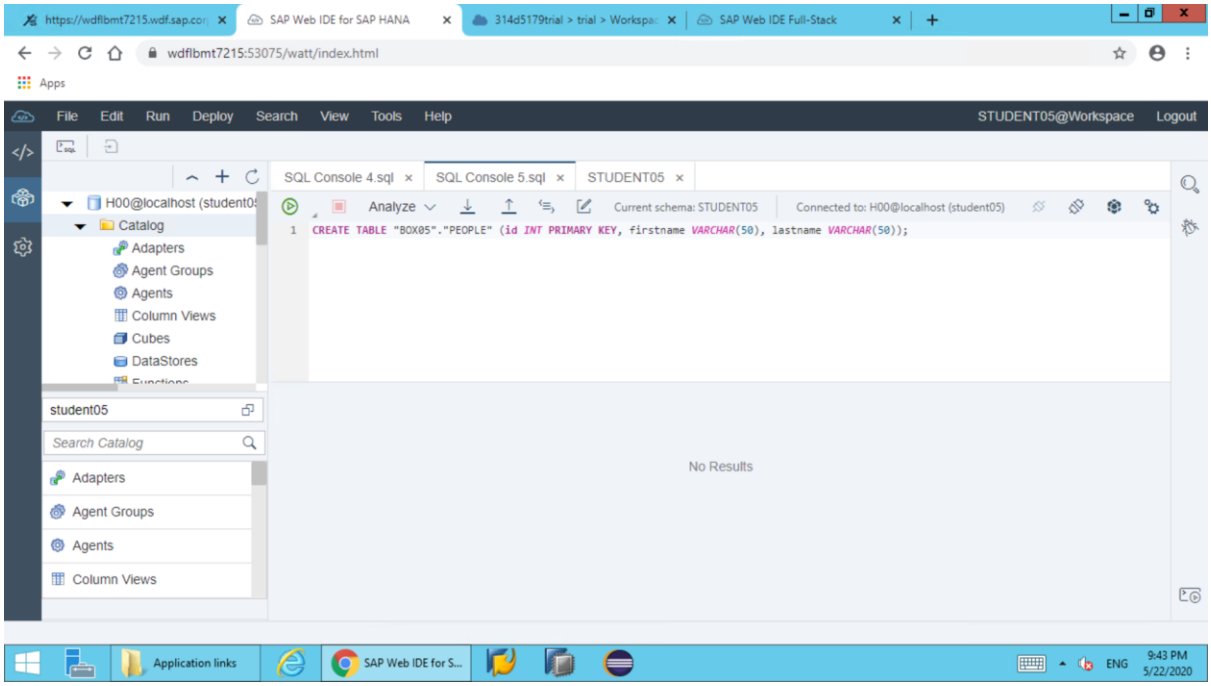
Create a schema



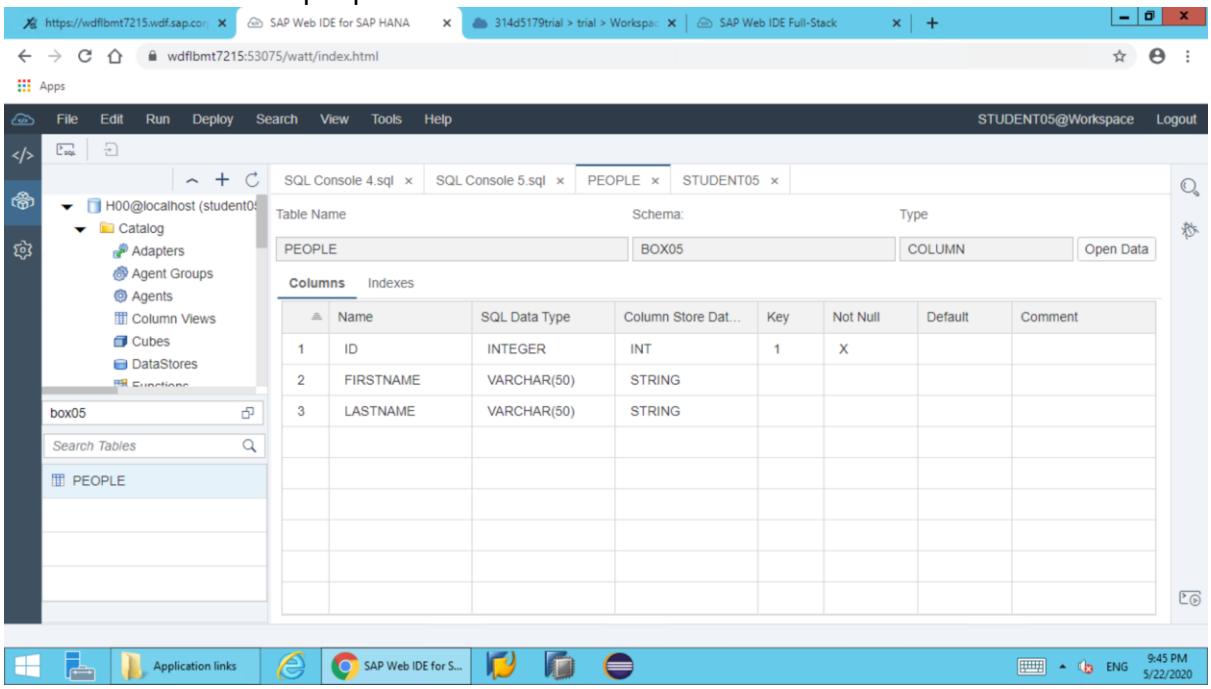


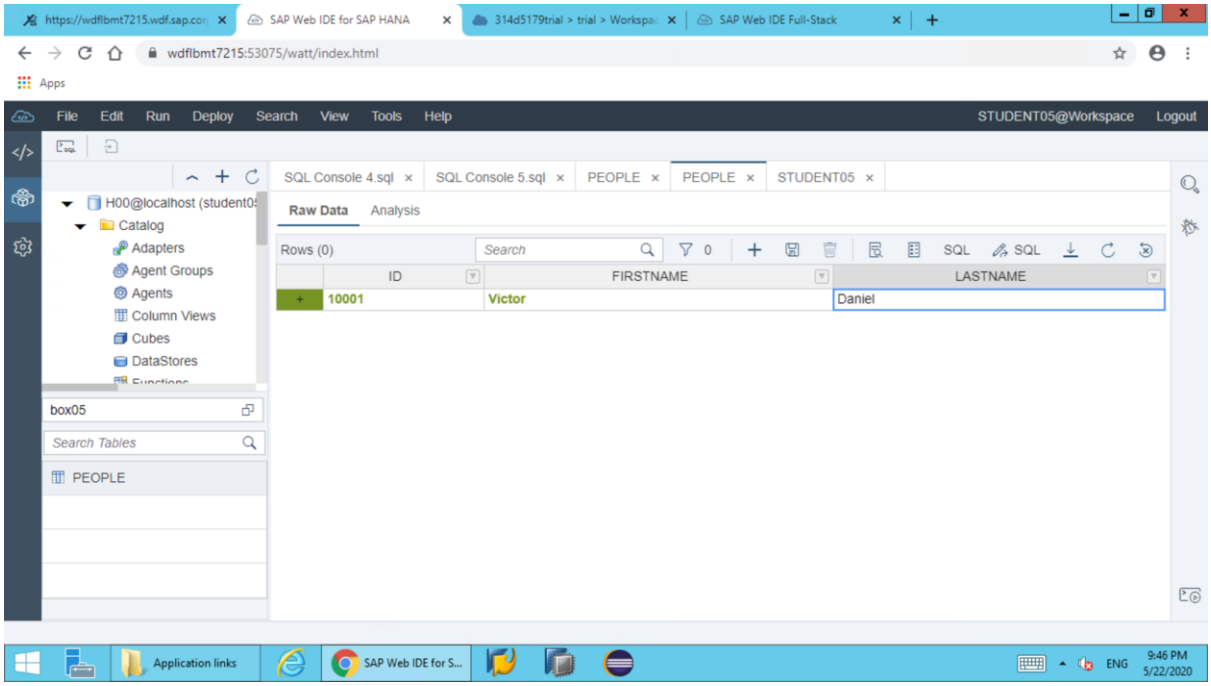
Now create a table named people

CREATE TABLE "BOX05"."PEOPLE" (id INT PRIMARY KEY, firstname VARCHAR(50), lastname VARCHAR(50));

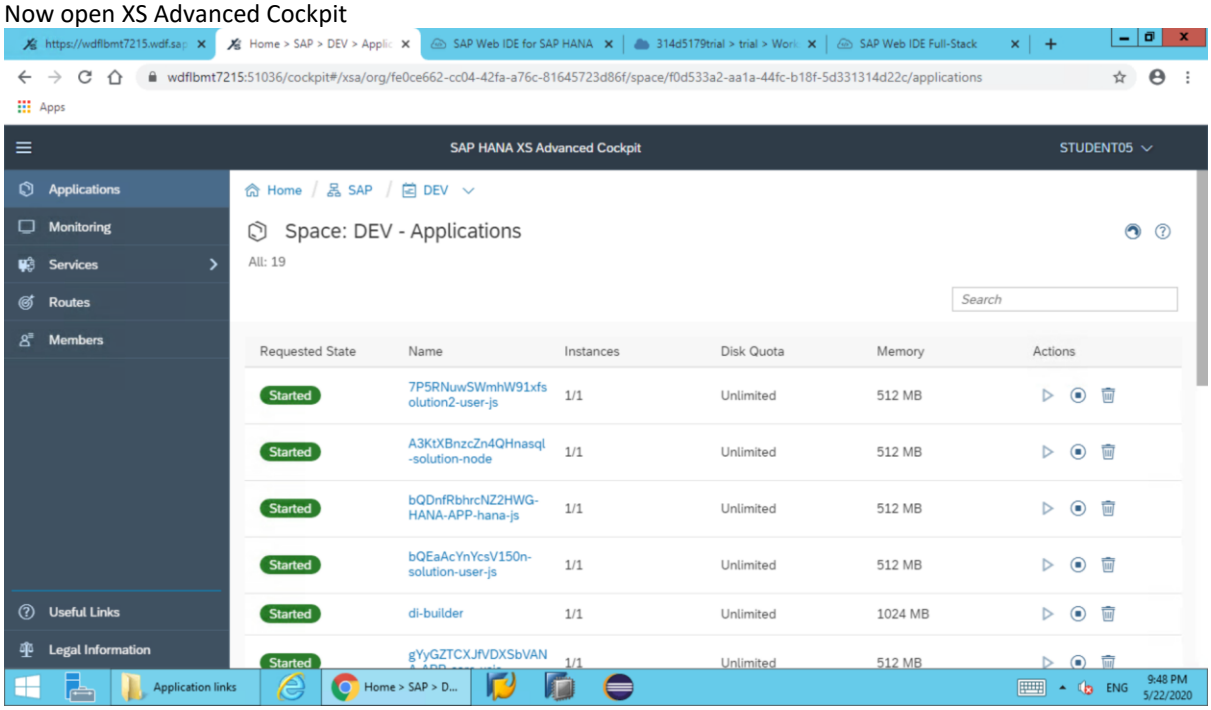


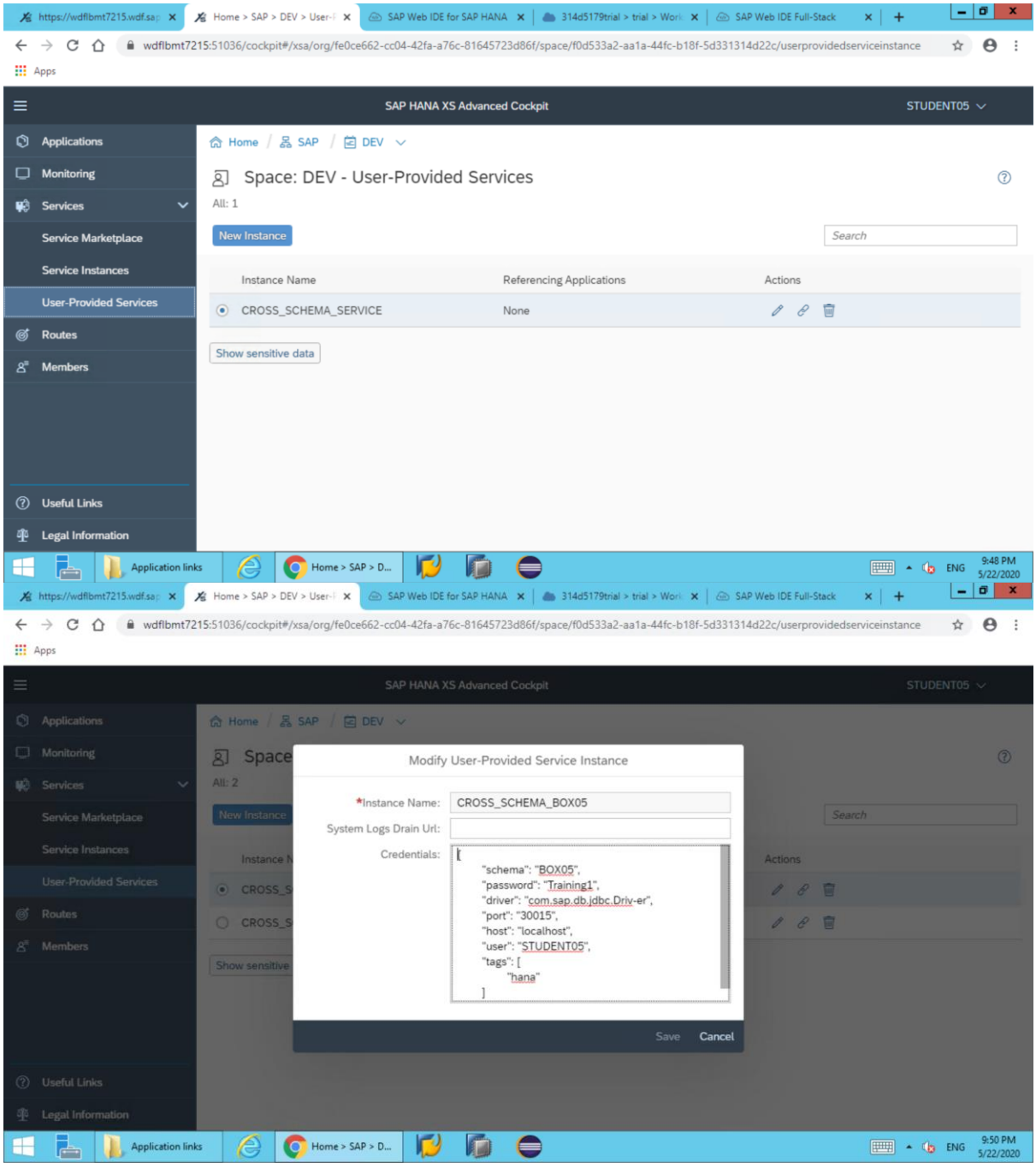
Now add some data in people table





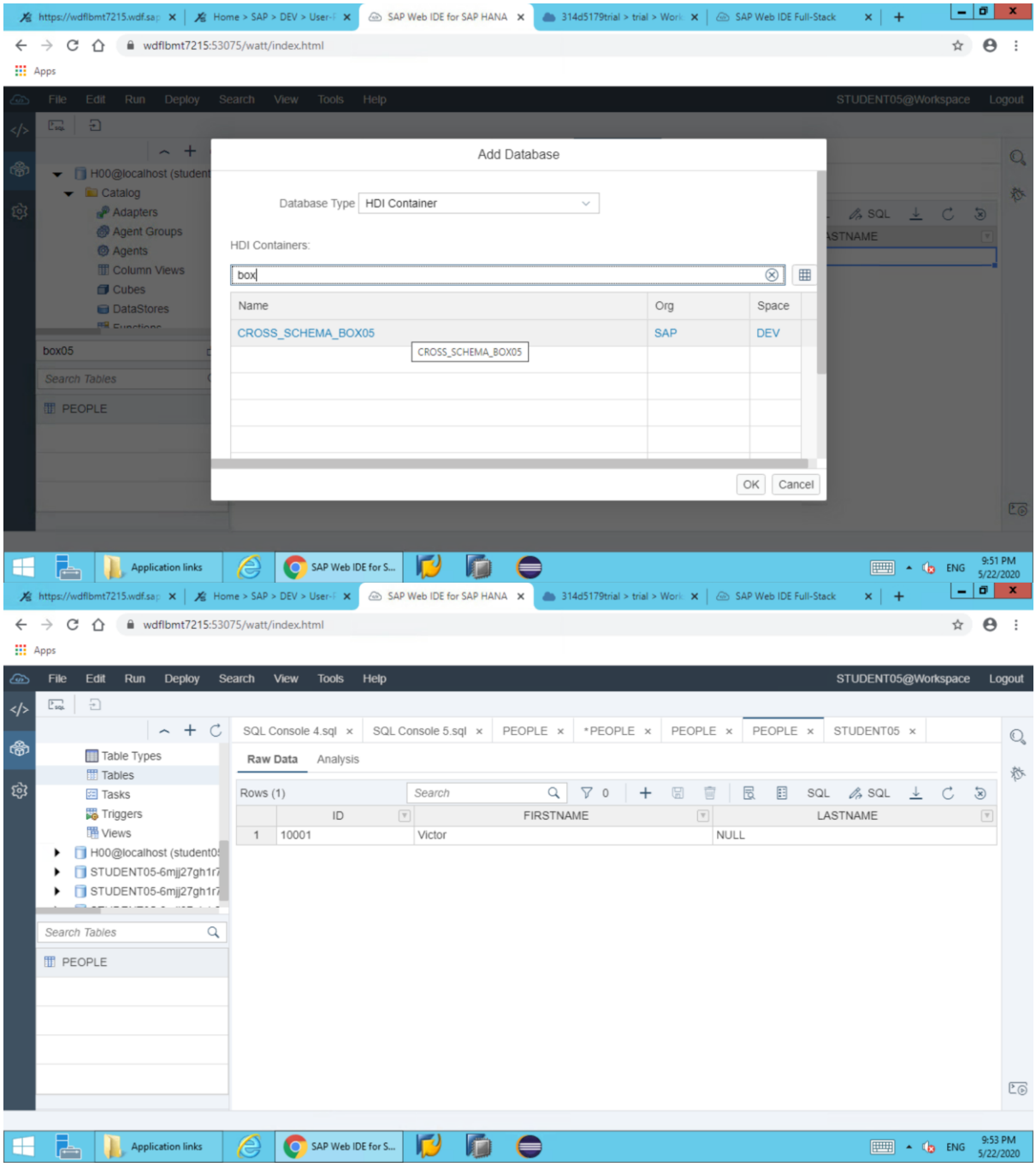
Now open XS Advanced Cockpit





{
"host": "localhost","port":
"30015","user":
"STUDENT##","password":
"Training1","driver":
"com.sap.db.jdbc.Driv-er","tags":
["hana"] ,"schema" : "BOX##" }

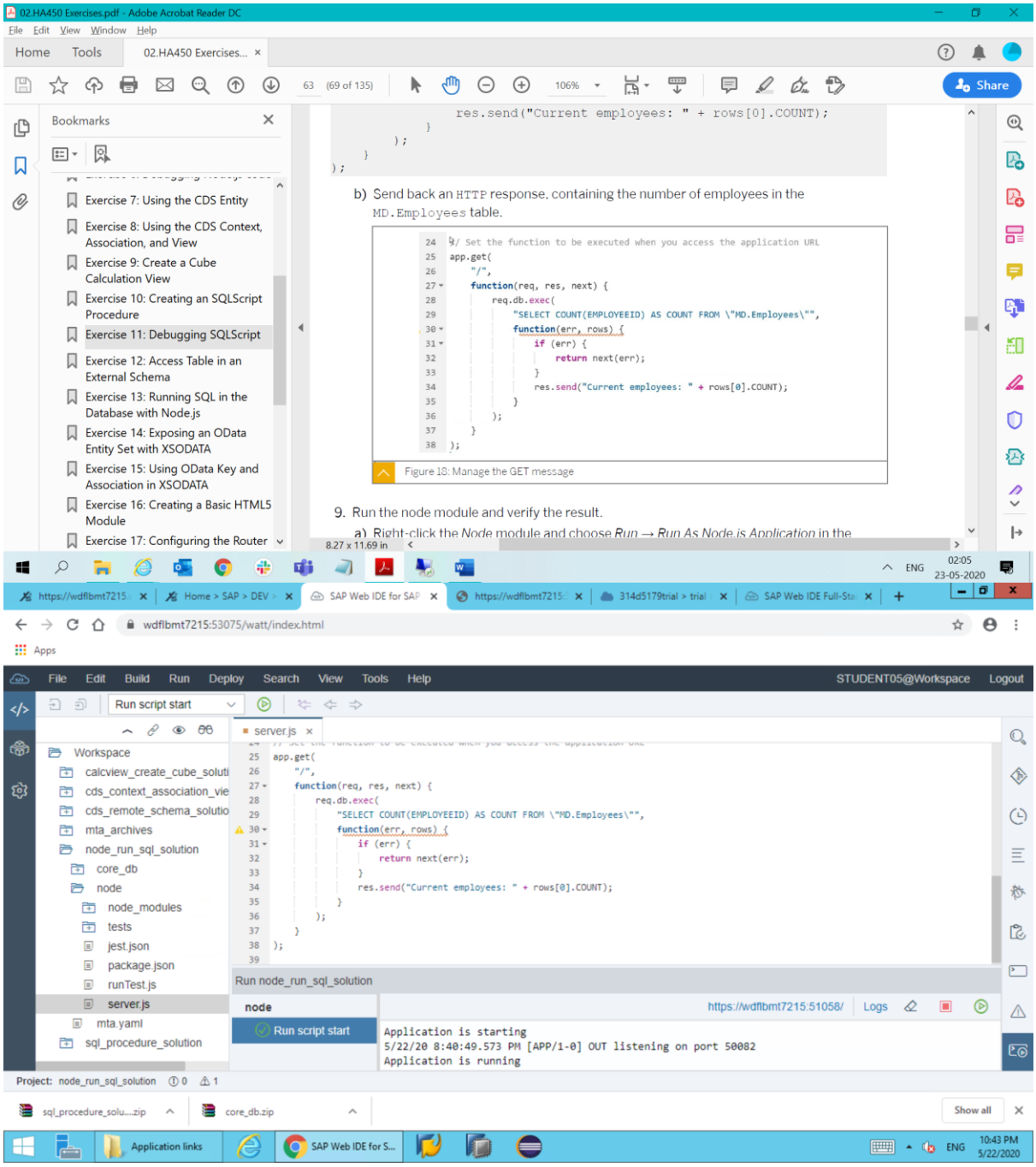
Now go back to webide and in database explorer. Add cross schema db



Now this remote db schema information is available. In order to run the sql commands we can do the heavy lifting using node.js . It has express.js extension which reads sql and transmits to the db layer.

Running SQL in the Database with Node.js

So remote database qureries can be written and transmitted using node.js application server layer.



Server.js

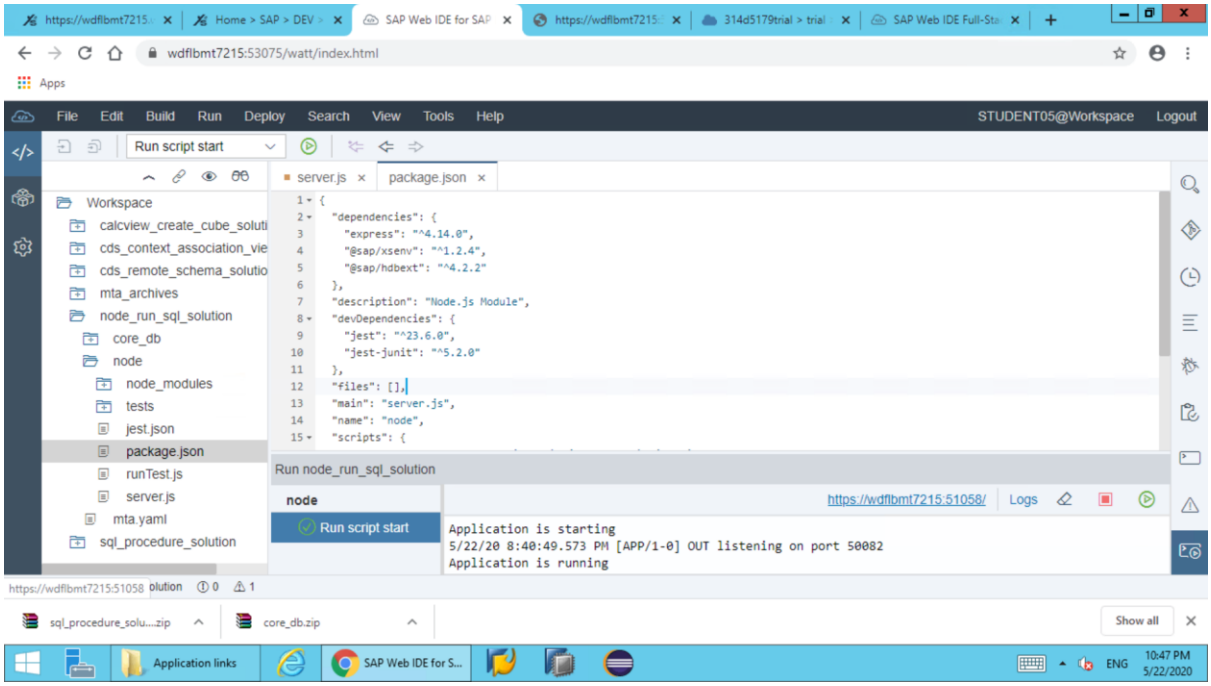
```
/*eslint no-console: 0*/
"use strict";

// Create the Express application framework
var express = require("express");
var app = express();

// Set the port the application listens to
var port = process.env.PORT || 3000;
app.listen(port, function() {
        console.log("listening on port " + port);
```

```
});

// Get XS services
var xsenv = require("@sap/xsenv");
var services = xsenv.getServices({
	hana: {tag: "hana"}
});

// Get HANA middleware and assign it to the application
var hdbext = require("@sap/hdbext");
app.use("/", hdbext.middleware(services.hana));

// Set the function to be executed when you access the application URL
app.get(
	"/",
	function(req, res, next) {
		req.db.exec(
			"SELECT COUNT(EMPLOYEEID) AS COUNT FROM \"MD.Employees\"",
			function(err, rows) {
				if (err) {
					return next(err);
				}
				res.send("Current employees: " + rows[0].COUNT);
			}
		);
	}
);
```

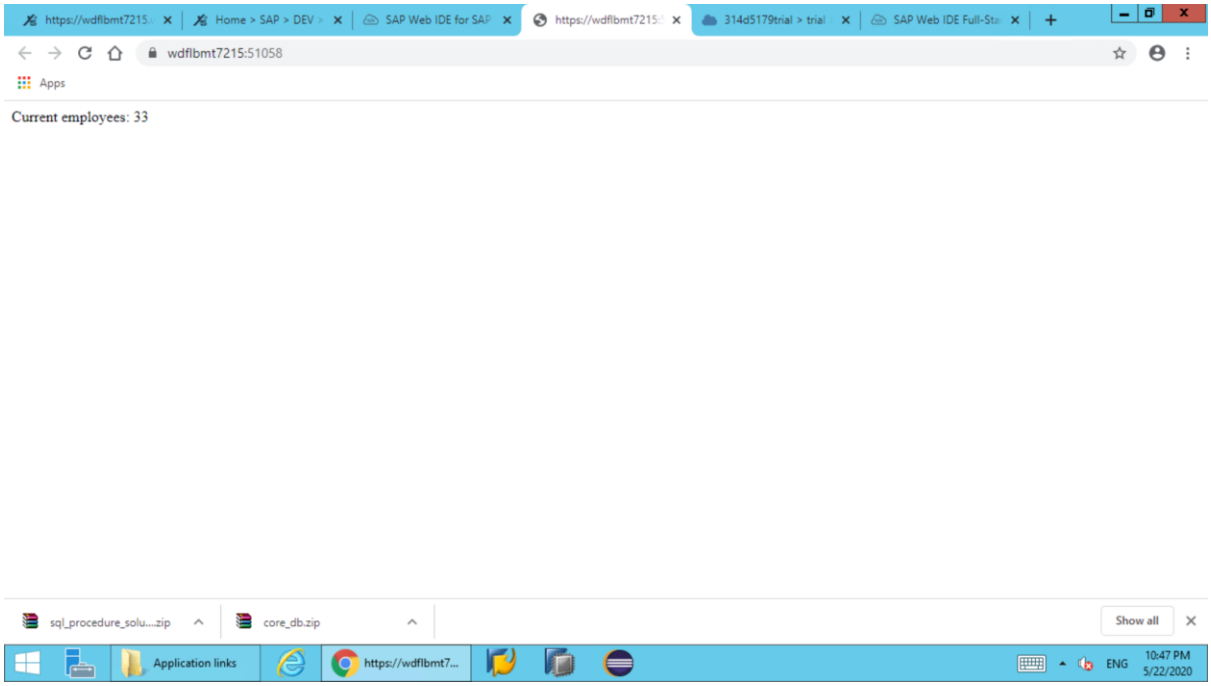
Package.json

```
{
  "dependencies": {
	"express": "^4.14.0",
	"@sap/xsenv": "^1.2.4",
	"@sap/hdbext": "^4.2.2"
  },
  "description": "Node.js Module",
  "devDependencies": {
    "jest": "^23.6.0",
    "jest-junit": "^5.2.0"
  },
  "files": [],
  "main": "server.js",
```

```
 "name": "node",
 "scripts": {
   "test": "node node_modules/jest/bin/jest --config jest.json",
   "test-coverage": "node node_modules/jest/bin/jest --coverage --config jest.json",
   "start": "node server.js"
 },
 "engines": {
   "node": "10.x"
 },
 "version": "1.0.0"
}
```



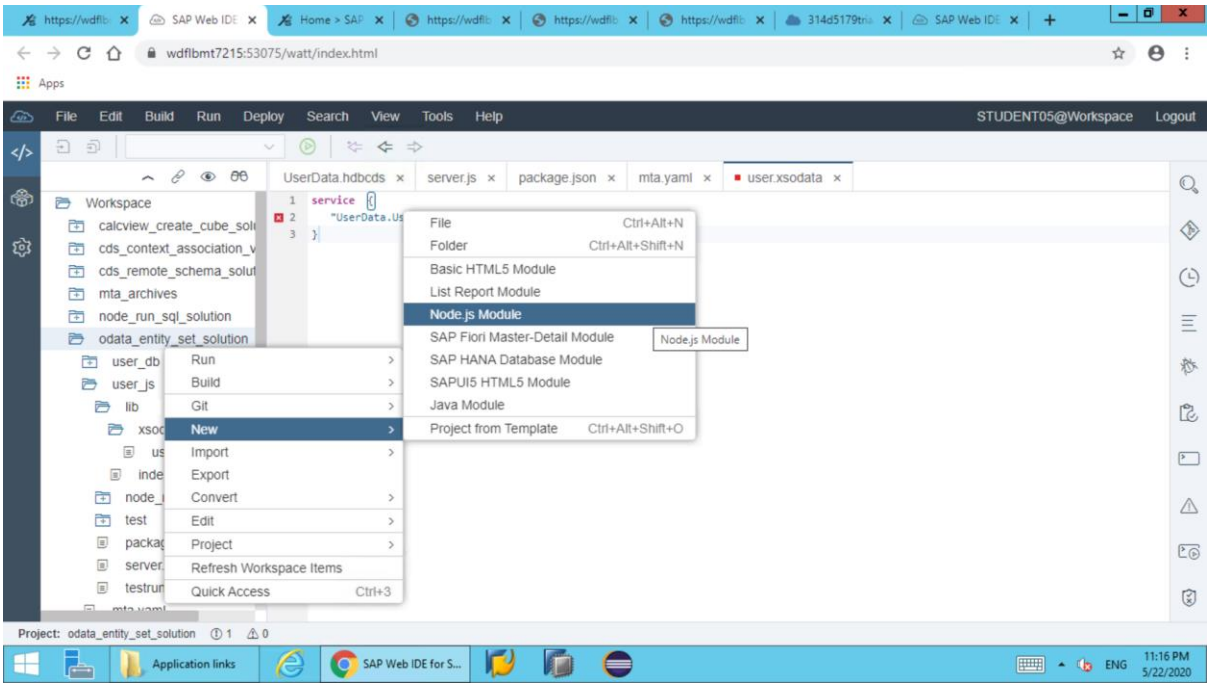
For node.js layer, only server.js and package.json are enough rest all files are not required. And they can be removed for sql commands.



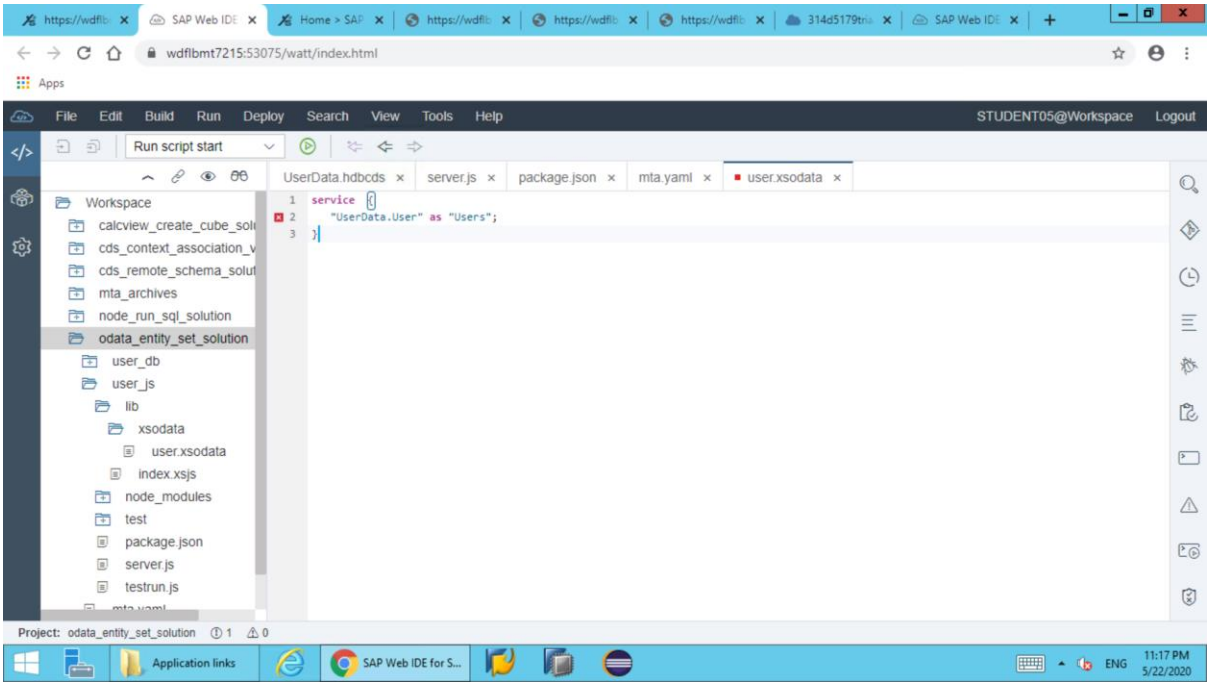
We could fetch no of employees count using sql command from node.js layer

Generating XSOdata

1. Create a node.js module with name user_js



Create a folder xsodata and add new file user.xsodata

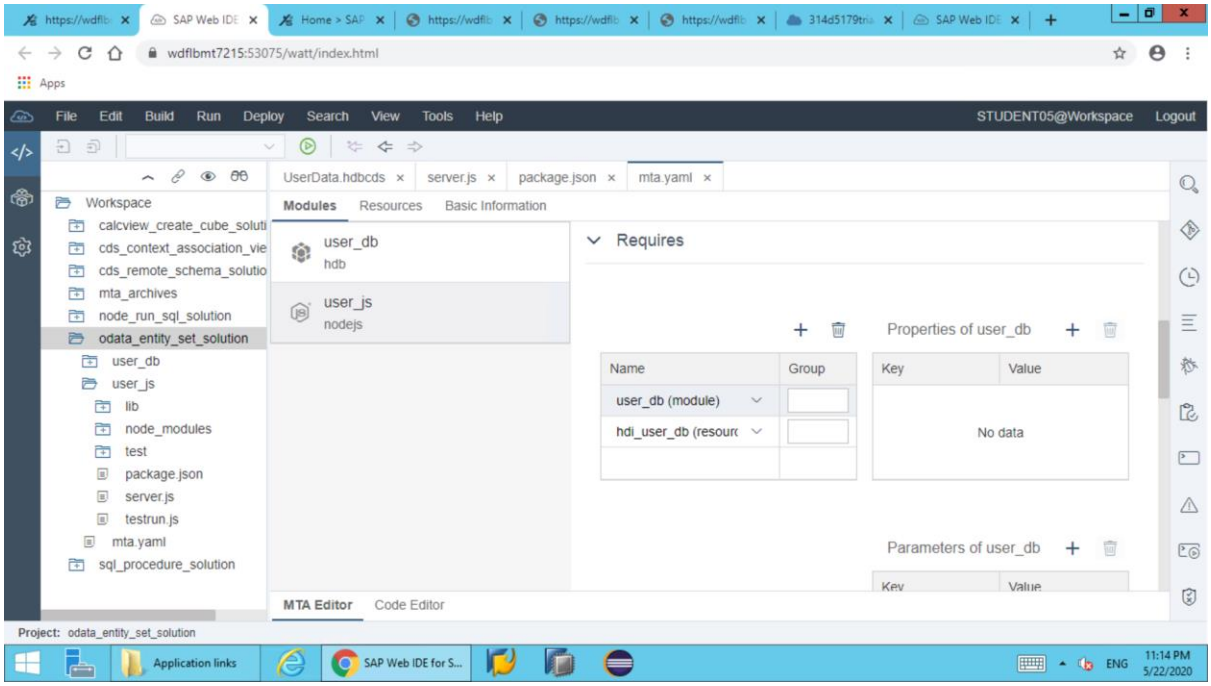


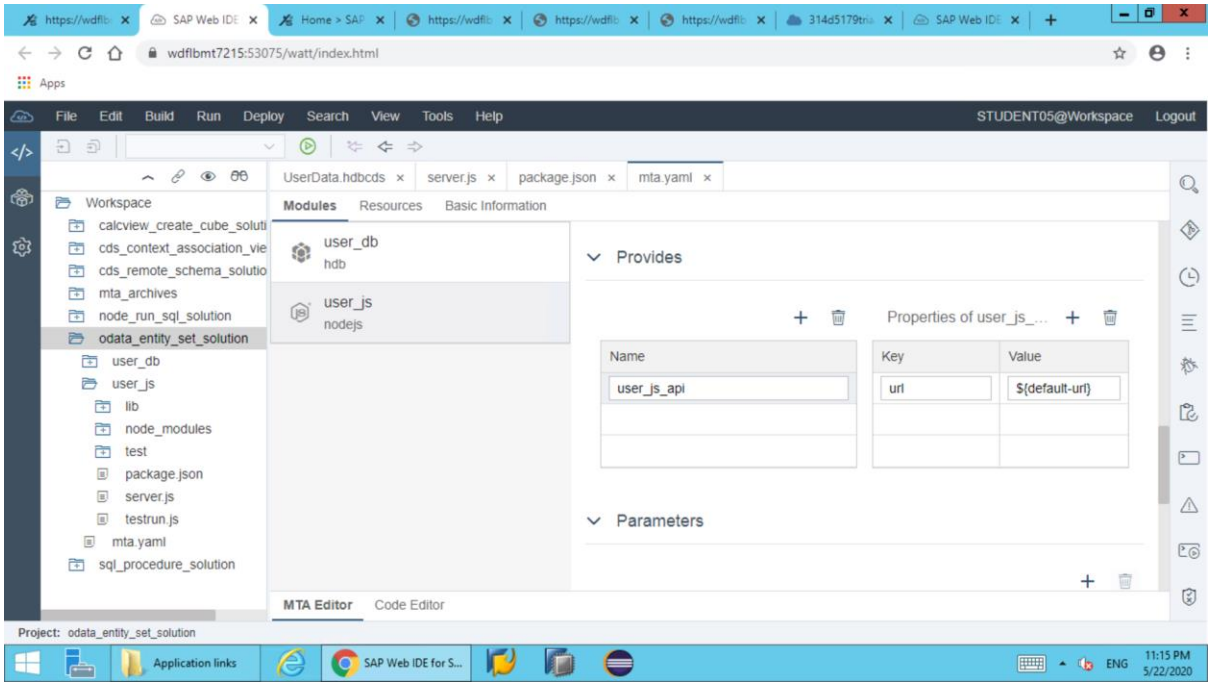
Add below snippet

```
    service {
"UserData.User" as "Users";
}
```

Now go to mta.yaml file.

Make sure these properties are defined properly using mta editor





Now verify the same in code editor as below

Mta.yaml

ID: odata_entity_set

_schema-version: '2.0'

version: 0.0.1

modules:

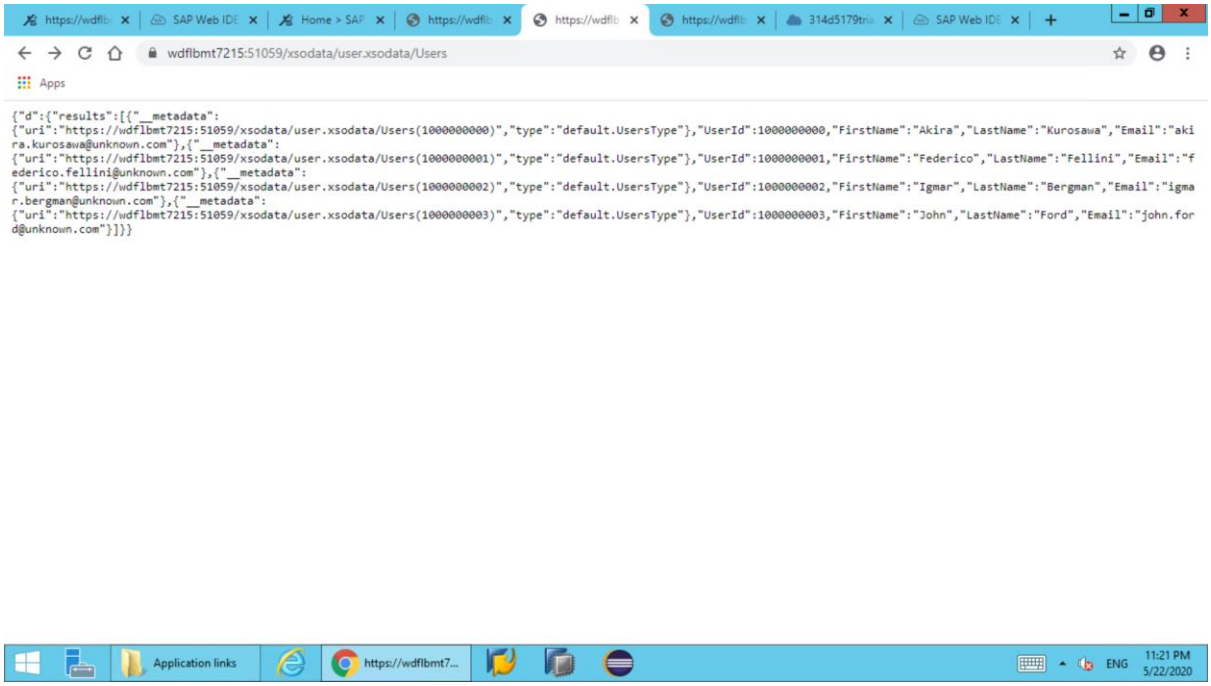
- name: user_db

```
    type: hdb
    path: user_db
    requires:
     - name: hdi_user_db
       properties:
         TARGET_CONTAINER: '~{hdi-container-name}'
  - name: user_js
    type: nodejs
    path: user_js
    provides:
     - name: user_js_api
       properties:
         url: '${default-url}'
    requires:
     - name: user_db
     - name: hdi_user_db
resources:
  - name: hdi_user_db
    properties:
      hdi-container-name: '${service-name}'
    type: com.sap.xs.hdi-container
```

Build and run as node.js application.

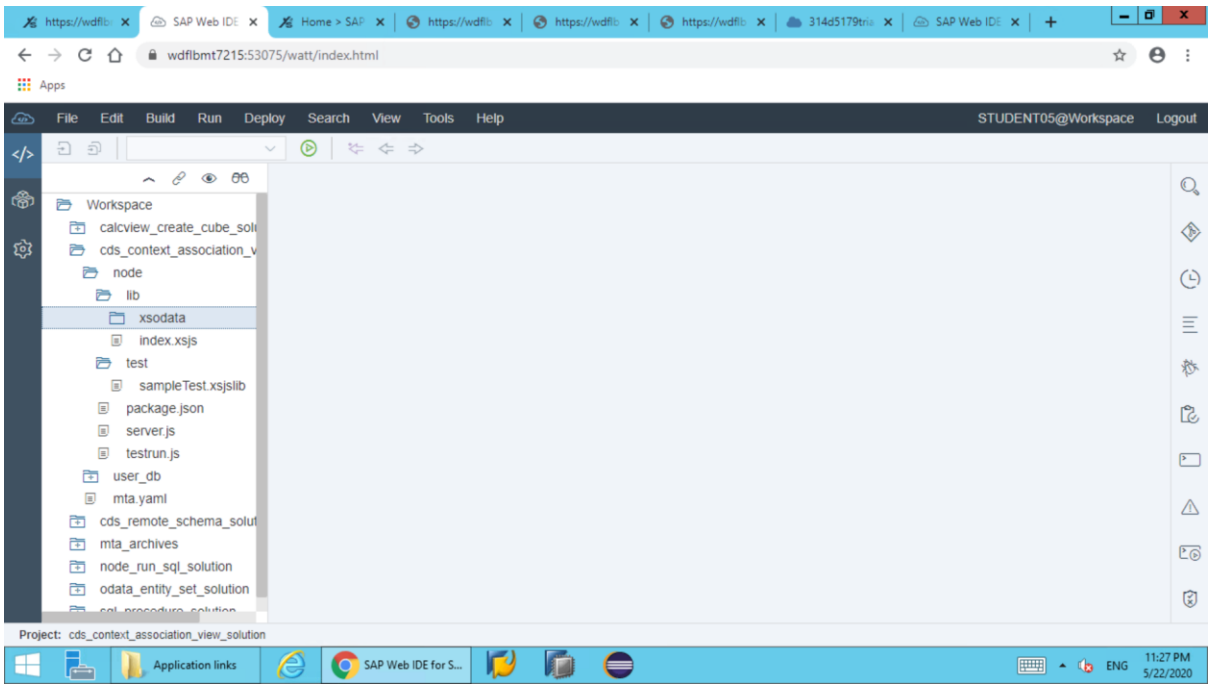
Replace https://wdflbmt7215:51059/index.xsjs with

https://wdflbmt7215:51059/xsodata/user.xsodata/Users

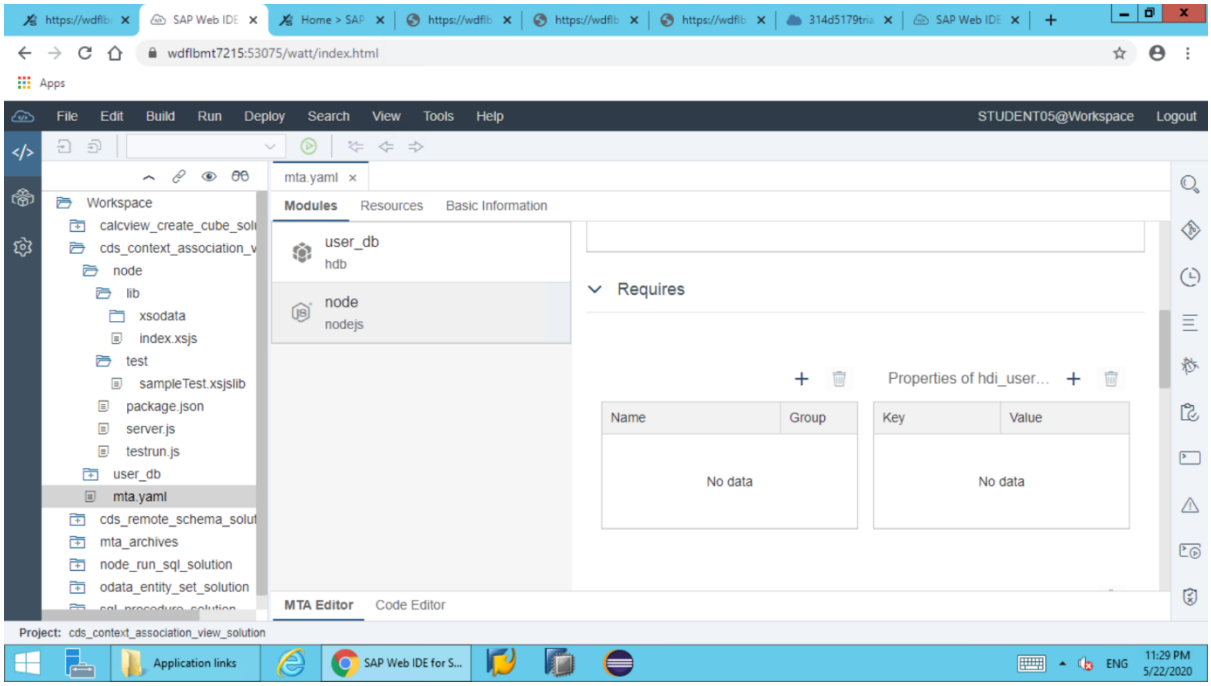


Creating XSOdata with Associations

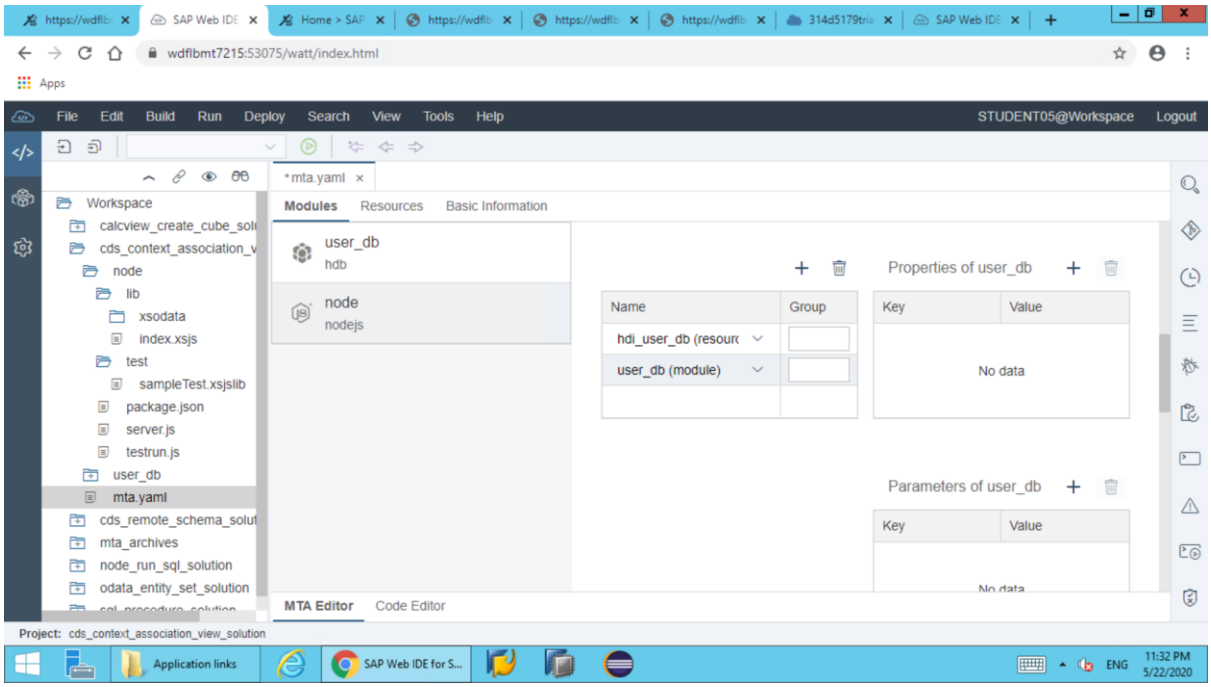
This is continuation to initial cds views with association exercise. Create node.js module with name node and add a subfolder xsodata in lib path.



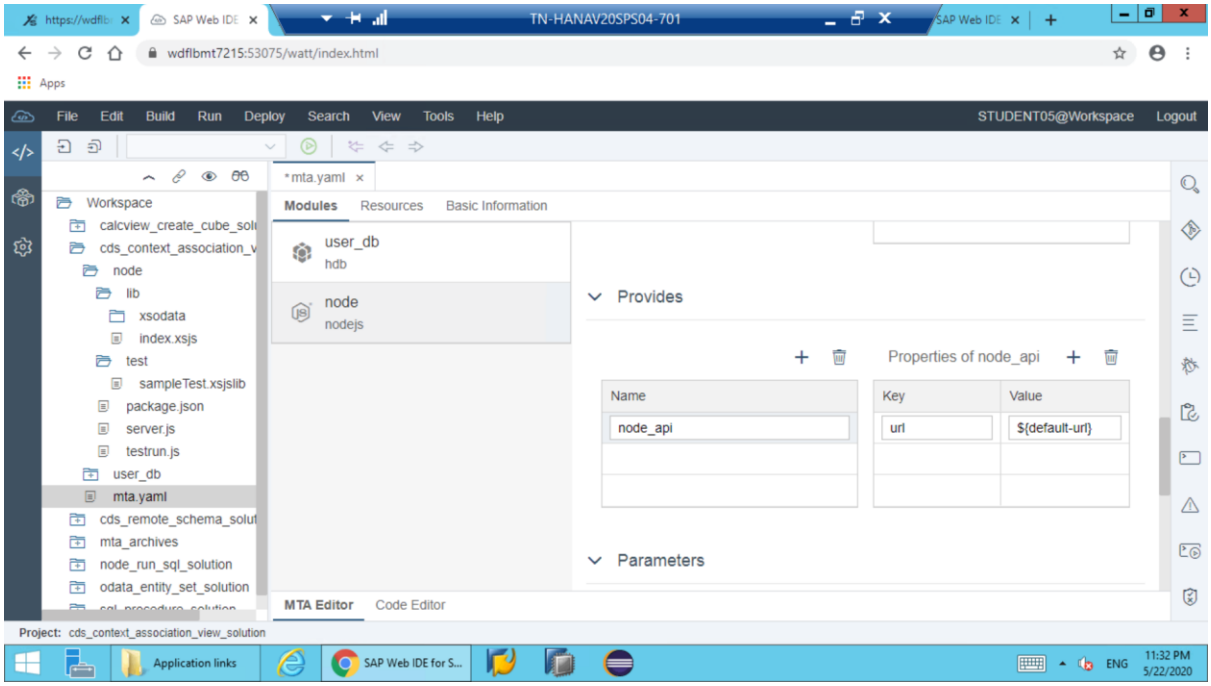
Go to mta.yaml file and select node module



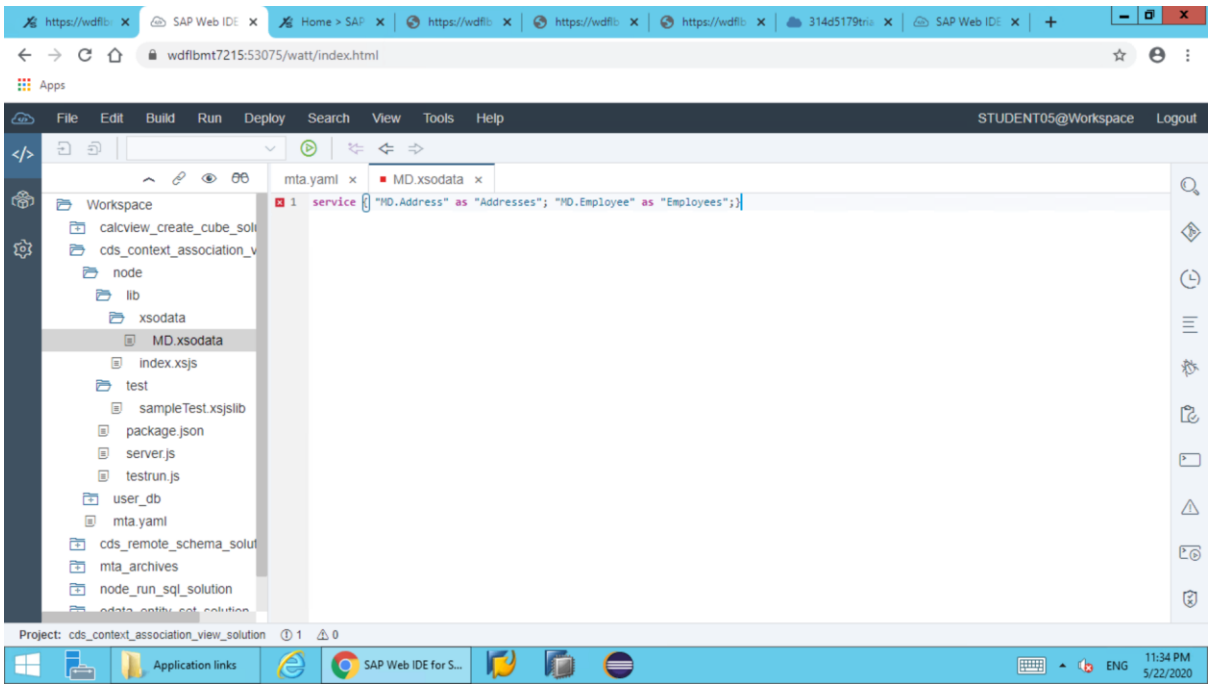
Add below values in Requires section



And in provide section , see below entries are available

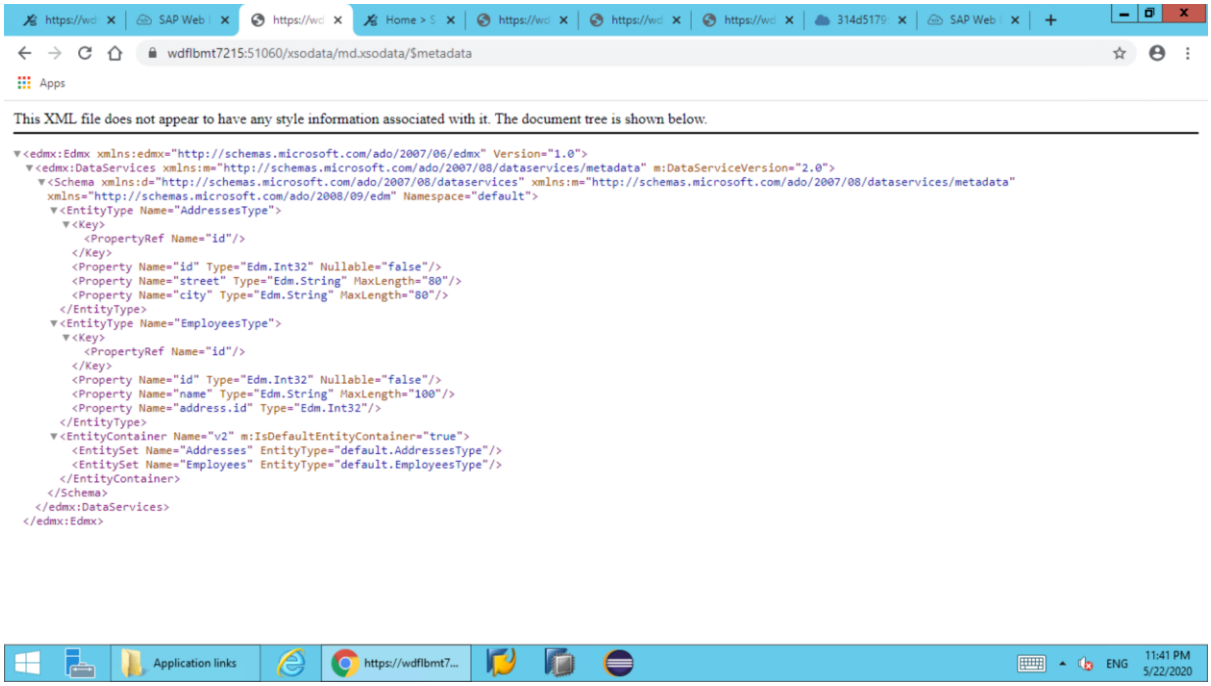


Now go to xsodata folder and add file md.xsodata




```
service {
 "MD.Address" as "Addresses";
"MD.Employee" as "Employees";
}
```

Now build and run the app as node.js application.



https://wdflbmt7215:51060/xsodata/md.xsodata/$metadata

Now lets add the Employee details view to look at the association.

For this we need to add md.xsodata file with viewname along with a key

```
service {
"MD.Address" as "Addresses";
 "MD.Employee" as "Employees";
"MD.EmployeeDetails" as "EmployeeDetails" key ("id");
}
```

Now we could display the views.
Lets add association . Add below snippet in MD.xsodata file

service {

"MD.Address" as "Addresses";
"MD.Employee" as "Employees";
"MD.EmployeeDetails" as "EmployeeDetails" key ("id");

association "EmployeeAddress" with referential constraint principal "Employees"("address.id") multiplicity "1" dependent "Addresses"("id") multiplicity "0..1";

}

Lets add navigation. Overwirte MD.xsodata with below snippet

service {

"MD.Address" as "Addresses";
"MD.Employee" as "Employees";
"MD.EmployeeDetails" as "EmployeeDetails" key ("id");

association "EmployeeAddress" with referential constraint principal "Employees"("address.id") multiplicity "1" dependent "Addresses"("id") multiplicity "0..1";

"MD.Address" as "Addresses" navigates ("EmployeeAddress" as "MyEmployee");
"MD.Employee" as "Employees" navigates ("EmployeeAddress" as "MyAddress");

}

In Simple form, snippet could be written as below

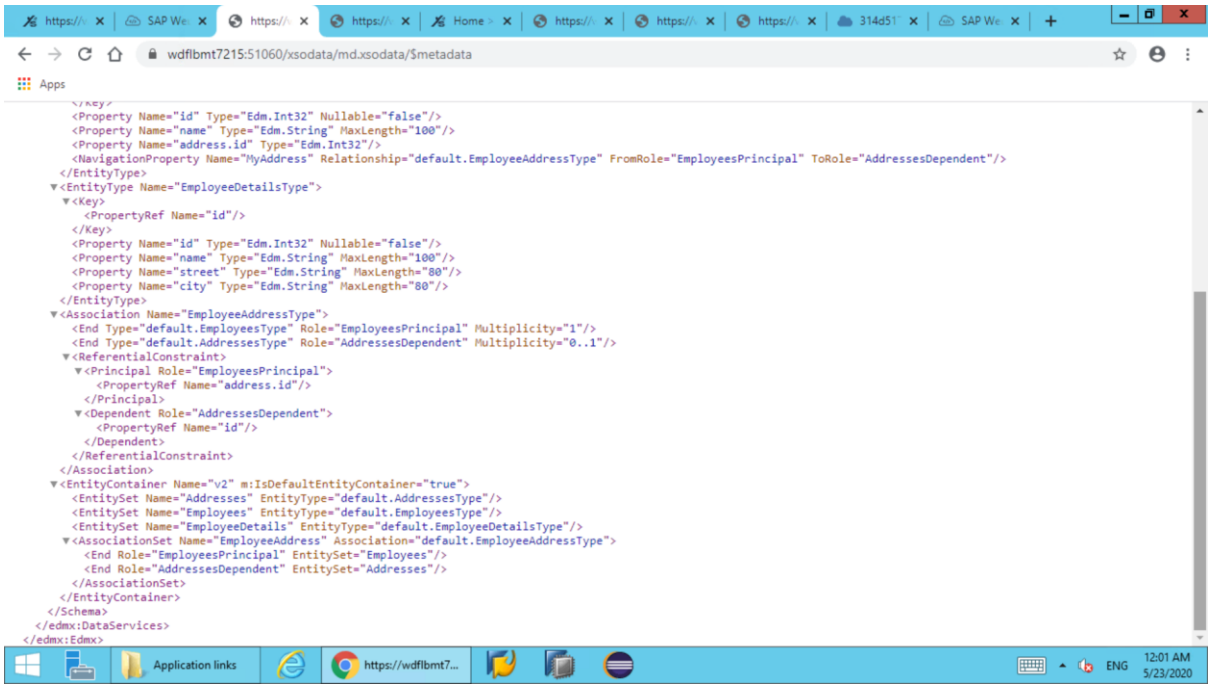
Final Snippet

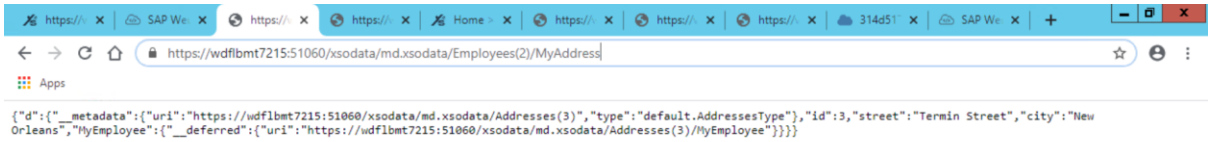
service {
"MD.Address" as "Addresses" navigates ("EmployeeAddress" as "MyEmployee");

"MD.Employee" as "Employees" navigates ("EmployeeAddress" as "MyAddress");

"MD.EmployeeDetails" as "EmployeeDetails" key ("id");

association "EmployeeAddress" with referential constraint principal "Employees"("address.id") multiplicity "1"
dependent "Addresses"("id") multiplicity "0..1";
}





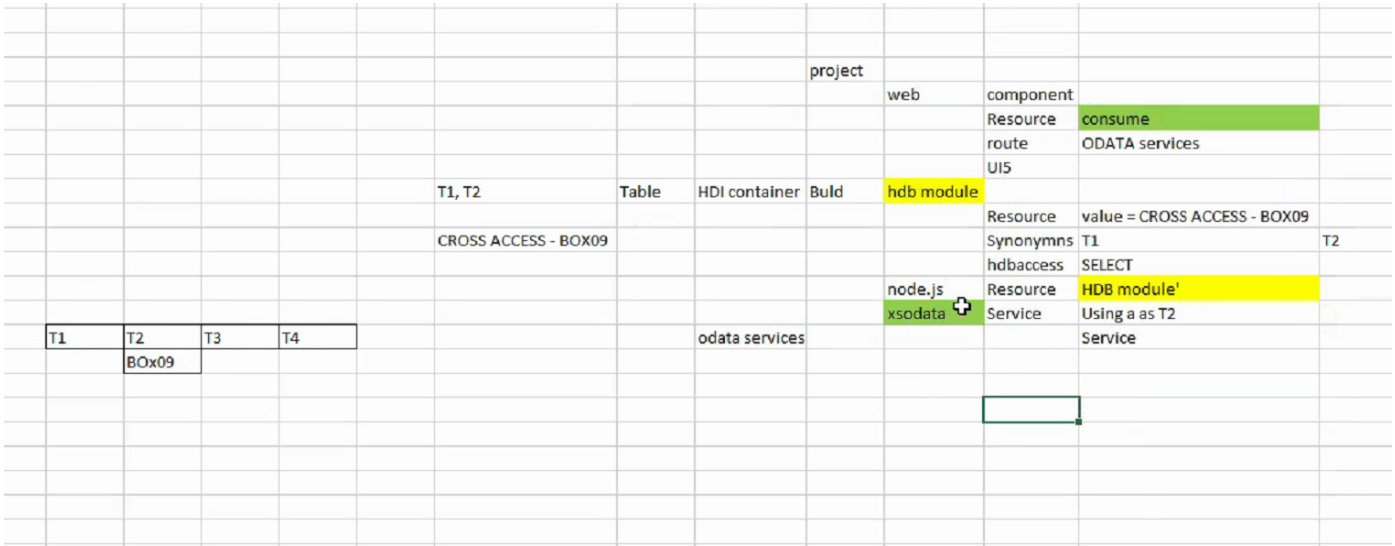
https://wdflbmt7215:51060/xsodata/md.xsodata/Employees(2)/MyAddress

Finally , we could navigate to association and display records. [Add green highlighted code and run directly if concept is clear]

😊

XSODATA Mapping Use Case Snapshot:

xsodata is consumed at Web Application end while it's a service at node -Application server layer.



Thank you.